МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕУЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

Рабочая программа профессионального модуля

ПМ 02. «Техническая поддержка и администрирование информационных ресурсов»

для специальности

09.02.09 «Веб-разработка»

Екатеринбург

2026

Чистяков Валерий Николаевич
я подтверждаю этот документ своей удостоверяющей подписью
место подписания
2026.03.18 15:28:25+05'00'

Рабочая программа ПМ 02. «Техническая поддержка и администрирование информационных ресурсов» разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.09 «Веб-разработка».

Организация-разработчик: ГАПОУ СО «Екатеринбургский монтажный колледж»

Разработчик Софьина Н.А. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК; Кузьмин В.П. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», 1КК;,

Рассмотрена методическим объединением Информационных технологий

Руководитель МО _____Софьина Н.А.

УТВЕРЖДАЮ

Заместитель директора по учебной работе

_____ Л.С.Хоринова

«13» марта 2026

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

1. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
«ПМ 02. Техническая поддержка и администрирование информационных ресурсов»

1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающихся должен освоить основной вид деятельности Техническая поддержка и администрирование информационных ресурсов, и соответствующие ему общие компетенции, и профессиональные компетенции:

1.1.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 02.	Использовать современные средства поиска, анализа и интерпретации и информационные профессиональной деятельности;
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 07.	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках.

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ВД 1	Техническая поддержка и администрирование информационных ресурсов.
ПК 2.1.	Устанавливать прикладное программное обеспечение и модулей информационных ресурсов, включая их настройку.
ПК 2.2.	Проводить работы по резервному копированию и развертыванию резервной копии информационных ресурсов.
ПК 2.3.	Настраивать права пользователей в соответствии с функциональными задачами (ролями) и на основании информации о поведенческих факторах.
ПК 2.4.	Применять программные средства обеспечения безопасности информации веб приложений.

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Владеть навыками	− подготовки программной среды для функционирования веб-приложения; − организации и обеспечения функционирования подсистемы резервного копирования и восстановления; − настройки прав доступа пользователя в существующей системе; − работы с инструментами мониторинга безопасности ИР; − выполнения типовых регламентных процедур по защите ИР;
Уметь	− соблюдать процедуру установки прикладного программного обеспечения в соответствии с документацией; − идентифицировать инциденты, возникающие при установке программного обеспечения, и принимать решение по изменению процедуры установки; − пользоваться нормативно-технической документацией в области программного обеспечения; − производить настройку параметров веб-сервера; − устанавливать систему управления базами данных (СУБД); − выполнять регламентные процедуры по резервированию данных; − устанавливать прикладное программное обеспечение для резервирования информационных ресурсов; − пользоваться нормативно-технической документацией в области программного обеспечения; − идентифицировать права пользователей в зависимости от функционала информационного ресурса; − регламентировать уровни прав и ролей пользователей информационных ресурсов; − применять регламентные процедуры управления правами доступа пользователей информационных ресурсов;
Знать	− принципы устройства и функционирования информационных ресурсов; − принципы устройства и функционирования программных средств и платформ для разработки веб-ресурсов; − современные стандарты взаимодействия компонентов распределенных приложений; − возможности ИР; − основы информационной безопасности веб-ресурсов; − принципы использования электронно-цифровых подписей и работы удостоверяющих центров; − инструменты и методы коммуникаций; − каналы коммуникаций; − модели коммуникаций; − технологии межличностной и групповой коммуникации в деловом взаимодействии, основ конфликтологии.

1.2. Количество часов, отводимое на освоение профессионального модуля

Всего часов: 372;

в том числе в форме практической подготовки: 216 часов;

Из них:
- на освоение МДК.02.01. Настройка и сопровождение информационных ресурсов: 92 часа;
в том числе в форме практической подготовки: 48 часов

- на освоение МДК.02.02. Обеспечение безопасности информационных ресурсов: 64 часа;
в том числе в форме практической подготовки: 24 часов;

практики, в том числе учебная: 36 часов;
производственная: 180 часов.

Промежуточная аттестация _____.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. Структура профессионального модуля

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля	Всего, час.	В т.ч. в форме практической. подготовки	Объем профессионального модуля, ак. час.						
				Обучение по МДК					Практики	
				Всего	В том числе					
					Лабораторных и практических занятий	Курсовых работ (проектов)	Самостоятельная работа	Промежуточная аттестация	Учебная	Производственная
1	2	3	4	5	6	7	8	9	10	11
ПК 2.1 ПК 2.5 ОК 1-5 ОК 7 ОК 9	Раздел 1. Настройка и сопровождение информационных ресурсов	**92**	48	**92**	48	0	0	0	**36**	**180**
ПК 2.2 ПК 2.3 ПК 2.4 ОК 1-5 ОК 7 ОК 9	Раздел 2. Обеспечение безопасности информационных ресурсов	**64**	24	**64**	24	0	0			
	Учебная практика	**36**	36						**36**	
	Производственная практика (по профилю специальности), часов	**180**	180							**180**
	Промежуточная аттестация	**0**	0							
	Всего:	**372**	**288**	**156**	**72**	**0**	**0**	**0**	**36**	**180**

2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовая работа (проект)	Объем, акад. ч / в том числе в форме практической подготовки, акад ч
1	**2**	**3**
Раздел 1. Настройка и сопровождение информационных ресурсов		
МДК.02.01. Настройка и сопровождение информационных ресурсов		
Тема 1.1. Установка прикладного программного обеспечения и модулей информационных ресурсов, включая их настройку	**Содержание**	**14**
	Понятие информационных ресурсов. Роль администратора. Задачи сопровождения и поддержки.	**2**
	Типы архитектур (клиент-сервер, веб-архитектура). Компоненты информационных ресурсов.	**2**
	Серверы, домены, хостинг, DNS, базы данных.	**2**
	Назначение серверных ОС. Linux и Windows Server.	**2**
	Подготовка сервера и установка служб.	**2**
	Назначение веб-сервера. Конфигурация Apache/Nginx.	**2**
	Создание пользователей. Разграничение прав.	**2**
	В том числе практических занятий и лабораторных работ	**18**
	Анализ структуры веб-сервера и сервисов.	**2**
	Анализ архитектуры существующих веб-ресурсов.	**2**

	Установка и начальная настройка сервера.	**2**
	Установка и запуск Apache/Nginx.	**2**
	Конфигурация доменов на сервере.	**2**
	Настройка прав доступа.	**2**
	Управление доступом к ресурсам.	**2**
	Настройка соединения.	**2**
	Использование инструментов мониторинга.	**2**
Тема 1.2. Обработка запросов заказчика в службе технической поддержки в соответствии с трудовым заданием	**Содержание**	**16**
	Организация хранения данных.	**2**
	Подключение и администрирование БД.	**2**
	Контроль загрузки системы и сервисов.	**2**
	Анализ лог-файлов.	**2**
	Методы резервного копирования данных.	**2**
	Методы восстановления системы.	**2**
	Методы повышения производительности.	**2**

	Плановое обслуживание и обновление.	**2**
	В том числе практических занятий и лабораторных работ	**12**
	Поиск и устранение ошибок.	**2**
	Создание резервных копий.	**2**
	Восстановление данных из резервной копии.	**2**
	Настройка параметров производительности.	**2**
	Установка обновлений.	**2**
	Администрирование и тестирование системы.	**2**
Самостоятельная работа 1. Подготовить конспект по теме «Информационные ресурсы и их роль в информационных системах». Рассмотреть виды информационных ресурсов и области их применения. 2. Изучить архитектуру клиент-серверных информационных систем. Составить схему взаимодействия клиента и сервера. 3. Изучить принципы работы доменных имен и системы DNS. Подготовить краткое описание процесса преобразования доменного имени в IP-адрес. 4. Изучить основные серверные операционные системы (Linux, Windows Server). Составить сравнительную таблицу их характеристик. 5. Изучить назначение и функции веб-серверов (Apache, Nginx). Подготовить краткое описание их возможностей и области применения. 6. Изучить способы мониторинга работы серверов. Описать инструменты контроля загрузки процессора, памяти и сетевых ресурсов. 7. Изучить методы резервного копирования данных. Подготовить краткое описание способов создания и хранения резервных копий.		**14**

Раздел 2. Обеспечение безопасности информационных ресурсов		
МДК.02.02. Обеспечение безопасности информационных ресурсов		
Тема 2.1. Резервное копирование и развертывание резервной копии информационных ресурсов	**Содержание**	**10**
	Основные понятия, цели и задачи защиты информации.	**2**
	Классификация угроз информационным ресурсам.	**2**
	Назначение политики безопасности и её основные положения.	**2**
	Организационные, программные и технические методы защиты.	**2**
	Основы криптографии и методы шифрования.	**2**
	В том числе практических и лабораторных занятий	**8**
	Определение возможных угроз информационным ресурсам.	**2**
	Использование программ для шифрования информации.	**2**
	Выявление угроз в компьютерных сетях.	**2**
	Конфигурирование firewall.	**2**
Тема 2.2. Настройка прав пользователей в соответствии с функциональными задачами (ролями)	**Содержание**	**10**
	Назначение, виды и применение электронной подписи.	**2**
	Основные угрозы в компьютерных сетях.	**2**
	Назначение и принцип работы firewall.	**2**
	Виды вредоносных программ и методы защиты.	**2**

	Типы уязвимостей веб-ресурсов.	**2**
	В том числе практических и лабораторных занятий	**8**
	Поиск уязвимостей веб-сайта.	**2**
	Настройка базовых механизмов защиты.	**2**
	Использование средств мониторинга.	**2**
	Установка и настройка антивируса.	**2**
Тема 2.3. Применение программных средств обеспечения безопасности информации веб-приложений	**Содержание**	**8**
	Основные методы защиты веб-ресурсов.	**2**
	Методы идентификации пользователей.	**2**
	Средства мониторинга информационных систем.	**2**
	Выявление и реагирование на инциденты.	**2**
	В том числе практических и лабораторных занятий	**8**
	Настройка прав доступа.	**2**
	Работа с логами безопасности.	**2**
	Выявление подозрительной активности.	**2**
	Проверка защищенности информационного ресурса.	**2**
Самостоятельная работа 1. Подготовить конспект по теме «Политика информационной безопасности». 2. Изучить виды электронной подписи и области их применения.		**12**

3. Изучить основные виды вредоносных программ. 4. Изучить методы аутентификации пользователей. 5. Подготовить описание этапов реагирования на инциденты информационной безопасности. 6. Подготовить сообщение «Современные методы защиты информационных ресурсов».	
Учебная практика Соадминистрирование компонентов информационной системы **Виды работ** 1. Ознакомление с архитектурой информационной системы и её основными компонентами. 2. Изучение используемых в информационной системе программных платформ, серверов приложений и баз данных. 3. Установка и настройка прикладного программного обеспечения для функционирования информационных ресурсов. 4. Установка и подключение модулей информационных ресурсов. 5. Настройка параметров программных модулей информационной системы. 6. Настройка среды функционирования информационной системы. 7. Создание и настройка учетных записей пользователей. 8. Настройка прав доступа пользователей в соответствии с их функциональными ролями. 9. Управление ролями пользователей и разграничение доступа к информационным ресурсам. 10. Анализ пользовательской активности и поведенческих факторов при настройке прав доступа. 11. Изучение методов обеспечения безопасности веб-приложений. 12. Применение программных средств защиты информации веб-приложений. 13. Настройка механизмов аутентификации и авторизации пользователей. 14. Организация резервного копирования информационных ресурсов. 15. Создание резервных копий данных информационной системы. 16. Выполнение восстановления информационных ресурсов из резервной копии. 17. Мониторинг состояния информационной системы и её компонентов.	**72**

18. Выявление и устранение ошибок в работе программных модулей. 19. Ведение технической документации по администрированию информационных ресурсов. 20. Подготовка отчёта по результатам прохождения учебной практики.	
Производственная практика **Виды работ** 1. Ознакомление со структурой организации, правилами внутреннего распорядка и требованиями техники безопасности. 2. Изучение структуры информационной системы организации и её основных компонентов. 3. Анализ используемых программных платформ, серверов приложений и баз данных. 4. Участие в установке и настройке прикладного программного обеспечения, используемого в информационной системе. 5. Установка и настройка программных модулей информационных ресурсов. 6. Настройка параметров функционирования прикладного программного обеспечения. 7. Администрирование компонентов информационной системы и контроль их работоспособности. 8. Создание, настройка и сопровождение учетных записей пользователей. 9. Настройка прав доступа пользователей к информационным ресурсам в соответствии с их функциональными ролями. 10. Разграничение доступа к данным и программным модулям системы. 11. Анализ пользовательской активности и применение поведенческих факторов при настройке прав доступа. 12. Применение программных средств обеспечения безопасности веб-приложений. 13. Настройка механизмов аутентификации и авторизации пользователей. 14. Участие в организации резервного копирования информационных ресурсов. 15. Создание резервных копий программных модулей и данных информационной системы. 16. Выполнение работ по восстановлению информационных ресурсов из резервных копий. 17. Мониторинг состояния серверов, программных модулей и информационных ресурсов.	**180**

18. Выявление и устранение ошибок в работе информационной системы. 19. Ведение технической документации по администрированию и сопровождению информационных ресурсов. 20. Подготовка отчёта по результатам прохождения производственной практики.	
Всего	**372**

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:

Лаборатория Программного обеспечения и сопровождения компьютерных систем, оснащенная в соответствии с п. 6.1.2.3 Примерной образовательной программы по специальности 09.02.09 Веб-разработка.

Оснащенные базы практики в соответствии с п 6.1.2.4 примерной образовательной программы по специальности 09.02.09 Веб-разработка.

Реализация образовательной программы предполагает обязательную учебную и производственную практику.

Учебная практика реализуется в мастерских профессиональной образовательной организации и требует наличия оборудования, инструментов, расходных материалов, обеспечивающих выполнение всех видов работ, определенных содержанием программ профессиональных модулей, в том числе оборудования и инструментов, используемых при проведении чемпионатов профессионального мастерства и указанных в инфраструктурных листах конкурсной документации по компетенции «Веб-технологии» (или их аналогов).

Производственная практика реализуется в организациях, обеспечивающих деятельность обучающихся в профессиональной области информационных технологий.

Оборудование предприятий и технологическое оснащение рабочих мест производственной практики должно соответствовать содержанию профессиональной деятельности и дать возможность обучающемуся овладеть профессиональными компетенциями по всем видам деятельности, предусмотренными программой, с использованием современных технологий, материалов и оборудования.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организации выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список может быть дополнен новыми изданиями.

3.2.1. Основные печатные издания

1. Лисьев, Г. А. Программное обеспечение компьютерных сетей и web-серверов: учебное пособие / Г.А. Лисьев, П.Ю. Романов, Ю.И. Аскерко. — Москва: ИНФРА-М, 2023. — 145 с

2. Хоффман, Эндрю Безопасность веб-приложений - Санкт-Петербург: Издательство Питер, 2021. - 336 с. - ISBN 978-5-4461-1786-4.

3.2.2. Основные электронные издания

1. Вехен Джульен. Безопасный DevOps. Эффективная эксплуатация систем. - Санкт-Петербург: Издательство Питер, 2020. - 432 с. - ISBN 978-5-4461-1336-1. - URL: https://ibooks.ru/bookshelf/365290/reading

2. Волох С. В. Ubuntu Linux с нуля. — 2-е изд., перераб. и доп. / С.В. Волох. - Санкт-Петербург: Издательство БХВ-Петербург, 2021. - 416 с. - ISBN 978-5-9775-6761-9. - URL: https://ibooks.ru/bookshelf/380034/reading

3. Хоффман Эндрю. Безопасность веб-приложений. - Санкт-Петербург: Издательство Питер, 2021. - 336 с. - ISBN 978-5-4461-1786-4. - URL: https://ibooks.ru/bookshelf/378722/. - Текст: электронный.

4. Форристал Д. Защита от хакеров Web-приложений/ Форристал Д. и др. пер. с англ. В. Зорина (Серия «Информационная безопасность»). / Д.И. Форристал. - Москва: Издательство ДМК Пресс, 2021. - 496 с. - ISBN 5-94074-258-0. - URL: https://ibooks.ru/bookshelf/364152/reading. - Текст: электронный.

4. Фленов М. Е. Web-сервер глазами хакера. — 3-е изд., перераб. и доп. — (Глазами хакера) / М.Е. Фленов. - Санкт-Петербург: Издательство БХВ-Петербург, 2021. - 256 с. - ISBN 978-5-9775-6795-4. - URL: https://ibooks.ru/bookshelf/380043/reading. - Текст: электронный.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
ПК 2.1 Устанавливать прикладное программное обеспечение и модулей информационных ресурсов, включая их настройку;	Оценка «**отлично**» - целевое веб-приложение способно выполняться согласно всем тестовым условиям и разработана необходимая документация Оценка «**хорошо**» - целевое веб-приложение способно выполняться согласно всем тестовым условиям. Оценка «**удовлетворительно**» - часть базовых компонентов ПО для веб-приложения установлено или подробно описаны требуемые операции.	Экзамен/зачет в форме собеседования: практическое задание по установке необходимого ПО для работы веб-приложения Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной практики
ПК 2.2 Проводить работы по резервному копированию и развертыванию резервной копии	Оценка **«отлично»** - Сформирован план создания резервных копий, настроено специальное ПО для резервирования, продемонстрирован процесс восстановления данных, в том числе автоматический.	Экзамен/зачет в форме собеседования: практическое задание по настройке автоматического резервирования и восстановления данных.

информационных ресурсов;	Оценка **«хорошо»** - Сформирован план создания резервных копий, настроено специальное ПО для резервирования. Оценка **«удовлетворительно»** - Сформирован план создания резервных копий, настроено специальное ПО для резервирования.	Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 2.3 Настраивать права пользователей в соответствии с функциональными задачами (ролями) и на основании информации о поведенческих факторах;	Оценка «**отлично**» - Настроены права доступа к ФС И БД, распределены роли в CMS, оформлен регламент доступа. Оценка «**хорошо**» - Настроены права доступа к ФС И БД, распределены роли в CMS. Оценка «**удовлетворительно**» - Распределены роли в CMS, оформлен регламент доступа.	Экзамен/зачет в форме собеседования: практическое задание по настройке прав доступа к ФС в ОС Linux и БД, распределение ролей в CMS. Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ПК 2.4 Применять программные средства обеспечения безопасности информации веб приложений;	Оценка «**отлично**» - проведен аудит безопасности веб-сервиса, настроено специальное ПО для обеспечения безопасности работы веб-приложения и составлен отчет с рекомендациями. Оценка «**хорошо**» - проведен аудит безопасности веб-сервиса, настроено специальное ПО для обеспечения безопасности работы веб-приложения и составлен отчет с рекомендациями по базовым характеристикам. Оценка «**удовлетворительно**» - проведен аудит безопасности веб-сервиса и составлен отчет с рекомендациями по базовым характеристикам.	Экзамен/зачет в форме собеседования: практическое задание по проведению аудита безопасности веб-сервиса и настройки специального ПО для обеспечения безопасности работы веб-приложения. Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ПК 2.5 Обрабатывать запросы заказчика в службе технической поддержке в соответствии с трудовым заданием;	Оценка «**отлично**» - составлена блок-схема работы оператора технической поддержки и решен инцидент от гипотетического пользователя. Оценка «**хорошо**» - составлена блок-схема работы оператора технической поддержки и решен инцидент от	Экзамен/зачет в форме собеседования: практическое задание по составлению блок-схемы работы оператора технической поддержки и следование инструкции при

	гипотетического пользователя с грубыми нарушениями. Оценка «**удовлетворительно**» - составлена блок-схема работы оператора технической поддержки или решен инцидент от гипотетического пользователя.	обращении гипотетического пользователя. Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;	− обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; − адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертное наблюдение за выполнением работ
ОК 02. Использовать современные средства поиска, анализа и интерпретации и информационные профессиональной деятельности;	− использование различных источников, включая электронные ресурсы, медиареурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;	− демонстрация ответственности за принятые решения − обоснованность самоанализа и коррекция результатов собственной работы;	

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	− взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; − обоснованность анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;	− Демонстрировать грамотность устной и письменной речи, ясность формулирования и изложения мыслей	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;	− эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; − демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.	− эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МДКн.02.01 НАСТРОЙКА И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.07 «Информационные системы и программирование»** и предназначен для проведения текущего контроля и промежуточной аттестации по дисциплине **МДК.02.01 «Настройка и сопровождение информационных ресурсов»**.

ФОС включает в себя комплекс теоретических вопросов и практических задач, моделирующих деятельность администратора веб-ресурсов, системного администратора и специалиста по эксплуатации информационных систем.

2. Цель и задачи курса

Цель: Формирование навыков развертывания, конфигурирования и технической поддержки работоспособности веб-ресурсов на серверных платформах.

Задачи:

- **Изучение архитектурных основ:** Понимание принципов функционирования веб-стеков, систем доменных имен (DNS) и механизмов взаимодействия «клиент-сервер».
- **Администрирование серверных ОС:** Овладение навыками настройки Linux и Windows Server для размещения информационных ресурсов.
- **Управление веб-службами:** Установка и тонкая настройка веб-серверов (Apache, Nginx) и систем управления базами данных (СУБД).
- **Обеспечение непрерывности работы:** Внедрение стратегий резервного копирования, планового обновления ПО и мониторинга производительности.
- **Диагностика и устранение неисправностей:** Развитие навыков анализа лог-файлов и оперативного исправления ошибок конфигурации.

3. Структура оценочных материалов

Содержание ФОС разделено на функциональные блоки, соответствующие этапам эксплуатации ресурсов:

- **Модуль «Инфраструктурная подготовка»:** Контроль умений по установке серверных ОС, настройке сетевых интерфейсов и базовых системных служб.
- **Модуль «Развертывание веб-сервисов»:** Проверка компетенций по конфигурированию веб-серверов, привязке доменов и управлению правами доступа к файловым ресурсам.
- **Модуль «Работа с данными и СУБД»:** Оценка навыков администрирования баз данных, настройки подключений и организации хранения информации.
- **Модуль «Эксплуатация и поддержка»:** Практические задания по созданию бэкапов, обновлению систем и оптимизации производительности серверов.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** принципы работы протоколов HTTP/HTTPS и DNS; архитектуру серверных операционных систем; методы оптимизации и защиты веб-серверов.
- **Уметь:** устанавливать и настраивать серверное ПО; автоматизировать процессы резервного копирования; анализировать журналы событий для поиска ошибок.
- **Владеть:** навыками работы в консоли (CLI) серверных ОС; методиками настройки виртуальных хостов; инструментами мониторинга нагрузки на систему.

5. Методические рекомендации по оцениванию

Оценивание результатов обучения проводится по следующим критериям:

- **Теоретическая часть:** Проверяется понимание взаимосвязи между компонентами информационной системы и знание регламентов обслуживания.
- **Практическая часть:** Оценивается по критерию доступности и стабильности настроенного ресурса. Ошибки, приводящие к невозможности доступа к сайту или потере данных (отсутствие бэкапов), считаются критическими и существенно снижают итоговый балл.

Тема 1.1. Установка прикладного программного обеспечения и модулей информационных ресурсов, включая их настройку

Краткое изложение планов хода лекционных занятий:

1. Понятие информационных ресурсов. Роль администратора. Задачи сопровождения и поддержки

О чем занятие: Введение в сферу эксплуатации информационных систем. Определение границ ответственности администратора и стандарты качества сервиса.

Ход занятия:

- **Информационный ресурс как объект управления:** Совокупность данных, программного обеспечения и аппаратной части.
- **Профиль компетенций администратора:** От мониторинга состояния до управления обновлениями и безопасностью.
- **Задачи сопровождения:** Регламентные работы, техническая поддержка пользователей и управление инцидентами.
- **SLA (Service Level Agreement):** Понятие уровня доступности ресурса и ответственности за простои.

2. Типы архитектур (клиент-сервер, веб-архитектура). Компоненты информационных ресурсов

О чем занятие: Разбор принципов построения современных распределенных систем. Логика взаимодействия между компонентами.

Ход занятия:

- **Архитектура «Клиент-Сервер»:** Разделение ролей. Двухзвенная и трехзвенная (3-tier) архитектуры.
- **Специфика веб-архитектуры:** Протоколы передачи данных, роль фронтенда и бэкенда.
- **Основные компоненты:** Уровень представления, уровень бизнес-логики и уровень хранения данных.
- **Масштабируемость:** Вертикальное и горизонтальное расширение систем.

3. Серверы, домены, хостинг, DNS, базы данных

О чем занятие: Инфраструктурный фундамент веб-ресурса. Порядок регистрации, размещения и адресации в сети.

Ход занятия:

- **Хостинг:** Виды размещения (Shared, VPS, Dedicated, Cloud) и их различия.
- **Доменная система имен (DNS):** Иерархия доменов, типы записей (A, CNAME, MX) и процесс резолвинга.

- **Серверное оборудование:** Специфика серверного железа в отличие от персональных компьютеров.
- **Роль СУБД:** Обеспечение структурированного хранения и быстрого доступа к информации.

4. Назначение серверных ОС. Linux и Windows Server

О чем занятие: Обзор операционных систем, предназначенных для работы в режиме 24/7 под высокой нагрузкой.

Ход занятия:

- **Отличия серверных ОС:** Отсутствие лишнего графического интерфейса (в Linux), приоритет сетевых служб, многопользовательская среда.
- **Linux (Debian/Ubuntu/CentOS):** Преимущества открытого кода, стабильность и работа через CLI (командную строку).
- **Windows Server:** Особенности лицензирования, роль Active Directory и интеграция с корпоративными сервисами Microsoft.
- **Критерии выбора ОС:** В зависимости от стека технологий (PHP/Python против .NET).

5. Подготовка сервера и установка служб

О чем занятие: Первичная настройка «чистой» операционной системы для выполнения конкретных бизнес-задач.

Ход занятия:

- **Базовая настройка:** Настройка сети, создание системных пользователей и разграничение прав (sudo).
- **Безопасность «из коробки»:** Отключение неиспользуемых портов и базовая настройка брандмауэра.
- **Управление пакетами:** Репозитории, установка системных зависимостей и библиотек.
- **Обновление системы:** Важность патчей безопасности и стратегии их установки.

6. Назначение веб-сервера. Конфигурация Apache/Nginx

О чем занятие: Изучение программного обеспечения, которое обрабатывает HTTP-запросы и отдает контент пользователю.

Ход занятия:

- **Принцип работы веб-сервера:** Обработка входящих соединений и проксирование запросов.
- **Apache HTTP Server:** Модульная структура, файлы `.htaccess` и гибкость настройки.
- **Nginx:** Высокая производительность, работа с асинхронными событиями и роль Reverse Proxy.

- **Сравнительный анализ:** В каких случаях эффективен Apache, а когда необходим Nginx (или их связка).

Краткое изложение планов хода практических занятий и примерные задания:

1. Анализ структуры веб-сервера и сервисов

Ход занятия: Изучение состава программного стека сервера. Работа с системными утилитами для просмотра запущенных процессов и служб. Определение портов, которые занимают активные сервисы.

Задание: Составить схему запущенных на сервере служб, указав их назначение и используемые сетевые порты.

2. Анализ архитектуры существующих веб-ресурсов

Ход занятия: Исследование взаимодействия между фронтендом, бэкендом и базой данных на примере реального сайта. Изучение заголовков ответов сервера для определения используемых технологий.

Задание: Используя инструменты разработчика в браузере, определить тип веб-сервера (Nginx/Apache) и версию серверного языка для трех заданных ресурсов.

3. Установка и начальная настройка сервера

Ход занятия: Развертывание операционной системы (Linux/Windows Server). Настройка сетевого интерфейса, статического IP-адреса и создание административного пользователя с ограниченными правами доступа. **Задание:** Установить серверную ОС, обновить пакеты до актуальных версий и настроить синхронизацию системного времени.

4. Установка и запуск Apache/Nginx

Ход занятия: Установка пакетов веб-сервера из репозиториев. Запуск службы и проверка её статуса. Разбор структуры конфигурационных файлов (nginx.conf или apache2.conf).

Задание: Установить Nginx или Apache, запустить службу и подтвердить работоспособность, открыв стандартную заглушку в браузере по IP-адресу сервера.

5. Конфигурация доменов на сервере

Ход занятия: Настройка виртуальных хостов (Virtual Hosts) или блоков сервера (Server Blocks). Привязка нескольких доменных имен к одному IP-адресу.

Задание: Создать конфигурационный файл для домена `test.local` и настроить сервер так, чтобы он отдавал уникальную страницу для этого домена.

6. Настройка прав доступа

Ход занятия: Изучение прав владения (`chown`) и разрешений (`chmod`) в файловой системе. Настройка прав для пользователя, под которым работает веб-сервер (например, `www-data`).

Задание: Установить права доступа на корневую папку сайта так, чтобы веб-сервер мог только читать файлы, а запись была разрешена только владельцу.

7. Управление доступом к ресурсам

Ход занятия: Ограничение доступа к специфическим папкам и файлам на уровне конфигурации веб-сервера. Создание запароленных разделов (Basic Auth).

Задание: Сконфигурировать веб-сервер таким образом, чтобы доступ к папке `/admin` был возможен только после ввода логина и пароля.

8. Настройка соединения

Ход занятия: Настройка проксирования запросов и перенаправлений (Redirects). Работа с параметрами Keep-Alive и лимитами соединений для оптимизации нагрузки.

Задание: Настроить автоматический редирект с протокола HTTP на HTTPS (при наличии сертификата) или с адреса `www` на адрес без `www`.

9. Использование инструментов мониторинга

Ход занятия: Знакомство с консольными мониторами ресурсов (`htop`, `nload`) и специализированными модулями веб-серверов (например, `stub_status`). Анализ логов доступа в реальном времени.

Задание: Настроить отображение страницы статистики веб-сервера и зафиксировать пиковую нагрузку на процессор при имитации множественных запросов.

Тема 1.2. Обработка запросов заказчика в службе технической поддержки в соответствии с трудовым заданием

Краткое изложение планов хода лекционных занятий:

1. Организация хранения данных

О чем занятие: Принципы размещения информации на сервере. Выбор файловых систем и структур хранения для обеспечения быстрого доступа и надежности.

Ход занятия:

- **Файловые системы:** Обзор ext4, XFS, NTFS; их особенности и области применения.
- **Структура хранения:** Иерархия директорий в серверных ОС (стандарт FHS в Linux).
- **Сетевые хранилища:** Понятие NAS и SAN, протоколы NFS и iSCSI.
- **Технологии RAID:** Уровни 0, 1, 5, 10 как способ обеспечения отказоустойчивости и скорости дисковой подсистемы.

2. Подключение и администрирование БД

О чем занятие: Роль систем управления базами данных (СУБД) в работе веб-ресурса. Основы установки и первичной настройки безопасности.

Ход занятия:

- **Выбор СУБД:** Реляционные (MySQL, PostgreSQL) и нереляционные (MongoDB, Redis) системы.
- **Инсталляция и конфигурация:** Оптимизация параметров памяти и лимитов соединений.
- **Управление доступом:** Создание пользователей БД и назначение привилегий (GRANT/REVOKE).
- **Сетевой доступ:** Безопасная настройка прослушивания интерфейсов и работа через сокеты.

3. Контроль загрузки системы и сервисов

О чем занятие: Инструменты и методики мониторинга текущего состояния аппаратных и программных ресурсов сервера.

Ход занятия:

- **Основные показатели (Метрики):** CPU Load, использование RAM, дисковый ввод-вывод (I/O) и сетевая активность.
- **Процессы и службы:** Отслеживание статуса критических сервисов (systemd, службы Windows).
- **Пороговые значения:** Определение критических уровней нагрузки, требующих вмешательства администратора.
- **Инструменты визуализации:** Краткий обзор систем мониторинга (Zabbix, Prometheus, Grafana).

4. Анализ лог-файлов

О чем занятие: Изучение журналов регистрации событий как главного инструмента диагностики проблем и аудита безопасности.

Ход занятия:

- **Типы логов:** Системные журналы, логи веб-сервера (access и error), логи баз данных.
- **Инструменты анализа:** Использование утилит `grep`, `tail`, `awk` для поиска ошибок.
- **Ротация логов:** Настройка `logrotate` для предотвращения переполнения дискового пространства.
- **Централизованный сбор:** Понятие ELK-стека (Elasticsearch, Logstash, Kibana) для анализа больших объемов журналов.

5. Методы резервного копирования данных

О чем занятие: Создание стратегии защиты данных от потери. Классификация методов и выбор инструментов бэкапа.

Ход занятия:

- **Виды бэкапов:** Полный (Full), инкрементальный и дифференциальный.
- **Объекты копирования:** Файлы конфигурации, дампы баз данных, пользовательский контент.
- **Правило «3-2-1»:** Классическая схема хранения резервных копий.
- **Автоматизация:** Использование планировщиков (cron) и специализированного ПО.

6. Методы восстановления системы

О чем занятие: Алгоритмы действий при сбоях. Проверка жизнеспособности резервных копий и минимизация времени простоя (RTO).

Ход занятия:

- **Сценарии восстановления:** Откат отдельных файлов, восстановление базы данных или полное развертывание системы из образа.
- **Тестирование бэкапов:** Почему наличие копии не гарантирует успешное восстановление.
- **Аварийные режимы:** Загрузка в режиме Single User Mode или использование Live-дисков для восстановления загрузчика и ФС.
- **План DRP:** Создание документации по аварийному восстановлению (Disaster Recovery Plan).

7. Методы повышения производительности

О чем занятие: Оптимизация работы сервера для обработки большего количества запросов при текущих ресурсах (Тюнинг).

Ход занятия:

- **Кэширование:** Настройка OPcache, использование Nginx как кэширующего прокси, объектное кэширование (Redis).
- **Оптимизация веб-сервера:** Сжатие контента (Gzip/Brotli), настройка Keep-Alive и пулов соединений.
- **Оптимизация БД:** Индексация таблиц и профилирование медленных запросов.
- **Балансировка нагрузки:** Распределение трафика между несколькими серверами.

8. Плановое обслуживание и обновление

О чем занятие: Регламентные работы по поддержанию стабильности и безопасности системы. Управление жизненным циклом ПО.

Ход занятия:

- **Стратегия обновлений:** Различия между патчами безопасности и обновлением версий (Upgrade).
- **Тестовые среды:** Использование Stage-серверов для проверки обновлений перед внедрением в Production.
- **Очистка системы:** Удаление временных файлов, старых ядер ОС и неиспользуемых пакетов.
- **Документирование изменений:** Ведение журнала администрирования (Change Log).

Краткое изложение планов хода практических занятий и примерные задания:

1. Поиск и устранение ошибок

Ход занятия: Изучение методологии поиска неисправностей (Troubleshooting). Работа с лог-файлами веб-сервера (`error.log`) и системными журналами. Анализ кодов ответов HTTP (4xx, 5xx) для локализации проблемы.

Задание: Найти и исправить три намеренно внесенные ошибки в конфигурации веб-сервера (неверный путь к корневой папке, ошибка в синтаксисе конфига и закрытый порт в брандмауэре).

2. Создание резервных копий

Ход занятия: Знакомство с инструментами архивации (`tar`, `zip`) и создания дампов баз данных (`mysqldump` или `pg_dump`). Настройка автоматического выполнения задач по расписанию через планировщик `cron`.

Задание: Создать скрипт, который архивирует содержимое сайта и делает дамп базы данных, сохраняя результат в отдельную директорию с указанием даты в названии файла.

3. Восстановление данных из резервной копии

Ход занятия: Отработка алгоритмов восстановления системы после сбоя. Развертывание данных из созданных ранее архивов и дампов. Проверка целостности восстановленной информации.

Задание: Имитировать удаление базы данных и части файлов сайта, после чего полностью восстановить работоспособность ресурса из последней резервной копии.

4. Настройка параметров производительности

Ход занятия: Тюнинг веб-сервера и серверного окружения. Включение модулей сжатия (Gzip) и кэширования на стороне сервера. Оптимизация лимитов процессов (Worker processes/connections).

Задание: Провести нагрузочное тестирование сервера "до" и "после" включения сжатия контента и кэширования статических файлов, зафиксировав разницу в скорости отдачи страниц.

5. Установка обновлений

Ход занятия: Работа с менеджерами пакетов (`apt`, `yum` или `Server Manager`). Разбор разницы между обновлением списка пакетов, обновлением безопасности и полным апгрейдом системы.

Задание: Проверить систему на наличие критических патчей безопасности, выполнить их установку и убедиться, что установленные службы (Nginx/Apache/СУБД) корректно перезапустились.

6. Администрирование и тестирование системы

Ход занятия: Итоговое комплексное тестирование настроенной инфраструктуры. Проверка доступности, безопасности и корректности взаимодействия всех компонентов (ОС — Веб-сервер — БД).

Задание: Провести полный технический аудит развернутого информационного ресурса по чеклисту (доступность по домену, работа SSL, наличие бэкапов в расписании, отсутствие лишних открытых портов).

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МДКн.02.02 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ РЕСУРСОВ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.07 «Информационные системы и программирование»** и предназначен для проведения текущего контроля и промежуточной аттестации по дисциплине **МДК.02.02. «Обеспечение безопасности информационных ресурсов»**.

ФОС представляет собой комплекс теоретических вопросов и практических кейсов, имитирующих реальные сценарии работы специалиста по информационной безопасности, системного администратора и технического аудитора.

2. Цель и задачи курса

Цель: Подготовка специалиста, способного организовать комплексную защиту информационных ресурсов, минимизировать риски утечки данных и обеспечить устойчивость систем к внешним и внутренним угрозам.

Задачи:

- **Изучение угроз и рисков:** Овладение методами классификации угроз, анализа уязвимостей и моделирования действий злоумышленников.
- **Применение криптографических методов:** Освоение принципов шифрования, хеширования и использования электронной подписи для обеспечения конфиденциальности и подлинности данных.
- **Администрирование средств защиты:** Настройка межсетевых экранов (Firewalls), антивирусного ПО и систем обнаружения вторжений (IDS).
- **Управление доступом и аудитом:** Настройка прав пользователей, внедрение многофакторной аутентификации и мониторинг логов безопасности.
- **Реагирование на инциденты:** Разработка планов восстановления данных и алгоритмов действий при обнаружении признаков взлома.

3. Структура оценочных материалов

Содержание ФОС коррелирует с ключевыми уровнями защиты информационной инфраструктуры:

- **Модуль «Организационно-правовая защита»:** Проверка знаний законодательства (ФЗ-152 и др.), регламентов работы и основ политики безопасности организации.
- **Модуль «Программная и техническая защита»:** Практические задания по настройке антивирусной защиты, конфигурации firewall и поиску уязвимостей в ПО.
- **Модуль «Сетевая безопасность и криптография»:** Оценка навыков шифрования данных, настройки защищенных каналов (VPN/SSL) и использования инструментов мониторинга трафика.
- **Модуль «Аудит и реагирование»:** Анализ системных журналов, выявление подозрительной активности и проведение итоговой проверки защищенности ресурса.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** виды угроз и классификацию вредоносного ПО; основы криптографии; принципы работы межсетевых экранов; правовые нормы в области ИБ.
- **Уметь:** настраивать права доступа к ресурсам; обнаруживать попытки несанкционированного доступа; выполнять шифрование файлов; настраивать параметры безопасности ОС.
- **Владеть:** навыками работы с логами безопасности; инструментарием для поиска уязвимостей веб-ресурсов; методиками настройки антивирусных комплексов.

5. Методические рекомендации по оцениванию

Оценивание строится на принципах практико-ориентированного подхода:

- **Теоретическая часть:** Оценивается владение терминологией, понимание архитектуры защиты и знание нормативных требований.
- **Практическая часть:** Оценивается по факту корректности настройки системы защиты. Наличие критических уязвимостей или неправильное распределение прав доступа, ведущее к компрометации данных, являются основанием для неудовлетворительной оценки.

Тема 2.1. Резервное копирование и развертывание резервной копии информационных ресурсов

Краткое изложение планов хода лекционных занятий:

1. Основные понятия, цели и задачи защиты информации

О чем занятие: Фундаментальные принципы обеспечения безопасности данных. Определение роли защиты информации в современном цифровом обществе.

Ход занятия:

- **Триада CIA:** Конфиденциальность, целостность и доступность как главные цели защиты.
- **Объекты и субъекты защиты:** Что мы защищаем и от кого.
- **Законодательная база:** Краткий обзор правовых аспектов и стандартов в области защиты информации.
- **Риски и ущерб:** Понятие ценности информации и последствия её компрометации.

2. Классификация угроз информационным ресурсам

О чем занятие: Систематизация потенциально опасных воздействий на информационные системы. Понимание природы возникновения угроз.

Ход занятия:

- **Естественные и искусственные угрозы:** Стихийные бедствия против человеческого фактора.
- **Преднамеренные и непреднамеренные угрозы:** От ошибок персонала до целенаправленных хакерских атак.
- **Внутренние и внешние векторы атак:** Инсайдеры и внешние злоумышленники.
- **Типология атак:** Вредоносное ПО, социальная инженерия, сетевые атаки (DoS/DDoS, MITM).

3. Назначение политики безопасности и её основные положения

О чем занятие: Создание регламента защиты организации. Роль управленческих решений в обеспечении устойчивости информационной системы.

Ход занятия:

- **Понятие политики безопасности:** Иерархия документов (от декларации до технических инструкций).
- **Модели управления доступом:** Избирательное (DAC), ролевое (RBAC) и мандатное (MAC) управление.
- **Идентификация, аутентификация и авторизация:** Различия и взаимосвязь процессов.
- **Аудит и мониторинг:** Необходимость протоколирования действий пользователей для расследования инцидентов.

4. Организационные, программные и технические методы защиты

О чем занятие: Эшелонированная оборона. Разбор инструментов и методов, применяемых на разных уровнях защиты системы.

Ход занятия:

- **Организационные меры:** Физический доступ, работа с кадрами, регламенты утилизации носителей.
- **Технические средства:** Межсетевые экраны (Firewalls), системы обнаружения вторжений (IDS/IPS).
- **Программные методы:** Антивирусная защита, контроль целостности файлов, использование песочниц (sandboxing).
- **Комплексный подход:** Создание системы защиты «в глубину» (Defense in Depth).

5. Основы криптографии и методы шифрования

О чем занятие: Математические методы обеспечения конфиденциальности и подлинности данных. Применение криптографии в современных протоколах.

Ход занятия:

- **Симметричное шифрование:** Принципы работы с общим ключом (AES, DES), преимущества и недостатки.
- **Асимметричное шифрование:** Пара ключей (публичный/приватный) и алгоритм RSA.
- **Хеширование:** Обеспечение целостности данных и безопасное хранение паролей.
- **Электронная подпись (ЭЦП):** Механизм подтверждения авторства и неизменности документа.

Краткое изложение планов хода практических занятий и примерные задания:

1. Определение возможных угроз информационным ресурсам

Ход занятия: Изучение методики моделирования угроз. Анализ конкретной информационной системы (например, корпоративной почты) на наличие уязвимостей. Классификация рисков по степени вероятности и тяжести последствий.

Задание: Составить реестр потенциальных угроз для локальной сети учебного кабинета, разделив их на внешние и внутренние.

2. Использование программ для шифрования информации

Ход занятия: Практическая работа с ПО для обеспечения конфиденциальности. Изучение принципов симметричного и асимметричного шифрования. Генерация ключей и создание зашифрованных контейнеров.

Задание: Зашифровать текстовый файл с использованием свободно распространяемого ПО (например, VeraCrypt или встроенных средств ОС) и передать ключ доступа преподавателю.

3. Выявление угроз в компьютерных сетях

Ход занятия: Работа с инструментами анализа сетевого трафика (снифферами). Мониторинг активных соединений и выявление подозрительной активности (сканирование портов, попытки подбора паролей).

Задание: С помощью утилиты анализа сети (например, Wireshark) обнаружить в трафике передачу данных по незащищенным протоколам (HTTP, FTP) и зафиксировать факт утечки информации.

4. Конфигурирование firewall

Ход занятия: Изучение принципов фильтрации пакетов. Настройка правил входящего и исходящего трафика. Работа с «белыми» и «черными» списками IP-адресов и портов.

Задание: Сконфигурировать брандмауэр операционной системы так, чтобы запретить все входящие соединения, кроме HTTP/HTTPS и специфического порта для удаленного управления.

Тема 2.2. Настройка прав пользователей в соответствии с функциональными задачами (ролями)

Краткое изложение планов хода лекционных занятий:

1. Назначение, виды и применение электронной подписи

О чем занятие: Электронная подпись (ЭП) как юридический и технический эквивалент ручной подписи. Обеспечение авторства и неизменности электронных документов.

Ход занятия:

- **Криптографическая основа:** Использование асимметричного шифрования (закрытый и открытый ключи).
- **Виды ЭП:** Простая, усиленная неквалифицированная и усиленная квалифицированная подписи; их различия и юридическая сила.
- **Инфраструктура открытых ключей (PKI):** Роль удостоверяющих центров в выдаче сертификатов.
- **Практическое применение:** Электронный документооборот, государственные услуги и банковские операции.

2. Основные угрозы в компьютерных сетях

О чем занятие: Классификация опасностей, возникающих при передаче данных. Анализ векторов атак на сетевую инфраструктуру.

Ход занятия:

- **Сетевая разведка:** Сбор информации, сканирование портов и анализ трафика (сниффинг).
- **Атаки на доступность:** Механизмы реализации DoS и DDoS атак; переполнение буфера.
- **Атаки на перехват и подмену:** Человек посередине (MITM), IP-спуфинг и DNS-спуфинг.
- **Угрозы беспроводных сетей:** Уязвимости протоколов Wi-Fi и методы несанкционированного доступа.

3. Назначение и принцип работы firewall

О чем занятие: Межсетевой экран как основной барьер между защищенной локальной сетью и внешней средой. Принципы фильтрации трафика.

Ход занятия:

- **Эшелонированная защита:** Место брандмауэра в архитектуре безопасности.
- **Уровни фильтрации:** Пакетные фильтры (L3), шлюзы сеансового уровня и прикладные шлюзы (WAF).
- **Методы анализа:** Stateful Inspection (анализ состояния соединений) против статической фильтрации.
- **Политики доступа:** Формирование правил по принципу «запрещено все, что не разрешено».

4. Виды вредоносных программ и методы защиты

О чем занятие: Классификация вредоносного ПО (Malware) по способам распространения и деструктивному воздействию.

Ход занятия:

- **Типология:** Вирусы, черви, троянские программы, шпионское ПО (Spyware) и программы-вымогатели (Ransomware).
- **Каналы заражения:** Съемные носители, почтовый фишинг, уязвимости ПО и социальная инженерия.
- **Методы обнаружения:** Сигнатурный анализ, эвристическое сканирование и использование «песочниц».
- **Профилактика:** Обновление систем, использование антивирусов и обучение пользователей кибергигиене.

5. Типы уязвимостей веб-ресурсов

О чем занятие: Анализ слабых мест веб-приложений, позволяющих злоумышленникам получить доступ к данным или серверу.

Ход занятия:

- **Классификация OWASP Top 10:** Обзор наиболее критичных рисков безопасности веб-приложений.
- **Инъекции:** SQL-инъекции, внедрение команд ОС и их влияние на базу данных.
- **Межсайтовый скриптинг (XSS):** Выполнение вредоносного кода в браузере пользователя.
- **Проблемы аутентификации:** Небезопасное хранение паролей и управление сессиями.

Краткое изложение планов хода практических занятий и примерные задания:

1. Поиск уязвимостей веб-сайта

Ход занятия: Знакомство с инструментами анализа безопасности (сканеры уязвимостей). Проверка веб-ресурса на наличие типичных ошибок: слабые пароли, открытые директории, отсутствие SSL-сертификата. Работа с вкладкой "Network" и "Security" в инструментах разработчика браузера.

Задание: Провести аудит учебного сайта на наличие SQL-инъекций или XSS-уязвимостей в полях ввода, используя специализированное ПО или ручное тестирование запросов.

2. Настройка базовых механизмов защиты

Ход занятия: Реализация политики безопасности на уровне операционной системы и веб-сервера. Настройка прав доступа к файлам и папкам. Установка запретов на запуск исполняемых файлов из недоверенных источников.

Задание: Настроить двухфакторную аутентификацию (2FA) для учетной записи и сконфигурировать политику сложности паролей (минимальная длина, спецсимволы) в системных настройках.

3. Использование средств мониторинга

Ход занятия: Работа с системными журналами событий (Event Viewer) и утилитами мониторинга ресурсов. Настройка оповещений о критических событиях безопасности: неудачные попытки входа, изменение системных файлов.

Задание: Настроить мониторинг сетевых соединений и выявить процессы, потребляющие аномально большой объем трафика или использующие нестандартные порты.

4. Установка и настройка антивируса

Ход занятия: Выбор и инсталляция антивирусного ПО. Сравнение различных режимов сканирования (быстрое, полное, выборочное). Настройка исключений и правил карантина. Обновление вирусных баз.

Задание: Провести полную проверку системы, настроить расписание автоматического сканирования и создать загрузочный USB-носитель для восстановления системы после заражения.

Тема 2.3. Применение программных средств обеспечения безопасности информации веб-приложений

Краткое изложение планов хода лекционных занятий:

1. Основные методы защиты веб-ресурсов

О чем занятие: Комплексный подход к обеспечению безопасности веб-приложений и серверов. Защита от наиболее распространенных векторов атак.

Ход занятия:

- **Защита на уровне протокола:** Роль SSL/TLS сертификатов и переход на защищенное соединение HTTPS.
- **Фильтрация входящего трафика:** Назначение и принцип работы Web Application Firewall (WAF).
- **Безопасная разработка:** Принципы валидации входных данных и экранирования спецсимволов для предотвращения инъекций.
- **Защита от DDoS-атак:** Методы кэширования, распределения нагрузки и использования CDN-сервисов.

2. Методы идентификации пользователей

О чем занятие: Механизмы распознавания субъектов в системе и проверка их полномочий. Разграничение понятий идентификации, аутентификации и авторизации.

Ход занятия:

- **Факторы аутентификации:** Что пользователь знает (пароль), что имеет (токен) и кем является (биометрия).
- **Многофакторная аутентификация (MFA):** Преимущества использования дополнительных кодов и push-уведомлений.
- **Технологии Single Sign-On (SSO):** Обеспечение сквозного доступа к различным сервисам через единую учетную запись.
- **Безопасное хранение учетных данных:** Использование соленых хешей (salted hashes) вместо хранения паролей в открытом виде.

3. Средства мониторинга информационных систем

О чем занятие: Методы непрерывного наблюдения за состоянием ИТ-инфраструктуры для своевременного обнаружения аномалий.

Ход занятия:

- **Сбор и анализ логов:** Роль систем логирования (Syslog) в фиксации действий пользователей и системных событий.
- **SIEM-системы:** Понятие Security Information and Event Management как централизованного узла анализа безопасности.
- **Мониторинг доступности и производительности:** Использование метрик для выявления скрытых атак (например, скрытого майнинга или сканирования сети).
- **Визуализация данных:** Использование дашбордов для оперативного контроля состояния информационной системы.

4. Выявление и реагирование на инциденты

О чем занятие: Алгоритм действий при обнаружении признаков взлома или утечки данных. Процесс минимизации ущерба и восстановления системы.

Ход занятия:

- **Жизненный цикл инцидента:** Обнаружение, локализация, устранение последствий и анализ причин.
- **Классификация инцидентов:** Определение приоритетов реагирования в зависимости от критичности ресурса.
- **Планы аварийного восстановления (DRP):** Регламенты работы персонала в условиях компрометации системы.
- **Постинцидентный анализ:** Сбор доказательной базы и внесение изменений в систему защиты для предотвращения повторных атак.

Краткое изложение планов хода практических занятий и примерные задания:

1. Настройка прав доступа

Ход занятия: Изучение моделей управления доступом. Работа с правами файловой системы (чтение, запись, выполнение) для различных групп пользователей. Настройка принципа «наименьших привилегий» в операционной системе.

Задание: Создать структуру папок для отдела, распределив права доступа так, чтобы рядовые сотрудники могли только читать документы, а руководители — редактировать их.

2. Работа с логами безопасности

Ход занятия: Знакомство с системными журналами (Event Viewer в Windows или /var/log в Linux). Изучение структуры записей: тип события, дата, инициатор, результат. Настройка фильтров для поиска критических ошибок.

Задание: Проанализировать журнал событий и найти записи о пяти неудачных попытках входа в систему, зафиксировав время и имена учетных записей.

3. Выявление подозрительной активности

Ход занятия: Мониторинг запущенных процессов и сетевых соединений в реальном времени. Использование диспетчера задач и консольных утилит (например, `netstat`) для обнаружения скрытой активности программ.

Задание: Запустить сетевой анализатор и выявить процесс, который отправляет пакеты на неизвестный внешний IP-адрес при отсутствии открытого браузера.

4. Проверка защищенности информационного ресурса

Ход занятия: Проведение комплексного аудита безопасности. Использование чеклистов для проверки парольной политики, наличия обновлений ПО и активности межсетевого экрана. Имитация простых векторов атак.

Задание: Провести итоговую проверку настроенного рабочего места на соответствие требованиям безопасности и составить краткий отчет о найденных «слабых местах».

Примерные вопросы к экзамену по МДК 02.01 Настройка и сопровождение информационных ресурсов

1. Дайте определение информационного ресурса и перечислите его основные компоненты.
2. Какова роль системного администратора при сопровождении информационных ресурсов?
3. Что такое архитектура «клиент-сервер»? Перечислите её основные особенности.
4. Чем отличается двухзвенная архитектура от трехзвенной?
5. Какие основные компоненты входят в структуру веб-приложения?
6. Что такое хостинг? Назовите основные виды хостинга.
7. Объясните принцип работы системы доменных имен DNS.
8. Какие типы DNS-записей вы знаете?
9. Какие функции выполняют серверные операционные системы?
10. Назовите основные различия Linux и Windows Server.
11. Что включает процесс подготовки сервера перед установкой веб-приложения?
12. Назовите основные функции веб-сервера.
13. Чем отличаются веб-серверы Apache и Nginx?
14. Что такое виртуальный хост (Virtual Host) и для чего он используется?

15. Что такое права доступа в файловой системе?
16. Какие типы прав доступа существуют в Linux?
17. Для чего используются команды chmod и chown?
18. Какие основные инструменты используются для мониторинга сервера?
19. Что такое лог-файл и для чего он используется?
20. Какие виды логов существуют в веб-сервере?
21. Назовите основные методы резервного копирования данных.
22. Чем отличается полный бэкап от инкрементального?
23. Что такое правило резервного копирования 3-2-1?
24. Какие инструменты используются для создания резервных копий?
25. Опишите основные этапы восстановления данных из резервной копии.
26. Какие основные показатели используются при мониторинге производительности сервера?
27. Что такое оптимизация производительности веб-сервера?
28. Какие методы используются для повышения производительности веб-ресурсов?
29. Что включает плановое обслуживание информационных ресурсов?
30. Как осуществляется обработка запросов пользователей в службе технической поддержки?

Примерные задания к экзамену по МДК 02.01 Настройка и сопровождение информационных ресурсов

1. Установить серверную операционную систему Linux и выполнить обновление пакетов.
2. Настроить статический IP-адрес сервера.
3. Установить веб-сервер Nginx или Apache и проверить его работоспособность через браузер.
4. Создать виртуальный хост для домена test.local и настроить отображение отдельной страницы.
5. Настроить перенаправление с HTTP на HTTPS.

6. Настроить права доступа к папке сайта так, чтобы веб-сервер имел только право чтения.
7. Создать нового пользователя системы и назначить ему ограниченные права администратора.
8. Настроить доступ к папке /admin только по логину и паролю.
9. Проанализировать структуру конфигурационного файла веб-сервера и описать его основные параметры.
10. Определить, какие сервисы запущены на сервере и какие порты они используют.
11. Настроить подключение к базе данных MySQL или PostgreSQL.
12. Создать пользователя базы данных и назначить ему необходимые права.
13. Найти ошибку в конфигурации веб-сервера и исправить её.
14. Проанализировать файл error.log и определить причину ошибки.
15. Создать архив сайта с помощью команды tar.
16. Создать дамп базы данных с помощью mysqldump или аналогичного инструмента.
17. Настроить автоматическое резервное копирование сайта с использованием cron.
18. Выполнить восстановление сайта из резервной копии.
19. Настроить мониторинг загрузки сервера с помощью утилит top, htop или аналогов.
20. Определить процессы, потребляющие наибольшее количество ресурсов.
21. Настроить сжатие статических файлов на веб-сервере.
22. Настроить кэширование статического контента.
23. Провести анализ HTTP-запросов сайта с помощью инструментов разработчика браузера.
24. Определить используемый веб-сервер и технологический стек сайта.
25. Проверить сервер на наличие открытых портов.
26. Настроить автоматическое обновление пакетов системы.

27. Выполнить аудит доступности веб-ресурса по чек-листу администратора.
28. Провести диагностику проблем доступа к сайту и устранить причину ошибки.
29. Составить план резервного копирования для веб-ресурса.
30. Подготовить краткий технический отчет по администрированию информационного ресурса.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕУЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

Рабочая программа профессионального модуля

ПМн.03 «Разработка веб приложения на стороне клиента»

для специальности

09.02.09 «Веб-разработка»

Екатеринбург

2026

Рабочая программа ПМн.03 «Разработка веб приложения на стороне клиента» разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.09 «Веб-разработка».

Организация-разработчик: ГАПОУ СО «Екатеринбургский монтажный колледж»

Разработчик Софьина Н.А. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК; Кузьмин В.П. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», 1КК;,

Рассмотрена методическим объединением Информационных технологий

Руководитель МО _____Софьина Н.А.

УТВЕРЖДАЮ

Заместитель директора по учебной работе

_____ Л.С.Хоринова

«13» марта 2026

СОДЕРЖАНИЕ

1. **ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
2. **СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
3. **УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
4. **КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

1. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

«ПМн.03 Разработка веб приложения на стороне клиента»

1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля, обучающихся должен освоить основной вид деятельности разработки веб приложения на стороне клиента и соответствующие ему общие компетенции, и профессиональные компетенции:

1.1.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности;
ОК 03.	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;
ОК 04.	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05.	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 06.	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
ОК 07.	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
ОК 08.	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках.

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ВД 3	Разработка веб приложения на стороне клиента
ПК 3.1.	Проектировать структуры разделов информационных ресурсов с целью создания эскиза и прототипа интерфейса пользователя.
ПК 3.2.	Разрабатывать интерфейс пользователя для информационных ресурсов с использованием стандартов в области веб-разработки.
ПК 3.3.	Создавать структуру кода веб-страницы информационных ресурсов в соответствии с дизайн-макетом.
ПК 3.4.	Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков.

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Владеть навыками	− разработки эскизов, схем, прототипов интерфейса пользователя информационного ресурса; − разработки дизайна компонентов интерфейса пользователя в соответствии со стандартами и требованиями заказчика; − разработки программного кода веб-страниц информационного ресурса, в том числе с использованием готовых технических решений; − разработки кроссбраузерной верстки веб-страниц информационного ресурса; − разработки клиентской части web-приложения в соответствии с техническим заданием (спецификацией).
Уметь	− применять программные средства для разработки интерфейса; − применять инструменты для оценки эффективности и удобства созданного интерфейса; − применять полученные данные для оптимизации интерфейса; − применять нормативные документы, определяющие требования к оформлению интерфейсов информационных ресурсов; − создавать адаптивный интерфейс web-ресурса; − применять специализированное программное обеспечение для верстки страниц информационных ресурсов; − использовать язык разметки страниц информационных ресурсов; − применять выбранные языки программирования для написания программного кода; − использовать выбранную среду программирования и средства системы управления базами данных; − использовать возможности имеющейся программной архитектуры информационного ресурса.

Знать	− современные принципы построения интерфейсов пользователя; − основные требования, предъявляемые к дизайну графических интерфейсов; − способы представления информации с учетом особенностей пользователя: возрастных, особенностей ограниченных возможностей здоровья и др.; − особенности отображения элементов интерфейсов web-ресурсов в различных браузерах; − правила реализации адаптивного интерфейса web-ресурса; − методы повышения читаемости программного кода; − синтаксис выбранного языка программирования, особенностей программирования на этом языке, стандартных библиотек языка программирования; − отраслевую нормативную техническую документацию; − особенности выбранной среды программирования; − компоненты программно-технических архитектур информационных ресурсов, существующих приложений и интерфейсов взаимодействия с ними; − сетевые протоколы и основы web-технологий; − современные стандарты взаимодействия компонентов распределенных приложений; − программные средства и платформы для разработки web-ресурсов; − основы информационной безопасности web-ресурсов.

1.2. Количество часов, отводимое на освоение профессионального модуля

Всего часов: 506;

в том числе в форме практической подготовки: 158 часов;

Из них:

- на освоение МДКн.03.01. Проектирование и дизайн интерфейсов: 102 часа;

в том числе в форме практической подготовки: 34 часов

- на освоение МДКн.03.02. Верстка страниц: 114 часа;

в том числе в форме практической подготовки: 70 часов;

- на освоение МДКн.03.03. Разработка клиентской части информационных ресурсов: 110 часов;

в том числе в форме практической подготовки: 54 часов;

практики, в том числе учебная: 36 часов;

производственная: 144 часов.

Промежуточная аттестация _____.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. Структура профессионального модуля

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля	Всего, час.	В т.ч. в форме практической. подготовки	Объем профессионального модуля, ак. час.						
				Обучение по МДК					Практики	
				Всего	В том числе					
					Лабораторных. и практических. занятий	Курсовых работ (проектов)	Самостоятельная работа	Промежуточная аттестация	Учебная	Производственная
1	2	3	4	5	6	7	8	9	10	11
ПК 3.1 ПК 3.2 ОК 1-9	Раздел 1. Проектирование и дизайн интерфейсов	**102**	34	**102**	34	0	0	0	**36**	**144**
ПК 3.2. ПК 3.3. ОК 1-9	Раздел 2. Верстка страниц	**114**	70	**114**	70		0			
ПК 3.4. ОК 1-9	Раздел 3. Разработка клиентской части информационных ресурсов	**110**	54	**110**	54	30	0			
	Учебная практика	**36**	36						**36**	

	Производственная практика (по профилю специальности), часов	**144**	**144**							**144**
	Промежуточная аттестация	**0**	0							
	Всего:	**506**	**338**	**326**	**158**	**30**	**0**	**0**	**36**	**144**

2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовой проект	Объем, акад. ч / в том числе в форме практической подготовки, акад ч
1	2	3
Раздел 1. Проектирование и дизайн интерфейсов		
МДКн.03.01. Проектирование и дизайн интерфейсов		
Тема 1.1. Проектирование и дизайн интерфейсов	**Содержание**	**20**
	Понятие пользовательского интерфейса. Роль UI/UX в разработке веб-приложений. Основные этапы проектирования интерфейса.	**2**
	Принципы пользовательского опыта. Потребности пользователей. Понятие usability.	**2**
	Законы юзабилити. Правила проектирования удобных интерфейсов. Типичные ошибки интерфейсов.	**2**
	Структура сайта. Навигация и организация контента. Карта сайта.	**2**
	Баланс, контраст, выравнивание, повторение. Визуальная иерархия элементов.	**2**
	Цветовые модели. Цветовые схемы. Психология цвета в веб-дизайне.	**2**
	Шрифты, читаемость текста, правила использования типографики.	**2**
	Понятие адаптивного и responsive дизайна. Подходы Mobile First.	**2**
	Низкоуровневые и высокоуровневые прототипы. Инструменты прототипирования.	**2**

	UI-паттерны, минимализм, микроанимации, дизайн-системы.	**2**
	В том числе практических занятий и лабораторных работ	**34**
	Анализ интерфейсов популярных веб-сайтов с точки зрения UX и usability.	**2**
	Создание карты сайта и пользовательских сценариев.	**2**
	Разработка user flow для веб-приложения.	**2**
	Создание wireframe главной страницы сайта.	**2**
	Создание wireframe внутренних страниц сайта.	**2**
	Разработка интерактивного прототипа в Figma.	**2**
	Подбор цветовой палитры для веб-интерфейса.	**2**
	Подбор и настройка шрифтов для интерфейса.	**2**
	Разработка макета для мобильной версии сайта.	**2**
	Разработка макета страницы приложения.	**2**
	Создание визуального макета страницы в Figma.	**2**
	Создание кнопок, форм, карточек, меню.	**2**
	Формирование набора UI-компонентов.	**2**

	Создание полного прототипа веб-ресурса.	**2**
	Проведение тестирования usability разработанного интерфейса.	**2**
	Исправление выявленных ошибок интерфейса.	**2**
	Представление проекта интерфейса и его защита.	**2**
Самостоятельная работа 1. Анализ литературы и интернет-источников по UX-дизайну. 2. Самостоятельный анализ интерфейсов популярных сайтов. 3. Создание структуры сайта по заданной тематике. 4. Освоение возможностей инструмента прототипирования. 5. Создание цветовой палитры для интерфейса. 6. Подбор шрифтов для веб-интерфейса. 7. Создание каркаса страницы сайта. 8. Изучение современных тенденций веб-дизайна. 9. Подготовка материалов для защиты проекта.		**18**
Раздел 2. Верстка страниц		
МДКн.03.02 Верстка страниц		
Тема 2.1. Верстка страниц	**Содержание**	**30**
	Понятие веб-верстки. Роль HTML и CSS в разработке веб-страниц.	**2**
	Структура HTML-документа. Основные теги и элементы.	**2**
	Заголовки, абзацы, списки, форматирование текста.	**2**
	Создание гиперссылок. Работа с изображениями.	**2**

	Создание и оформление таблиц.	**2**
	Элементы форм, поля ввода, кнопки.	**2**
	Назначение CSS. Подключение таблиц стилей.	**2**
	Типы селекторов. Приоритет и каскадность.	**2**
	Margin, padding, border, width, height.	**2**
	Шрифты, размеры, межстрочные интервалы.	**2**
	Цветовые свойства и фоновые изображения.	**2**
	Основы гибкой верстки с использованием Flexbox.	**2**
	Использование CSS Grid для построения макетов.	**2**
	Медиа-запросы. Подход Mobile First.	**2**
	БЭМ-методология, кроссбраузерность, оптимизация верстки.	**2**
	В том числе практических и лабораторных занятий	**70**
	Создание базовой структуры HTML-страницы.	**2**
	Использование заголовков, абзацев и списков.	**2**
	Вставка изображений и настройка атрибутов.	**2**
	Создание внутренних и внешних ссылок.	**2**
	Разработка таблиц в HTML.	**2**

	Настройка структуры таблиц.	**2**
	Разработка формы ввода данных.	**2**
	Использование различных типов полей ввода.	**2**
	Подключение таблиц стилей к HTML-странице.	**2**
	Применение различных типов CSS-селекторов.	**2**
	Работа со шрифтами и текстовыми стилями.	**2**
	Настройка цветового оформления страницы.	**2**
	Настройка отступов и границ элементов.	**2**
	Верстка структуры страницы.	**2**
	Верстка меню сайта.	**2**
	Использование фоновых изображений.	**2**
	Верстка карточек товаров или статей.	**2**
	Создание макета страницы с Flexbox.	**2**
	Верстка блока с использованием Flexbox.	**2**
	Создание сеточного макета страницы.	**2**
	Разработка сложной сетки страницы.	**2**
	Создание header сайта.	**2**

	Создание footer сайта.	**2**
	Разработка основной части страницы.	**2**
	Верстка страницы по готовому макету.	**2**
	Продолжение верстки страницы.	**2**
	Использование медиа-запросов.	**2**
	Адаптация страницы под мобильные устройства.	**2**
	Оптимизация интерфейса для планшетов.	**2**
	Проверка отображения в различных браузерах.	**2**
	Анализ и исправление ошибок.	**2**
	Улучшение структуры CSS-кода.	**2**
	Применение методологии БЭМ.	**2**
	Завершение проекта страницы сайта.	**2**
	Представление выполненной верстки сайта.	**2**
Самостоятельная работа 1. Изучение дополнительных HTML-тегов. 2. Изучение дополнительных свойств CSS. 3. Анализ структуры и верстки популярных сайтов. 4. Изучение принципов гибкой верстки. 5. Изучение сеточной системы CSS Grid. 6. Изучение медиа-запросов. 7. Подготовка материалов для практического проекта.		14

Раздел 3. Разработка клиентской части информационных ресурсов		
МДКн.03.03 Разработка клиентской части информационных ресурсов		
Тема 3.1. Разработка клиентской части информационных ресурсов	**Содержание**	**30**
	Назначение клиентских технологий. Роль JavaScript в веб-разработке.	**2**
	Синтаксис языка. Переменные и типы данных.	**2**
	Арифметические, логические и сравнительные операторы.	**2**
	if, else, switch. Управление логикой программы.	**2**
	for, while, do...while. Итерации в программах.	**2**
	Объявление функций. Параметры и возвращаемые значения.	**2**
	Создание и обработка массивов. Основные методы.	**2**
	Создание объектов и работа со свойствами.	**2**
	Структура DOM. Доступ к элементам страницы.	**2**
	Обработчики событий. Пользовательские события.	**2**
	Обработка данных форм. Проверка данных.	**2**
	Promise, async/await.	**2**
	Получение данных с сервера. Использование Fetch API.	**2**
	LocalStorage и SessionStorage.	**2**

	Отладка, оптимизация и структура клиентского кода.	**2**
	В том числе практических и лабораторных занятий	**84**
	Подключение скриптов к веб-странице.	**2**
	Создание переменных и вывод данных.	**2**
	Работа с различными типами данных.	**2**
	Выполнение вычислений в JavaScript.	**2**
	Реализация логики с использованием if.	**2**
	Использование switch.	**2**
	Использование цикла for.	**2**
	Использование while и do...while.	**2**
	Разработка пользовательских функций.	**2**
	Использование параметров и возврата значений.	**2**
	Создание и обработка массивов.	**2**
	Использование методов массивов.	**2**
	Создание объектов.	**2**
	Изменение и получение свойств объектов.	**2**
	Получение элементов страницы.	**2**

	Изменение содержимого элементов.	**2**
	Изменение стилей элементов.	**2**
	Создание элементов страницы.	**2**
	Удаление элементов страницы.	**2**
	Обработка кликов мыши.	**2**
	Обработка событий клавиатуры.	**2**
	Создание интерактивных элементов интерфейса.	**2**
	Получение данных формы.	**2**
	Проверка данных формы.	**2**
	Реализация проверки ввода пользователя.	**2**
	Использование Promise.	**2**
	Использование async/await.	**2**
	Получение данных с сервера.	**2**
	Обработка полученных данных.	**2**
	Работа с JSON-данными.	**2**
	Сохранение данных в браузере.	**2**
	Получение и удаление данных.	**2**

	Разработка интерактивных элементов.	**2**
	Создание простого веб-виджета.	**2**
	Начало разработки клиентского приложения.	**2**
	Продолжение разработки.	**2**
	Работа с DOM и событиями.	**2**
	Добавление работы с API.	**2**
	Сохранение данных пользователя.	**2**
	Улучшение структуры кода.	**2**
	Поиск и исправление ошибок.	**2**
	Демонстрация разработанного приложения.	**2**
Самостоятельная работа 1. Изучение синтаксиса и возможностей языка. 2. Изучение методов работы с DOM. 3. Выполнение тренировочных заданий. 4. Изучение Promise и async/await. 5. Изучение получения данных из API. 6. LocalStorage и SessionStorage. 7. Подготовка итогового проекта.		**26**
Курсовой проект		
Примерная тематика курсовых проектов 1. Разработка клиентской части веб-приложения онлайн-калькулятора. 2. Разработка клиентского интерфейса системы управления задачами.		

3. Разработка клиентской части интерактивного каталога товаров. 4. Разработка клиентского приложения погодного информера с использованием API. 5. Разработка клиентской части веб-приложения заметок пользователя. 6. Разработка клиентского интерфейса системы регистрации и авторизации пользователей. 7. Разработка клиентской части веб-приложения для планирования расписания. 8. Разработка клиентского интерфейса интерактивной галереи изображений. 9. Разработка клиентской части приложения для учета личных расходов. 10. Разработка клиентского интерфейса системы поиска информации по каталогу. 11. Разработка клиентской части веб-приложения для управления списком покупок. 12. Разработка клиентского интерфейса системы онлайн-опросов и голосований. 13. Разработка клиентской части приложения просмотра новостей с использованием API. 14. Разработка клиентского интерфейса интерактивной карты сайта. 15. Разработка клиентской части веб-приложения для ведения личного дневника. 16. Разработка клиентского интерфейса системы управления контактами пользователя. 17. Разработка клиентской части веб-приложения для бронирования услуг. 18. Разработка клиентского интерфейса онлайн-таймера и системы напоминаний. 19. Разработка клиентской части интерактивного тестирования пользователей. 20. Разработка клиентского интерфейса системы фильтрации и сортировки данных. 21. Разработка клиентской части веб-приложения для учета задач команды. 22. Разработка клиентского интерфейса системы отзывов и комментариев пользователей. 23. Разработка клиентской части веб-приложения для управления списком дел. 24. Разработка клиентского интерфейса системы поиска фильмов с использованием API. 25. Разработка клиентской части приложения для просмотра статистических данных. 26. Разработка клиентского интерфейса системы управления событиями календаря. 27. Разработка клиентской части веб-приложения для создания заметок с тегами. 28. Разработка клиентского интерфейса интерактивного учебного тренажера. 29. Разработка клиентской части веб-приложения для работы со списком пользователей. 30. Разработка клиентского интерфейса системы поиска и фильтрации информации.	
Учебная практика Разработка веб приложения на стороне клиента **Виды работ** 1. Ознакомление с техническим заданием на разработку клиентской части веб-приложения. 2. Анализ предметной области и определение требований к пользовательскому интерфейсу. 3. Изучение используемых технологий разработки клиентской части веб-приложений.	**36**

4. Настройка среды разработки и инструментов программирования. 5. Проектирование структуры веб-страниц и навигации информационного ресурса. 6. Разработка прототипа пользовательского интерфейса веб-приложения. 7. Создание HTML-разметки веб-страниц. 8. Разработка стилей оформления интерфейса с использованием CSS. 9. Реализация адаптивной верстки веб-страниц. 10. Программирование интерактивных элементов интерфейса с использованием JavaScript. 11. Реализация обработки пользовательских событий (нажатие кнопок, ввод данных, отправка форм). 12. Использование библиотек и фреймворков для разработки клиентской части веб-приложения. 13. Реализация взаимодействия клиентской части с сервером (отправка и получение данных). 14. Обработка и отображение данных, полученных от серверной части приложения. 15. Реализация проверки корректности пользовательских данных (валидация форм). 16. Тестирование функциональности клиентской части веб-приложения. 17. Отладка программного кода клиентской части приложения. 18. Проверка корректности отображения интерфейса в различных веб-браузерах. 19. Оптимизация кода и улучшение пользовательского интерфейса. **20.** Подготовка отчёта по результатам выполнения учебной практики.	
Производственная практика **Виды работ** 1. Ознакомление со структурой организации, правилами внутреннего распорядка и техникой безопасности на предприятии. 2. Изучение используемых в организации технологий разработки веб-приложений и инструментов программирования. 3. Анализ технического задания на разработку или доработку клиентской части веб-приложения. 4. Изучение структуры существующего веб-проекта и архитектуры клиентской части приложения. 5. Настройка рабочей среды разработки и подключение к системе контроля версий. 6. Разработка структуры веб-страниц и пользовательского интерфейса информационного ресурса. 7. Создание HTML-разметки страниц веб-приложения. 8. Разработка стилей оформления пользовательского интерфейса с использованием CSS. 9. Реализация адаптивной верстки для корректного отображения интерфейса на различных устройствах. 10. Разработка интерактивных элементов пользовательского интерфейса с использованием JavaScript. 11. Реализация обработки пользовательских действий (события, формы, элементы управления). 12. Использование библиотек и фреймворков для разработки клиентской части веб-приложения. 13. Реализация взаимодействия клиентской части с сервером (отправка запросов и получение данных).	**144**

14. Обработка и отображение данных, полученных от серверной части приложения. 15. Реализация проверки корректности пользовательских данных и валидации форм. 16. Проведение тестирования клиентской части веб-приложения. 17. Выполнение отладки программного кода и устранение ошибок. 18. Проверка корректности отображения интерфейса в различных веб-браузерах. 19. Оптимизация структуры и производительности клиентского кода. 20. Участие в командной разработке проекта и работе с системой контроля версий. 21. Подготовка технической документации на разработанные компоненты интерфейса. 22. Подготовка отчёта по результатам прохождения производственной практики.	
Всего	**506**

.

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:

Лаборатория Разработки веб-приложений, оснащенная в соответствии с п. 6.1.2.3 Примерной образовательной программы по специальности 09.02.09 Веб-разработка.

Оснащенные базы практики в соответствии с п 6.1.2.4 примерной образовательной программы по специальности 09.02.09 Веб-разработка.

Реализация образовательной программы предполагает обязательную учебную и производственную практику.

Учебная практика реализуется в мастерских профессиональной образовательной организации и требует наличия оборудования, инструментов, расходных материалов, обеспечивающих выполнение всех видов работ, определенных содержанием программ профессиональных модулей, в том числе оборудования и инструментов, используемых при проведении чемпионатов профессионального мастерства и указанных в инфраструктурных листах конкурсной документации по компетенции «Веб-технологии» (или их аналогов).

Производственная практика реализуется в организациях, обеспечивающих деятельность обучающихся в профессиональной области информационных технологий.

Оборудование предприятий и технологическое оснащение рабочих мест производственной практики должно соответствовать содержанию профессиональной деятельности и дать возможность обучающемуся овладеть профессиональными компетенциями по всем видам деятельности, предусмотренными программой, с использованием современных технологий, материалов и оборудования.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организации выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список может быть дополнен новыми изданиями.

3.2.1. Основные печатные издания

1. Немцова, Т. И. Компьютерная графика и web-дизайн учеб. пособие / Т.И. Немцова, Т.В. Казанкова, А.В. Шнякин и др. М.: ИД «ФОРУМ»: ИНФРА-М, 2023. — 400 с.
2. Кириченко, А. В. Html5+css3. Основы современного web - дизайна / А. В. Кириченко, А. А. Хрусталев. — 2-е изд. — Санкт-Петербург: Издательство Наука и Техника, 2019. — 352 с. — ISBN 978-5-94387-786-5
3. Кириченко, А. В. JavaScript для FrontEnd-разработчиков. Написание. Тестирование. Развертывание / А. В. Кириченко. — Санкт-Петербург: Издательство Наука и Техника, 2020. — 320 с. — ISBN 978-5-94387-789-6
4. Перлова О. Н. Проектирование и разработка информационных систем: учебник / О. Н. Перлова, О. П. Ляпина, А. В. Гусев. – 3-е изд., испр. – Москва: Издательство ИЦ «Академия», 2020. – 256 с. – (Профессиональное образование). – 25 экз.

3.2.2. Основные электронные издания

1. Немцова, Т. И. Компьютерная графика и web-дизайн [Электронный ресурс]: учеб. пособие / Т.И. Немцова, Т.В. Казанкова, А.В. Шнякин; под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 400 с. // ZNANIUM.COM: электронно-библиотечная система. – Режим доступа: http://znanium.com/catalog.php, ограниченный. – Загл. с экрана.
2. Система федеральных образовательных порталов Информационно - коммуникационные технологии в образовании. [Электронный ресурс] – режим доступа: http://www.ict.edu.ru (2022)
3. Спецификация HTML/DOM/CSS. [Электронный ресурс] – режим доступа: http://w3.org (2022)
4. Справочник по HTML/CSS. [Электронный ресурс] – режим доступа: https://webref.ru (2022)
5. Современный учебник Javascript. [Электронный ресурс] – режим доступа:http://learn.javascript.ru (2022)

3.2.3. Дополнительные источники

1. Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3: учебное пособие / А. В. Диков. — Санкт-Петербург: Издательство Лань, 2022. — 188 с. — ISBN 978-5-8114-3822-8
2. Никольский, А. П. JAVASCRIPT на примерах. Практика, практика и только практика: учебное пособие / А. П. Никольский. — Санкт-Петербург: Издательство Наука и Техника, 2018. — 272 с. — ISBN 978-5-94387-762-9

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
ПК 3.1. Проектировать структуры разделов информационных ресурсов с целью создания эскиза и прототипа интерфейса пользователя.	Оценка «**отлично**» - разработан и реализован отзывчивый дизайн веб – приложения с использованием специальных графических редакторов, применением относительных размеров, контрольных точек и вложенных объектов; макет корректно отображается на различных устройствах; заданные элементы интегрированы в дизайн оптимальным образом; разработанный дизайн полностью соответствует современным стандартам. Оценка «**хорошо**» - разработан и реализован отзывчивый дизайн веб – приложения с использованием специальных графических редакторов, применением нескольких методов; макет корректно отображается на большинстве устройств; заданные элементы интегрированы в общий дизайн; разработанный дизайн соответствует современным стандартам. Оценка «**удовлетворительно**» - разработан и реализован отзывчивый дизайн веб – приложения с использованием специальных графических редакторов, применением нескольких методов; большинство заданных элементов интегрировано в дизайн; макет корректно отображается на одном устройстве; разработанный дизайн в основном соответствует современным стандартам.	Экзамен/зачет в форме собеседования: практическое задание по разработке технического задания на проектирование веб-приложения Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ПК 3.2. Разрабатывать интерфейс пользователя для информационных ресурсов с использованием стандартов в области веб-разработки.	Оценка **«отлично»** - разработан пользовательский интерфейс с помощью профессионального инструментария; разработана и обоснована схема пользовательского веб-интерфейса; во всех элементах приложения учтены требования стандартов к пользовательскому интерфейсу и корпоративный стиль.	Экзамен/зачет в форме собеседования: практическое задание по разработке графических макетов для веб-приложений и интеграции новых графических элементов.

	Оценка **«хорошо»** - разработан и обоснован пользовательский интерфейс с помощью профессионального инструментария; разработана схема пользовательского веб-интерфейса; во всех элементах приложения учтены требования стандартов к пользовательскому интерфейсу и корпоративный стиль. Оценка **«удовлетворительно»** - разработан пользовательский интерфейс с помощью профессионального инструментария; разработана общая схема пользовательского веб-интерфейса; во всех элементах приложения не полностью учтены требования стандартов к пользовательскому интерфейсу и корпоративный стиль.	Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
ПК 3.3. Создавать структуру кода веб-страницы информационных ресурсов в соответствии с дизайн-макетом.	Оценка «**отлично**» - интерфейс пользователя разработан и корректно функционирует в полном соответствии с техническим заданием; приложение предварительно смоделировано (применены объектные модели); использованы анимационные эффекты; код оформлен в соответствии со стандартами кодирования. Оценка «**хорошо**» - интерфейс пользователя разработан и функционирует в соответствии с техническим заданием; приложение предварительно смоделировано; использованы анимационные эффекты; код оформлен в соответствии со стандартами кодирования. Оценка «**удовлетворительно**» - интерфейс пользователя разработан и функционирует; приложение предварительно смоделировано; использованы анимационные эффекты; код оформлен с незначительными отклонениями от стандартов кодирования.	Экзамен/зачет в форме собеседования: практическое задание по тестированию и отладке веб – приложения по предложенному тест-плану. Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ПК 3.4. Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков	Оценка «**отлично**» - веб приложение разработано и корректно функционирует в полном соответствии с техническим заданием в среде программирования с использованием открытых библиотек; приложение предварительно смоделировано (применены объектные модели); код оформлен в соответствии со стандартами кодирования.	Экзамен/зачет в форме собеседования: практическое задание по тестированию и отладке веб – приложения по предложенному тест-плану.

программирования, библиотек и фреймворков.	Оценка «**хорошо**» - веб приложение разработано и работоспособно в соответствии с техническим заданием в среде программирования с использованием открытых библиотек; приложение предварительно смоделировано; код оформлен в соответствии со стандартами кодирования. Оценка «**удовлетворительно**» - веб приложение разработано и работоспособно в соответствии с техническим заданием в среде программирования с использованием открытых библиотек; код оформлен с незначительными отклонениями от стандартов кодирования.	Защита отчетов по практическим и лабораторным работам Экспертное наблюдение за выполнением различных видов работ во время учебной/ производственной
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;	– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; – адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	Экспертное наблюдение за выполнением работ
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;	– использование различных источников, включая электронные ресурсы, медиареурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;	– демонстрация ответственности за принятые решения – обоснованность самоанализа и коррекция результатов собственной работы;	

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	– взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; – обоснованность анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;	– Демонстрировать грамотность устной и письменной речи, ясность формулирования и изложения мыслей	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;	– соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик.	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;	– эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; – демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;	– эффективность использования средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.	
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.	– эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
МДКн.03.01 ПРОЕКТИРОВАНИЕ И ДИЗАЙН ИНТЕРФЕЙСОВ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.09 «Веб-разработка»** и предназначен для проведения текущего контроля и промежуточной аттестации по дисциплине **МДК.03.01 «Проектирование и дизайн интерфейсов»**.

ФОС представляет собой комплекс теоретических вопросов и практических заданий, имитирующих реальные задачи UX/UI-дизайнера и проектировщика интерфейсов в контексте современной веб-индустрии.

2. Цель и задачи курса

Цель: Подготовка квалифицированного специалиста, способного самостоятельно проектировать логику взаимодействия и визуальное оформление пользовательских интерфейсов для веб-приложений различной сложности.

Задачи:

- **Изучение психологии пользователя:** Освоение принципов пользовательского опыта (UX), законов юзабилити и методов исследования целевой аудитории.
- **Проектирование архитектуры:** Овладение навыками построения карт сайтов, пользовательских сценариев (User Flow) и информационной структуры ресурса.
- **Визуальное проектирование:** Изучение основ композиции, типографики и колористики для создания эстетичных и функциональных интерфейсов.
- **Прототипирование:** Внедрение инструментов быстрого макетирования (Figma) и создание интерактивных прототипов в рабочий процесс.
- **Адаптивность и доступность:** Настройка интерфейсов для корректной работы на мобильных устройствах и обеспечение доступности контента.

3. Структура оценочных материалов

Содержание ФОС коррелирует с этапами жизненного цикла проектирования интерфейса:

- **Модуль «Аналитика и логика (UX)»:** Оценка умения проводить юзабилити-аудит, создавать персонажей и проектировать логические схемы взаимодействия.
- **Модуль «Структурное проектирование»:** Контроль навыков создания низкоуровневых макетов (wireframes) и информационной архитектуры веб-ресурса.
- **Модуль «Визуальный дизайн (UI)»:** Практические задания по работе с цветом, шрифтами и графикой; создание высокоуровневых визуальных макетов в графических редакторах.
- **Модуль «Интерактивность и тестирование»:** Проверка компетенций в области создания кликабельных прототипов и проведения юзабилити-тестирования на пользователях.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** законы юзабилити; психологию цветового восприятия; принципы построения адаптивных сеток; методы тестирования интерфейсов.
- **Уметь:** создавать прототипы разной степени детализации; работать с компонентами и дизайн-системами; подготавливать дизайн-макеты к передаче в разработку.
- **Владеть:** инструментарием Figma; навыками создания User Flow и CJM; методиками анализа интерфейсов конкурентов.

5. Методические рекомендации по оцениванию

Оценивание строится на принципах **ориентированности на пользователя (User-Centered Design)**:

- **Теоретическая часть:** Оценивается владение терминологией дизайна, понимание паттернов поведения пользователей и знание гайдлайнов современных платформ.
- **Практическая часть:** Оценивается по факту удобства и логичности созданного интерфейса. Критическими ошибками считаются: нарушение логики переходов, затрудняющее выполнение целевого действия; отсутствие визуальной иерархии; несоблюдение принципов адаптивности.

Краткое изложение планов хода лекционных занятий:

1. Понятие пользовательского интерфейса. Роль UI/UX в разработке веб-приложений. Основные этапы проектирования интерфейса.

О чем занятие: Определение интерфейса как системы взаимодействия. Разделение зон ответственности UI и UX. Общий алгоритм создания продукта от идеи до реализации.

Ход занятия:

- **Терминология:** интерфейс как инструмент решения задач пользователя.
- **UI vs UX:** функциональность и логика против визуального воплощения и эстетики.
- **Жизненный цикл:** анализ, проектирование, прототипирование и тестирование.
- **Бизнес-роль:** влияние качественного интерфейса на конверсию и лояльность.

2. Принципы пользовательского опыта. Потребности пользователей. Понятие usability.

О чем занятие: Изучение психологии поведения пользователя. Определение критериев удобства (юзабилити) и способов выявления потребностей аудитории.

Ход занятия:

- **Пирамида потребностей в UX:** от функциональности к ценности и удовольствию.
- **Понятие Usability:** метрики простоты освоения и эффективности интерфейса.
- **Профилирование:** создание персонажей (Personas) и анализ их целей.
- **Ментальные модели:** как пользователи ожидают увидеть работу системы.

3. Законы юзабилити. Правила проектирования удобных интерфейсов. Типичные ошибки интерфейсов.

О чем занятие: Фундаментальные правила проектирования, основанные на когнитивной психологии. Анализ распространенных проблем, мешающих взаимодействию.

Ход занятия:

- **Ключевые законы:** Закон Фиттса, закон Хика, закон Якоба и принцип «Бритвы Оккама».
- **Эвристики Якоба Нильсена:** чек-лист для проверки интерфейса на прочность.
- **Обратная связь:** важность информирования пользователя о состоянии системы.
- **Антипаттерны:** разбор примеров перегруженных и запутанных интерфейсов.

4. Структура сайта. Навигация и организация контента. Карта сайта.

О чем занятие: Логическое построение скелета проекта. Методы группировки информации и проектирование путей перемещения пользователя.

Ход занятия:

- **Информационная архитектура:** методы классификации и сортировки контента.
- **Типы навигации:** глобальная, локальная, контекстная и вспомогательная.
- **Проектирование Sitemap:** создание иерархической модели страниц.
- **User Flow:** визуализация путей движения пользователя для выполнения задач.

5. Баланс, контраст, выравнивание, повторение. Визуальная иерархия элементов.

О чем занятие: Применение базовых принципов графического дизайна для управления вниманием. Создание структуры через визуальные акценты.

Ход занятия:

- **Принципы CRAP:** Контраст, Повторение, Выравнивание, Приближенность.
- **Визуальный вес:** методы выделения главного над второстепенным.
- **Паттерны сканирования:** F-паттерн и Z-паттерн чтения веб-страниц.
- **Работа с «воздухом»:** использование пустого пространства как элемента композиции.

6. Цветовые модели. Цветовые схемы. Психология цвета в веб-дизайне.

О чем занятие: Техническая сторона работы с цветом и его эмоциональное воздействие. Правила создания гармоничных палитр.

Ход занятия:

- **Технические модели:** RGB vs CMYK, работа с HEX и HSL в вебе.
- **Цветовой круг:** создание схем (комплиментарные, триады, аналогичные).
- **Семантика цвета:** культурные и психологические ассоциации различных оттенков.
- **Доступность:** обеспечение контрастности для удобства чтения (стандарты WCAG).

7. Шрифты, читаемость текста, правила использования типографики.

О чем занятие: Работа с текстовым контентом как основным носителем информации. Правила выбора и оформления шрифтов.

Ход занятия:

- **Классификация:** Антиква, Гротеск, Акциденция. Когда и что выбирать.
- **Параметры текста:** кегль, интерлиньяж, кернинг и длина строки для веба.
- **Шрифтовые пары:** принципы сочетания заголовков и основного текста.
- **Читабельность:** влияние веб-безопасных шрифтов и сервисов (Google Fonts) на UX.

8. Понятие адаптивного и responsive дизайна. Подходы Mobile First.

О чем занятие: Обеспечение корректного отображения интерфейса на разных устройствах. Стратегии проектирования под мобильные платформы.

Ход занятия:

- **Responsive vs Adaptive:** разница между гибкой сеткой и фиксированными брейкпоинтами.
- **Концепция Mobile First:** проектирование от минимального экрана к десктопу.
- **Сенсорный ввод:** особенности проектирования кнопок и жестов (Tap targets).
- **Оптимизация контента:** скрытие или перестроение блоков под разные разрешения.

9. Низкоуровневые и высокоуровневые прототипы. Инструменты прототипирования.

О чем занятие: Переход от структуры к визуализации. Инструментарий и уровни детализации макетов.

Ход занятия:

- **Low-fi прототипы:** скетчи и варфреймы для быстрой проверки логики.
- **High-fi прототипы:** детальные макеты с реальным контентом и дизайном.
- **Интерактивность:** создание кликабельных связей для имитации работы сайта.
- **Обзор ПО:** Figma как основной индустриальный стандарт совместной работы.

10. UI-паттерны, минимализм, микроанимации, дизайн-системы.

О чем занятие: Современные тренды и методы ускорения разработки. Создание целостного визуального языка продукта.

Ход занятия:

- **UI-паттерны:** использование стандартных решений (карточки, модальные окна, табы).
- **Микроанимации:** функциональный отклик интерфейса на действия пользователя.
- **Минимализм в вебе:** принцип «меньше значит больше» для снижения когнитивной нагрузки.
- **Дизайн-системы:** компоненты, стили и правила для масштабируемых проектов.

Краткое изложение планов хода практических занятий и примерные задания:

1. Анализ интерфейсов популярных веб-сайтов с точки зрения UX и usability

Ход занятия: Изучение методов экспертной оценки интерфейсов. Анализ конкурентов с использованием 10 эвристик Нильсена. Выявление сильных и слабых сторон в логике взаимодействия.

Задание: Выбрать два популярных интернет-магазина и составить сравнительную таблицу их юзабилити-ошибок и удачных решений.

2. Создание карты сайта и пользовательских сценариев

Ход занятия: Проектирование информационной архитектуры. Определение основных групп пользователей и целей, которые они преследуют на сайте.

Задание: Разработать иерархическую карту сайта (Sitemap) для онлайн-сервиса бронирования и описать 3 сценария поведения пользователя.

3. Разработка user flow для веб-приложения

Ход занятия: Визуализация пути пользователя от точки входа до целевого действия. Проектирование логики переходов между экранами.

Задание: Построить схему User Flow для процесса регистрации и оформления заказа в приложении по доставке еды.

4. Создание wireframe главной страницы сайта

Ход занятия: Компоновка структуры блоков страницы без использования графики и цветов. Фокус на расположении элементов навигации и контента.

Задание: Создать низкоуровневый черно-белый эскиз (Wireframe) главной страницы новостного портала.

5. Создание wireframe внутренних страниц сайта

Ход занятия: Проектирование структуры второстепенных страниц. Обеспечение единообразия элементов управления на разных уровнях вложенности.

Задание: Разработать Wireframe страницы «Личный кабинет» и «Карточка товара», соблюдая общую логику навигации.

6. Разработка интерактивного прототипа в Figma

Ход занятия: Знакомство с интерфейсом Figma. Создание связей между фреймами (Prototype mode), настройка триггеров и анимации переходов.

Задание: Связать созданные ранее wireframes в единый кликабельный прототип, имитирующий переход с главной страницы на внутренние.

7. Подбор цветовой палитры для веб-интерфейса

Ход занятия: Работа с цветовым кругом и онлайн-генераторами палитр. Проверка контрастности текста и фона на соответствие стандартам доступности (WCAG).

Задание: Разработать основную и акцентную цветовые схемы для финансового приложения и обосновать выбор с точки зрения психологии цвета.

8. Подбор и настройка шрифтов для интерфейса

Ход занятия: Работа с библиотеками шрифтов (Google Fonts). Создание типографической сетки: настройка размеров заголовков, основного текста и межстрочных интервалов.

Задание: Сформировать шрифтовую пару (заголовок + основной текст) и настроить иерархию текстовых стилей в дизайн-макете.

9. Разработка макета для мобильной версии сайта

Ход занятия: Применение подхода Mobile First. Адаптация контента под узкие экраны: изменение сетки, скрытие второстепенных элементов в «бургер-меню».

Задание: Переработать макет главной страницы сайта для разрешения экрана мобильного телефона (360–390px).

10. Разработка макета страницы приложения

Ход занятия: Специфика дизайна мобильных приложений: использование гайдлайнов (Human Interface Guidelines или Material Design). Проектирование таб-баров и навигационных панелей.

Задание: Оформить визуальный макет одного функционального экрана приложения (например, «Настройки профиля»).

11. Создание визуального макета страницы в Figma

Ход занятия: Работа с изображениями, иконками и эффектами (тени, скругления). Превращение wireframe в законченный высокоуровневый дизайн (UI).

Задание: Выполнить финальную отрисовку главной страницы сайта в цвете с использованием выбранной типографики и контента.

12. Создание кнопок, форм, карточек, меню

Ход занятия: Отрисовка базовых элементов интерфейса. Работа с состояниями элементов (Hover, Active, Disabled). Соблюдение консистентности стиля.

Задание: Отрисовать набор кнопок в разных состояниях и форму обратной связи для своего проекта.

13. Формирование набора UI-компонентов

Ход занятия: Понятие атомарного дизайна. Создание мастер-компонентов и вариантов (Variants) в Figma для упрощения масштабирования проекта.

Задание: Собрать библиотеку используемых в проекте элементов (UI Kit): цвета, шрифты, кнопки, поля ввода и иконки.

14. Создание полного прототипа веб-ресурса

Ход занятия: Сборка всех отрисованных страниц в единую интерактивную систему. Настройка сложных переходов и микроанимаций.

Задание: Подготовить финальный интерактивный прототип сайта, состоящий минимум из 5 полностью отрисованных страниц.

15. Проведение тестирования usability разработанного интерфейса

Ход занятия: Методика «коридорного» тестирования. Постановка задач тестировщикам. Наблюдение за действиями пользователей и фиксация затруднений.

Задание: Провести тестирование своего прототипа на двух добровольцах и составить список выявленных проблем в навигации или логике.

16. Исправление выявленных ошибок интерфейса

Ход занятия: Работа над ошибками по результатам тестов. Рефакторинг дизайн-решений. Оптимизация путей пользователя.

Задание: Внести изменения в прототип Figma, исправляющие критические замечания, полученные на этапе тестирования.

17. Представление проекта интерфейса и его защита

Ход занятия: Подготовка презентации проекта. Обоснование выбранных UX-решений, цветовой гаммы и типографики перед аудиторией.

Задание: Публично защитить свой проект, продемонстрировать интерактивный прототип и ответить на вопросы по логике его работы.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МДКн.03.02 ВЕРСТКА СТРАНИЦ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование» (или смежным) и предназначен для проведения текущего контроля и промежуточной аттестации по разделу **Тема 2.1. «Верстка страниц»** в рамках профессионального модуля МДК.03.02.

ФОС представляет собой совокупность контрольно-измерительных материалов, включающих теоретические задания на знание стандартов W3C и практические работы по созданию адаптивных, семантически корректных веб-интерфейсов.

2. Цель и задачи курса

Цель: Формирование профессиональных компетенций, необходимых для качественного преобразования графических макетов в интерактивный и кроссбраузерный код, соответствующий современным стандартам веб-разработки.

Задачи:

- **Освоение языков разметки и стилизации:** Глубокое изучение синтаксиса HTML5 и возможностей CSS3 для управления структурой и визуальным представлением данных.
- **Изучение технологий компоновки:** Овладение современными методами построения сеток (Flexbox, CSS Grid) для создания интерфейсов любой сложности.
- **Обеспечение адаптивности:** Внедрение принципов Responsive Design и стратегии Mobile First для корректного отображения ресурсов на всех типах устройств.
- **Стандартизация и методология:** Применение профессиональных подходов (БЭМ-нейминг, семантическая разметка) для обеспечения чистоты и масштабируемости кода.
- **Валидация и оптимизация:** Навыки проверки кода на соответствие стандартам и оптимизация верстки для высокой скорости загрузки.

3. Структура оценочных материалов

Содержание ФОС разделено на функциональные блоки, соответствующие этапам разработки интерфейса:

- **Модуль «Структура и семантика»:** Проверка умения логически верно размечать текстовый и мультимедийный контент, проектировать формы и таблицы.
- **Модуль «Визуальное проектирование»:** Оценка навыков работы с типографикой, цветовыми схемами, фоновыми изображениями и блочной моделью CSS.
- **Модуль «Современные Layout-технологии»:** Практические задания на построение гибких интерфейсов с использованием модулей Flexbox и Grid.
- **Модуль «Адаптивность и профессиональная сборка»:** Контроль компетенций в области медиа-запросов, применения методологии БЭМ и кроссбраузерной оптимизации.

4. Планируемые результаты освоения

В результате освоения темы обучающийся должен демонстрировать следующие результаты:

- **Знать:** стандарты HTML5 и CSS3; правила семантической разметки; принципы построения сеток; специфику адаптивного дизайна; основы методологии БЭМ.
- **Уметь:** верстать страницы по графическим макетам; использовать современные инструменты позиционирования; адаптировать контент под мобильные платформы; работать с инструментами разработчика (DevTools).
- **Владеть:** навыками написания чистого, валидного кода; методами оптимизации графики и стилей; инструментарием для создания интерактивных прототипов и финальной верстки.

5. Методические рекомендации по оцениванию

Оценивание результатов обучения проводится комплексно:

- **Теоретическая часть:** Оценивается понимание стандартов веба, знание свойств CSS и структуры HTML-документа через контрольные вопросы или тесты.
- **Практическая часть:** Оценивается по принципу «живого кода». Ключевыми критериями являются:
    1. **Точность:** соответствие сверстанной страницы исходному макету (Pixel Perfect).
    2. **Адаптивность:** корректное перестроение блоков при изменении ширины экрана.
    3. **Валидность:** отсутствие ошибок в HTML и CSS коде.
    4. **Культура кода:** использование понятных имен классов (БЭМ), отсутствие дублирования стилей.

Краткое изложение планов хода лекционных занятий:

1. Понятие веб-верстки. Роль HTML и CSS в разработке веб-страниц.

О чем занятие: Определение верстки как этапа преобразования дизайна в код. Разделение ответственности между структурой контента и его визуальным представлением.

Ход занятия:

- **Веб-технологии:** место верстки в цикле создания IT-продукта (Frontend).
- **Стек технологий:** роль HTML как каркаса и CSS как инструмента стилизации.
- **Браузеры:** как движки рендеринга считывают и отображают код.
- **Семантика:** важность использования правильных тегов для SEO и доступности.

2. Структура HTML-документа. Основные теги и элементы.

О чем занятие: Изучение анатомии HTML-страницы и базового синтаксиса языка разметки.

Ход занятия:

- **Скелет страницы:** обязательные элементы `<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`.
- **Секция Head:** назначение мета-тегов, заголовка окна и подключение ресурсов.
- **Синтаксис:** парные и одиночные теги, иерархия вложенности элементов.
- **Атрибуты:** способы расширения функциональности тегов через глобальные и специфичные атрибуты.

3. Заголовки, абзацы, списки, форматирование текста.

О чем занятие: Инструменты для создания логической структуры текстового контента и его базового оформления.

Ход занятия:

- **Иерархия заголовков:** использование `h1-h6` для логической разметки документа.
- **Текстовые блоки:** работа с абзацами `p` и принудительными переносами строки.
- **Списки:** создание маркированных (`ul`), нумерованных (`ol`) и вложенных списков.
- **Форматирование:** логическое выделение элементов (`strong`, `em`) и работа со спецсимволами.

4. Создание гиперссылок. Работа с изображениями.

О чем занятие: Принципы связывания страниц в единую систему и интеграция графического контента.

Ход занятия:

- **Гипертекст:** тег `<a>`, работа с атрибутами `href`, `target` и `title`.

- **Пути к файлам:** разница между абсолютными и относительными ссылками.
- **Графика:** вставка изображений через `<img>`, важность атрибута `alt`.
- **Форматы:** выбор между JPEG, PNG, SVG и современным WebP.

5. Создание и оформление таблиц.

О чем занятие: Методы представления структурированных данных и правила разметки табличных элементов.

Ход занятия:

- **Структура таблицы:** назначение тегов `table`, `tr`, `td`, `th`.
- **Семантика:** группировка контента внутри `thead`, `tbody` и `tfoot`.
- **Объединение:** использование атрибутов `colspan` и `rowspan` для сложных ячеек.
- **Оформление:** базовое управление границами и отступами внутри таблицы.

6. Элементы форм, поля ввода, кнопки.

О чем занятие: Проектирование интерфейсов для сбора данных и взаимодействия с пользователем.

Ход занятия:

- **Контейнер формы:** тег `form`, атрибуты `action` и `method`.
- **Input-поля:** обзор типов (text, password, checkbox, radio, email).
- **Интерактив:** создание кнопок `button` и выпадающих списков `select`.
- **Юзабилити:** связка полей с подписями через тег `label`.

7. Назначение CSS. Подключение таблиц стилей.

О чем занятие: Введение в каскадные таблицы стилей и способы их интеграции в проект.

Ход занятия:

- **Концепция CSS:** отделение оформления от структуры.
- **Методы внедрения:** внешние файлы (`link`), внутренние стили (`style`) и инлайновые атрибуты.
- **Синтаксис:** структура правила (селектор, свойство, значение).
- **Преимущества:** централизованное управление дизайном всего сайта.

8. Типы селекторов. Приоритет и каскадность.

О чем занятие: Изучение механизмов выбора элементов и разрешения конфликтов между стилями.

Ход занятия:

- **Базовые селекторы:** по тегу, классу и идентификатору (ID).
- **Комбинации:** селекторы потомков, дочерние и контекстные селекторы.

- **Каскад:** как браузер определяет, какой стиль применить последним.
- **Вес (Специфичность):** расчет приоритета селектора и использование `!important`.

9. Margin, padding, border, width, height.

О чем занятие: Изучение блочной модели CSS (Box Model) — основы управления пространством.

Ход занятия:

- **Box Model:** различие между контентом, внутренними отступами (`padding`) и внешними (`margin`).
- **Рамки:** настройка границ через `border`.
- **Размеры:** управление габаритами блока и свойство `box-sizing`.
- **Схлопывание:** особенности поведения вертикальных внешних отступов.

10. Шрифты, размеры, межстрочные интервалы.

О чем занятие: Работа с типографикой и настройка читабельности текстовых блоков.

Ход занятия:

- **Шрифты:** подключение семейств через `font-family`, веб-безопасные шрифты.
- **Размеры:** единицы измерения (px, em, rem) и свойство `font-size`.
- **Интерлиньяж:** управление высотой строки через `line-height`.
- **Начертание:** жирность, курсив и трансформация регистра текста.

11. Цветовые свойства и фоновые изображения.

О чем занятие: Визуальное оформление элементов через цвет, градиенты и фоновую графику.

Ход занятия:

- **Цвет:** использование HEX, RGB и именованных цветов.
- **Фон:** свойства `background-image`, `background-repeat` и `background-size`.
- **Позиционирование:** управление положением фона через `background-position`.
- **Прозрачность:** работа с альфа-каналом в формате RGBA.

12. Основы гибкой верстки с использованием Flexbox.

О чем занятие: Современный метод выравнивания элементов по одной оси.

Ход занятия:

- **Контейнер и элементы:** активация `display: flex`.
- **Направления:** управление главной и поперечной осями (`flex-direction`).
- **Выравнивание:** свойства `justify-content` и `align-items`.
- **Управление весом:** распределение пространства через `flex-grow` и `flex-shrink`.

13. Использование CSS Grid для построения макетов.

О чем занятие: Создание сложных двумерных сеток (колонок и рядов) любой конфигурации.

Ход занятия:

- **Сетка:** создание сетки через `grid-template-columns` и `rows`.
- **Единица fr:** гибкое распределение долей свободного пространства.
- **Размещение:** позиционирование элементов в конкретных ячейках или областях.
- **Расстояния:** управление зазорами между колонками через `gap`.

14. Медиа-запросы. Подход Mobile First.

О чем занятие: Создание адаптивных интерфейсов, подстраивающихся под размеры экрана.

Ход занятия:

- **Viewport:** назначение мета-тега для корректного масштабирования на смартфонах.
- **@media:** синтаксис условий по ширине и высоте экрана.
- **Mobile First:** методология написания стилей от простого мобильного вида к сложному десктопу.
- **Контрольные точки:** выбор оптимальных Breakpoints для перестроения верстки.

15. БЭМ-методология, кроссбраузерность, оптимизация верстки.

О чем занятие: Профессиональные стандарты написания кода, поддержка браузеров и ускорение загрузки.

Ход занятия:

- **БЭМ:** правила именования классов (Блок-Элемент-Модификатор) для чистоты кода.
- **Кроссбраузерность:** использование вендорных префиксов и сброс стилей (Reset/Normalize).
- **Валидация:** проверка кода на соответствие стандартам W3C.
- **Оптимизация:** минификация файлов, спрайты и сжатие ресурсов для повышения скорости.

Краткое изложение планов хода практических занятий и примерные задания:

1. Создание базовой структуры HTML-страницы

Ход занятия: Знакомство с редактором кода. Создание файла `index.html`. Написание обязательной структуры документа и проверка отображения в браузере.

Задание: Создать пустой HTML-документ, содержащий корректный DOCTYPE, теги `html`, `head` (с кодировкой и заголовком) и `body`.

2. Использование заголовков, абзацев и списков

Ход занятия: Наполнение страницы текстовым контентом. Практика вложенности тегов. Разметка логических блоков текста.

Задание: Разместить на странице статью с одним заголовком `h1`, двумя `h2` и маркированным списком из 5 пунктов.

3. Вставка изображений и настройка атрибутов

Ход занятия: Подготовка графических файлов. Использование тега `img`. Настройка путей к файлам и текстового описания для доступности.

Задание: Разместить на странице три изображения разных форматов (jpg, png, svg), настроив для каждого корректный путь и атрибут `alt`.

4. Создание внутренних и внешних ссылок

Ход занятия: Создание многостраничной структуры. Работа с якорными ссылками внутри страницы и ссылками на сторонние ресурсы.

Задание: Создать навигационное меню из 3 пунктов, ведущих на разные секции текущей страницы, и одну ссылку на внешний сайт.

5. Разработка таблиц в HTML

Ход занятия: Разметка данных в виде сетки. Использование тегов `table`, `tr`, `td`. Понимание логики формирования строк и столбцов.

Задание: Создать таблицу расписания занятий на 3 дня с использованием заголовков столбцов `th`.

6. Настройка структуры таблиц

Ход занятия: Работа со сложными табличными структурами. Объединение ячеек. Применение семантических тегов `thead` и `tbody`.

Задание: Модифицировать таблицу из предыдущего задания, объединив несколько ячеек по горизонтали и вертикали.

7. Разработка формы ввода данных

Ход занятия: Проектирование каркаса формы. Использование контейнера `form`. Настройка атрибутов для отправки данных.

Задание: Создать форму «Обратная связь», включающую поле для имени, e-mail и кнопку отправки.

8. Использование различных типов полей ввода

Ход занятия: Расширение функционала форм. Работа с чекбоксами, радиокнопками, выпадающими списками и текстовыми областями.

Задание: Добавить в форму элементы выбора пола (radio), согласия с правилами (checkbox) и поле для длинного сообщения (textarea).

9. Подключение таблиц стилей к HTML-странице

Ход занятия: Создание файла `style.css`. Изучение приоритетов подключения (инлайновые, внутренние, внешние).

Задание: Подключить внешний файл стилей к HTML-документу и изменить цвет фона страницы через CSS.

10. Применение различных типов CSS-селекторов

Ход занятия: Практика точечного обращения к элементам. Использование классов, идентификаторов и сложных комбинаций селекторов.

Задание: Стилизовать список так, чтобы каждый четный элемент имел другой цвет, используя соответствующие селекторы.

11. Работа со шрифтами и текстовыми стилями

Ход занятия: Настройка типографики. Подключение шрифтов. Управление размером, жирностью и межстрочным интервалом.

Задание: Установить для основного текста шрифт без засечек, задать высоту строки 1.5 и выровнять текст по ширине.

12. Настройка цветового оформления страницы

Ход занятия: Работа с цветовыми схемами. Использование HEX, RGB и прозрачности. Стилизация рамок и теней.

Задание: Оформить блоки текста, задав им разные цвета фона и границы (border) с использованием различных форматов записи цвета.

13. Настройка отступов и границ элементов

Ход занятия: Работа с Box Model. Визуальное разделение блоков. Настройка внутренних (padding) и внешних (margin) отступов.

Задание: Создать три блока и расположить их друг под другом с внешним отступом 20px, добавив внутренние поля для текста.

14. Верстка структуры страницы

Ход занятия: Создание макета с использованием семантических тегов HTML5 (`header`, `main`, `footer`, `section`, `aside`).

Задание: Разметить каркас главной страницы сайта, выделив основные логические зоны документа.

15. Верстка меню сайта

Ход занятия: Превращение списка ссылок в горизонтальную или вертикальную навигационную панель. Стилизация состояний ссылок (`:hover`).

Задание: Сверстать горизонтальное меню, убрав стандартные маркеры списка и добавив эффект изменения цвета при наведении.

16. Использование фоновых изображений

Ход занятия: Работа со свойством `background`. Масштабирование и позиционирование фонов. Создание декоративных блоков. **З**

адание: Задать для шапки сайта фоновое изображение, которое будет полностью перекрывать блок без повторений.

17. Верстка карточек товаров или статей

Ход занятия: Создание повторяющихся независимых блоков. Группировка изображения, заголовка, описания и кнопки внутри одной карточки.

Задание: Сверстать сетку из трех одинаковых карточек товаров с использованием рамок и внутренних отступов.

18. Создание макета страницы с Flexbox

Ход занятия: Настройка контейнера `display: flex`. Распределение блоков по горизонтали и вертикали. **Задание:** Расположить логотип и меню в шапке сайта по разным сторонам (краям) страниц

ы, используя `justify-content`.

19. Верстка блока с использованием Flexbox

Ход занятия: Отработка сложных приемов Flexbox. Выравнивание элементов разной высоты. Управление порядком отображения.

Задание: Создать блок преимуществ из 4 элементов, которые равномерно распределяются по ширине страницы и переносятся на новую строку.

20. Создание сеточного макета страницы

Ход занятия: Знакомство с CSS Grid. Определение колонок и рядов. Создание базовой сетки макета.

Задание: Разработать сетку страницы, состоящую из 3 колонок, где боковые колонки имеют фиксированную ширину, а центральная — гибкую.

21. Разработка сложной сетки страницы

Ход занятия: Использование областей сетки (`grid-template-areas`). Создание асимметричных макетов.

Задание: Реализовать макет, в котором один из элементов (например, баннер) занимает сразу несколько колонок и строк.

22. Создание header сайта

Ход занятия: Детальная проработка верхней части сайта. Размещение логотипа, поиска, контактов и навигации.

Задание: Полностью сверстать шапку сайта по макету, обеспечив правильное позиционирование всех вложенных элементов.

23. Создание footer сайта

Ход занятия: Стилизация подвала страницы. Размещение копирайта, ссылок на соцсети и карты сайта.

Задание: Сверстать многоколоночный футер с контактной информацией и ссылками на разделы сайта.

24. Разработка основной части страницы

Ход занятия: Наполнение контентной области. Верстка текстовых блоков, медиа-файлов и интерактивных элементов.

Задание: Сверстать главную секцию страницы, включающую текстовое описание и галерею из 4 изображений.

25. Верстка страницы по готовому макету

Ход занятия: Начало итоговой сборки страницы. Перенос графических решений и размеров из дизайн-макета в код.

Задание: Приступить к верстке целевой страницы (Landing Page), соблюдая все отступы и размеры согласно образцу.

26. Продолжение верстки страницы

Ход занятия: Реализация оставшихся блоков макета. Проверка целостности дизайна и вложенности элементов.

Задание: Доверстать недостающие секции страницы (отзывы, форма подписки) и проверить корректность работы всех ссылок.

27. Использование медиа-запросов

Ход занятия: Изучение синтаксиса `@media`. Определение точек перелома (breakpoints) для изменения стилей.

Задание: Написать стили, которые меняют цвет фона и размер шрифта при уменьшении ширины экрана до 768px.

28. Адаптация страницы под мобильные устройства

Ход занятия: Перестроение колонок в один ряд. Скрытие лишних элементов. Адаптация меню под узкие экраны.

Задание: Сделать так, чтобы сетка из 3 колонок на мобильных устройствах (до 480px) превращалась в одну колонку.

29. Оптимизация интерфейса для планшетов

Ход занятия: Настройка промежуточных состояний макета. Корректировка размеров кнопок и шрифтов для удобства касания.

Задание: Настроить отображение карточек товаров для планшетной версии (в две колонки вместо трех).

30. Проверка отображения в различных браузерах

Ход занятия: Тестирование верстки в Chrome, Firefox, Safari. Использование инструментов разработчика для поиска несоответствий. **Задание:** Выявить и зафиксировать визуальные различия в отображении шрифтов и теней в разных браузерах.

31. Анализ и исправление ошибок

Ход занятия: Исправление багов верстки. Работа с переполнением блоков (overflow) и позиционированием элементов.

Задание: Устранить выявленные в ходе тестирования ошибки (например, «выпадение» текста из кнопок или наложение блоков).

32. Улучшение структуры CSS-кода

Ход занятия: Очистка кода от дублей. Группировка свойств. Использование CSS-переменных для цветов и размеров.

Задание: Вынести повторяющиеся значения цветов в переменные (`--main-color`) и переписать стили с их использованием.

33. Применение методологии БЭМ

Ход занятия: Рефакторинг классов по схеме Блок-Элемент-Модификатор. Повышение независимости компонентов.

Задание: Переименовать классы в верстке шапки и карточек товаров в соответствии с правилами БЭМ.

34. Завершение проекта страницы сайта

Ход занятия: Финальная проверка кода на валидность. Оптимизация изображений и путей. Подготовка проекта к сдаче.

Задание: Проверить верстку через W3C Validator, исправить предупреждения и собрать проект в архив.

35. Представление выполненной верстки сайта

Ход занятия: Демонстрация готового проекта. Рассказ о выбранных технических решениях и методах адаптации.

Задание: Публично продемонстрировать верстку сайта в браузере, показав корректную работу адаптивности и интерактивных элементов.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МДКн.03.03 РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ ИНФОРМАЦИОННЫХ РЕСУРСОВ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург

2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.07 «Информационные системы и программирование»** и предназначен для проведения текущего контроля и промежуточной аттестации по разделу **МДК.03.03 «Разработка клиентской части информационных ресурсов»**.

ФОС включает в себя комплекс практических заданий и контрольных вопросов, направленных на проверку навыков программирования на языке JavaScript, умения манипулировать объектной моделью документа (DOM) и организовывать взаимодействие с серверной частью приложений.

2. Цель и задачи курса

Цель: Подготовка специалиста, обладающего компетенциями для создания динамических, интерактивных интерфейсов веб-приложений и реализации сложной логики на стороне клиента.

Задачи:

- **Изучение основ алгоритмизации:** Овладение базовыми конструкциями языка (циклы, условия, функции) для реализации программной логики.
- **Управление DOM-структурой:** Освоение методов динамического изменения контента, стилей и структуры веб-страницы без её перезагрузки.
- **Обработка пользовательского взаимодействия:** Настройка реакций приложения на действия пользователя (события мыши, клавиатуры, ввод в формы).
- **Асинхронное программирование:** Изучение механизмов работы с внешними данными (API, Fetch, JSON) и управления асинхронными процессами.
- **Работа с хранилищами данных:** Использование возможностей браузера для сохранения состояния приложения и персональных настроек пользователя.

3. Структура оценочных материалов

Содержание ФОС структурировано по принципу постепенного усложнения задач — от основ языка до создания полноценного клиентского приложения:

- **Модуль «Базовый синтаксис и логика»:** Оценка знаний типов данных, операторов, циклов и навыков создания пользовательских функций.
- **Модуль «Манипуляция DOM и события»:** Практические кейсы по динамическому созданию элементов интерфейса и обработке интерактивных сценариев.
- **Модуль «Асинхронность и сетевое взаимодействие»:** Проверка умений работать с Promise, async/await и интеграция приложения с внешними API через Fetch.
- **Модуль «Проектирование приложения»:** Итоговое практическое задание по разработке комплексного клиентского решения с использованием локального хранилища и рефакторингом кода.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** стандарты языка JavaScript (ECMAScript); архитектуру DOM; принципы работы асинхронного кода; методы валидации форм; форматы обмена данными (JSON).
- **Уметь:** разрабатывать скрипты для интерактивных элементов; выполнять запросы к серверу; отлаживать код в консоли браузера; структурировать клиентский код согласно современным методикам.
- **Владеть:** навыками создания Single Page Application (SPA) логики; методами работы с Web Storage; инструментарием Chrome DevTools для поиска и исправления логических ошибок.

5. Методические рекомендации по оцениванию

Оценивание результатов строится на анализе качества программного кода и функциональности итогового продукта:

- **Теоретическая часть:** Оценивается понимание принципов работы движка JavaScript, области видимости и цепочек промисов.
- **Практическая часть:** Оценивается по следующим критериям:
    1. **Работоспособность:** отсутствие ошибок в консоли и выполнение поставленной функциональной задачи.
    2. **Чистота кода:** отсутствие избыточных конструкций, правильное именование переменных и функций.
    3. **Безопасность:** наличие проверок входных данных (валидация) и корректная обработка ошибок при запросах к API.
    4. **UX/UI отклик:** наличие визуальной обратной связи для пользователя при выполнении асинхронных операций (лоадеры, сообщения об успехе/ошибке).

Краткое изложение планов хода лекционных занятий:

1. Назначение клиентских технологий. Роль JavaScript в веб-разработке.

О чем занятие: Место JavaScript в современном стеке (HTML/CSS/JS). Понятие динамического контента и клиентской логики. Среды выполнения кода.

Ход занятия:

- **История и эволюция:** от простых сценариев до мощных веб-приложений.
- **Клиент-серверное взаимодействие:** что происходит в браузере и зачем нужен JS.
- **Подключение сценариев:** тег `<script>`, внешние файлы, атрибуты `async` и `defer`.
- **Возможности и ограничения:** что JS может и чего не может делать в браузере из соображений безопасности.

2. Синтаксис языка. Переменные и типы данных.

О чем занятие: Базовые правила написания кода и классификация данных, которыми оперирует программа.

Ход занятия:

- **Объявление переменных:** различия между `let`, `const` и устаревшим `var`.
- **Типы данных:** примитивы (string, number, boolean, null, undefined) и ссылочные типы.
- **Строгий режим:** назначение `'use strict'`.
- **Динамическая типизация:** как JavaScript преобразует типы данных «на лету».

3. Операторы: арифметические, логические и сравнительные.

О чем занятие: Инструменты для выполнения вычислений и проверки условий. Основы математики и логики в JS.

Ход занятия:

- **Математика:** базовые операции и инкремент/декремент.
- **Сравнение:** строгое (===) и нестрогое (==) равенство, приоритеты операторов.
- **Логика:** операторы И (`&&`), ИЛИ (`||`), НЕ (`!`) и их комбинирование.
- **Шаблонные строки:** использование обратных кавычек для интерполяции выражений.

4. Конструкции if, else, switch. Управление логикой программы.

О чем занятие: Механизмы ветвления, позволяющие программе принимать решения в зависимости от входных данных.

Ход занятия:

- **Условный оператор:** синтаксис `if`, `else if` и `else`.

- **Тернарный оператор:** сокращенная форма записи условий.
- **Множественный выбор:** использование конструкции `switch...case` и значение `default`.
- **Логическое сокращение:** использование операторов `&&` и `||` для краткой записи условий.

5. Циклы: for, while, do...while. Итерации в программах.

О чем занятие: Способы организации многократно повторяющихся действий и обхода структур данных.

Ход занятия:

- **Цикл for:** классический счетчик, инициализация, условие и шаг.
- **Циклы while:** выполнение кода, пока условие истинно.
- **Прерывание и пропуск:** использование инструкций `break` и `continue`.
- **Бесконечные циклы:** причины возникновения и способы предотвращения.

6. Объявление функций. Параметры и возвращаемые значения.

О чем занятие: Функции как основные строительные блоки кода. Повторное использование логики и инкапсуляция.

Ход занятия:

- **Function Declaration vs Expression:** способы создания функций и их различия.
- **Аргументы:** передача данных в функцию, значения по умолчанию.
- **Результат работы:** оператор `return` и важность возвращаемого значения.
- **Стрелочные функции:** краткий синтаксис и особенности контекста.

7. Создание и обработка массивов. Основные методы.

О чем занятие: Работа со списками данных. Изучение мощного инструментария для трансформации и фильтрации массивов.

Ход занятия:

- **Инициализация:** создание массивов и доступ к элементам по индексу.
- **Мутирующие методы:** `push`, `pop`, `shift`, `unshift`, `splice`.
- **Методы перебора:** детальное изучение `forEach`, `map`, `filter`, `reduce`.
- **Поиск:** использование `find`, `includes` и `indexOf`.

8. Создание объектов и работа со свойствами.

О чем занятие: Объекты как способ хранения сложных структур данных. Понятие «ключ-значение».

Ход занятия:

- **Литерал объекта:** создание, добавление и удаление свойств.
- **Доступ к данным:** точечная запись против квадратных скобок.
- **Методы объектов:** использование функций как свойств объекта.
- **Контекст `this`:** базовое представление о работе контекста внутри объекта.

9. Структура DOM. Доступ к элементам страницы.

О чем занятие: Document Object Model как интерфейс для взаимодействия JavaScript с HTML-разметкой.

Ход занятия:

- **Дерево документа:** узлы, элементы и текстовые данные.
- **Поиск элементов:** современные методы `querySelector` и `querySelectorAll`.
- **Манипуляция:** изменение текстового содержимого (`textContent`) и HTML-структуры (`innerHTML`).
- **Атрибуты и классы:** программное управление стилями и свойствами тегов.

10. Обработчики событий. Пользовательские события.

О чем занятие: Создание интерактивности. Реакция страницы на действия пользователя (клики, ввод, прокрутка).

Ход занятия:

- **Метод addEventListener:** назначение слушателей событий.
- **Объект события (event):** получение координат клика, нажатой клавиши и целевого элемента.
- **Всплытие и погружение:** как события распространяются по дереву DOM.
- **Отмена действий:** предотвращение стандартного поведения браузера через `preventDefault`.

11. Обработка данных форм. Проверка данных.

О чем занятие: Взаимодействие пользователя с элементами ввода. Валидация данных перед отправкой.

Ход занятия:

- **Доступ к полям:** получение значений из `input`, `select`, `checkbox`.
- **События форм:** `submit`, `input`, `change`, `blur`.
- **Регулярные выражения:** базовое применение для проверки форматов (email, телефон).
- **Фидбек:** динамическое отображение ошибок валидации пользователю.

12. Асинхронность в JS: Promise, async/await.

О чем занятие: Работа с операциями, которые занимают время. Решение проблемы «зависания» интерфейса.

Ход занятия:

- **Event Loop:** краткий обзор того, как JS выполняет асинхронный код.
- **Промисы (Promises):** состояния (pending, fulfilled, rejected) и цепочки `.then()/.catch()`.
- **Синтаксический сахар:** использование `async` и `await` для написания «линейного» асинхронного кода.
- **Обработка ошибок:** конструкция `try...catch`.

13. Получение данных с сервера. Использование Fetch API.

О чем занятие: Связь клиентской части с внешним миром. Понятие API и формата JSON.

Ход занятия:

- **Протокол HTTP:** кратко о запросах (GET, POST) и ответах сервера.
- **Функция fetch:** отправка запросов и получение ответов.
- **Работа с JSON:** преобразование текстовых данных в объекты JavaScript.
- **Динамическое обновление:** рендеринг полученных данных на страницу без перезагрузки.

14. LocalStorage и SessionStorage.

О чем занятие: Сохранение состояния приложения между перезагрузками страницы.

Ход занятия:

- **Web Storage API:** различия между локальным и сессионным хранилищем.
- **Методы:** `setItem`, `getItem`, `removeItem`, `clear`.
- **Безопасность и лимиты:** что можно и что нельзя хранить в браузере.
- **Практика:** сохранение настроек пользователя (например, темная тема).

15. Отладка, оптимизация и структура клиентского кода.

О чем занятие: Профессиональные подходы к написанию качественного кода и поиску ошибок.

Ход занятия:

- **DevTools:** использование вкладки Console, Sources и точек останова (breakpoints).
- **Оптимизация:** влияние скриптов на скорость загрузки и производительность (Reflow/Repaint).
- **Модульность:** разбиение кода на логические файлы (ES-модули).
- **Чистый код:** именование, комментирование и избавление от «магических чисел».

Краткое изложение планов хода практических занятий и примерные задания:

1. **Подключение скриптов к веб-странице.**

   Ход занятия: Разбор способов подключения (внешний файл, встроенный скрипт). Использование атрибутов `defer` и `async`.

   Задание: Подключить файл `script.js` к HTML-документу и вывести сообщение «Скрипт подключен» в консоль браузера.

2. **Создание переменных и вывод данных.**

   Ход занятия: Правила именования. Использование `let` и `const`. Методы вывода данных: `console.log()`, `alert()`.

   Задание: Объявить переменные для хранения имени пользователя и года рождения, вывести их значения в консоль.

3. **Работа с различными типами данных.**

   Ход занятия: Изучение строк, чисел, boolean, null и undefined. Определение типа через `typeof`.

   Задание: Создать переменные разных типов и вывести в консоль результат их сложения (конкатенации) со строками.

4. **Выполнение вычислений в JavaScript.**

   Ход занятия: Арифметические операторы (+, -, *, /, %). Инкремент и декремент. Порядок выполнения операций.

   Задание: Написать калькулятор для вычисления площади круга по заданному радиусу.

5. **Реализация логики с использованием if.**

   Ход занятия: Условные операторы и блоки `else`, `else if`. Логические операторы (&&, ||, !).

   Задание: Написать программу, которая проверяет возраст пользователя и выдает разные сообщения для детей, подростков и взрослых.

6. **Использование switch.**

   Ход занятия: Синтаксис `switch`, работа с `case`, `break` и `default`. Сравнение с `if`.

   Задание: Создать программу, которая выводит название месяца по его порядковому номеру.

7. **Использование цикла for.**

Ход занятия: Структура цикла: инициализация, условие, шаг. Обход числовых последовательностей.

Задание: Вывести в консоль все четные числа от 0 до 20.

8. **Использование while и do...while.**

   Ход занятия: Циклы с предусловием и постусловием. Риски бесконечных циклов.

   Задание: Реализовать запрос числа у пользователя через `prompt` до тех пор, пока не будет введено число 10.

9. **Разработка пользовательских функций.**

   Ход занятия: Объявление функций (Declaration). Повторное использование кода. Область видимости.

   Задание: Написать функцию `sayHello`, которая выводит приветствие в консоль.

10. **Использование параметров и возврата значений.**

    Ход занятия: Передача аргументов. Оператор `return`. Анонимные и стрелочные функции.

    Задание: Создать функцию `sum(a, b)`, которая возвращает сумму двух чисел.

11. **Создание и обработка массивов.**

    Ход занятия: Индексация. Свойство `length`. Доступ к элементам массива.

    Задание: Создать массив из названий фруктов и заменить второй элемент на другой.

12. **Использование методов массивов.**

    Ход занятия: Изучение `push`, `pop`, `shift`, `unshift`, `forEach`, `map`, `filter`.

    Задание: Из массива чисел создать новый массив, содержащий только положительные числа (метод `filter`).

13. **Создание объектов.**

    Ход занятия: Структура «ключ-значение». Литералы объектов.

    Задание: Описать объект `car` с полями `brand`, `model` и `year`.

14. **Изменение и получение свойств объектов.**

Ход занятия: Обращение через точку и квадратные скобки. Удаление свойств (`delete`).

Задание: Добавить объекту `user` новое свойство `isAdmin` и изменить значение существующего свойства `name`.

15. **Получение элементов страницы.**

    Ход занятия: Работа с `document.getElementById`, `querySelector`, `querySelectorAll`.

    Задание: Найти на странице кнопку и текстовый абзац по их ID и сохранить их в переменные.

16. **Изменение содержимого элементов.**

    Ход занятия: Свойства `textContent`, `innerHTML`, `innerText`.

    Задание: При загрузке страницы изменить текст в заголовке `h1` на «Добро пожаловать в JS».

17. **Изменение стилей элементов.**

    Ход занятия: Работа со свойством `style`. Манипуляции с классами через `classList` (add, remove, toggle).

    Задание: Реализовать кнопку, которая при нажатии меняет цвет фона страницы.

18. **Создание элементов страницы.**

    Ход занятия: Метод `createElement`. Добавление в DOM через `append`, `prepend`, `appendChild`.

    Задание: Написать скрипт, который создает новый пункт списка `li` и добавляет его в существующий список `ul`.

19. **Удаление элементов страницы.**

    Ход занятия: Методы `remove()`, `removeChild()`. Очистка контейнеров.

    Задание: Реализовать функцию, которая удаляет последний элемент из списка при клике на специальную кнопку.

20. **Обработка кликов мыши.**

    Ход занятия: Слушатель `click`. Объект события `event`. Предотвращение стандартного поведения.

Задание: Создать кнопку-счетчик, которая увеличивает число в абзаце при каждом нажатии.

21. **Обработка событий клавиатуры.**

    Ход занятия: События `keydown`, `keyup`. Определение нажатой клавиши через `event.key`.

    Задание: Реализовать перемещение квадрата по странице с помощью клавиш-стрелок.

22. **Создание интерактивных элементов интерфейса.**

    Ход занятия: Совмещение работы со стилями и событиями. Создание вкладок или аккордеонов.

    Задание: Сверстать и запрограммировать модальное окно (открытие и закрытие по кнопке).

23. **Получение данных формы.**

    Ход занятия: Свойство `value` для текстовых полей, `checked` для чекбоксов. Работа с `FormData`.

    Задание: Считать данные из полей «Логин» и «Пароль» и вывести их в консоль при нажатии кнопки «Войти».

24. **Проверка данных формы.**

    Ход занятия: Валидация типов данных. Обязательные поля. Обработка события `submit`.

    Задание: Запретить отправку формы, если поле «Имя» пустое, и вывести предупреждение.

25. **Реализация проверки ввода пользователя.**

    Ход занятия: Регулярные выражения. Проверка паттернов (email, телефон) в реальном времени.

    Задание: Подсвечивать поле ввода красным, если введенный пароль короче 8 символов.

26. **Использование Promise.**

    Ход занятия: Состояния промиса. Цепочки `.then()` и `.catch()`. Создание собственных промисов.

    Задание: Создать промис, который имитирует задержку загрузки данных (2 секунды) и возвращает успех.

27. **Использование async/await.**

    Ход занятия: Синтаксис асинхронных функций. Обработка ошибок через `try...catch`.

    Задание: Переписать код с промисами из предыдущего задания на использование `async/await`.

28. **Получение данных с сервера.**

    Ход занятия: Использование `Fetch API`. Понятие GET-запроса.

    Задание: Сделать запрос к публичному API и получить массив имен пользователей.

29. Обработка полученных данных.

    Ход занятия: Работа с ответом сервера. Метод `.json()`. Проверка статуса ответа.

    Задание: Вывести в консоль количество полученных с сервера объектов.

30. **Работа с JSON-данными.**

    Ход занятия: Методы `JSON.parse()` и `JSON.stringify()`. Обмен данными.

    Задание: Превратить объект пользователя в строку JSON и сохранить её в переменную.

31. **Сохранение данных в браузере.**

    Ход занятия: Работа с `localStorage` и `sessionStorage`. Лимиты хранилища.

    Задание: Сохранить имя пользователя в `localStorage` при нажатии на кнопку «Запомнить меня».

32. **Получение и удаление данных.**

    Ход занятия: Методы `getItem`, `removeItem`, `clear`. Проверка наличия данных.

    Задание: При загрузке страницы проверить наличие имени в `localStorage` и, если оно есть, вывести приветствие.

33. **Разработка интерактивных элементов.**

    Ход занятия: Использование библиотек или чистый JS для сложных UI-компонентов (слайдеры, выпадающие списки).

    Задание: Реализовать фильтрацию списка элементов по мере ввода текста в поисковую строку.

34. **Создание простого веб-виджета.**

    Ход занятия: Сборка логики в единый функциональный блок. Виджет погоды или курса валют.

    Задание: Создать виджет «Список дел» (To-Do List) с возможностью добавления и удаления задач.

35. **Начало разработки клиентского приложения.**

    Ход занятия: Проектирование структуры проекта. Определение функционала. Верстка основного интерфейса.

    Задание: Создать заготовку приложения (каркас) и инициализировать основные переменные состояния.

36. **Продолжение разработки.**

    Ход занятия: Написание функциональных модулей. Инкапсуляция логики.

    Задание: Реализовать логику переключения между экранами или категориями в приложении.

37. **Работа с DOM и событиями.**

    Ход занятия: Настройка динамического обновления контента приложения в ответ на действия пользователя.

    Задание: Реализовать отображение детальной информации об объекте при клике на его карточку.

38. **Добавление работы с API.**

    Ход занятия: Интеграция приложения с реальными данными через серверные запросы.

    Задание: Настроить автоматическое обновление данных в приложении каждые 30 секунд.

39. Сохранение данных пользователя.

    Х**од занятия:** Реализация сохранения прогресса или настроек в `localStorage`.

    Задание: Добавить в приложение возможность сохранять товары в «Избранное» с доступом после перезагрузки.

40. Улучшение структуры кода.

    Ход занятия: Рефакторинг. Удаление дублирования. Комментирование кода. Разделение на функции.

Задание: Переписать повторяющиеся части кода приложения в виде отдельных функций.

41. **Поиск и исправление ошибок.**

    Ход занятия: Тестирование сценариев. Отладка через `debugger`. Проверка кроссбраузерности.

    Задание: Протестировать приложение на ввод некорректных данных и исправить возникающие ошибки.

42. **Демонстрация разработанного приложения.**

    Ход занятия: Презентация проекта. Рассказ о технической реализации и использованных методах.

    Задание: Продемонстрировать работу итогового приложения, показать код и ответить на вопросы по логике его работы.

Методические указания по выполнению курсового проекта

по МДКн.03.03 «Разработка клиентской части информационных ресурсов»

1. Общие положения

Курсовой проект является одной из форм самостоятельной учебной работы студентов и направлен на закрепление теоретических знаний и формирование практических навыков разработки клиентской части информационных ресурсов.

В процессе выполнения курсового проекта студент должен продемонстрировать способность анализировать техническое задание, проектировать интерфейс информационного ресурса, разрабатывать программный код клиентской части и применять современные технологии веб-разработки.

Целью курсового проекта является приобретение практического опыта разработки клиентской части веб-приложений с использованием языков программирования, библиотек и фреймворков.

2. Цели и задачи курсового проекта

Цель курсового проекта

Формирование практических навыков разработки клиентской части информационных ресурсов с использованием современных программных технологий.

Задачи курсового проекта

В ходе выполнения курсового проекта студент должен:

- изучить предметную область разрабатываемого информационного ресурса;
- проанализировать техническое задание;
- разработать структуру и макет пользовательского интерфейса;
- реализовать клиентскую часть веб-приложения;
- использовать современные библиотеки и фреймворки;
- реализовать взаимодействие клиентской части с сервером;
- обеспечить обработку пользовательских действий;
- выполнить тестирование и отладку программного кода;
- подготовить документацию по разработанному программному продукту.

3. Формируемая профессиональная компетенция

В ходе выполнения курсового проекта формируется следующая профессиональная компетенция:

ПК 3.4. Создавать программный код на стороне клиента в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков.

Студент должен уметь:

- разрабатывать пользовательские интерфейсы веб-приложений;
- применять языки разметки и программирования для клиентской разработки;
- использовать библиотеки и фреймворки для создания интерактивных интерфейсов;
- реализовывать обработку пользовательских событий;
- обеспечивать взаимодействие клиентской части с серверными сервисами.

4. Тематика курсовых проектов

Тематика курсовых проектов должна быть связана с разработкой интерфейсов информационных ресурсов и веб-приложений.

Примеры тем:

- разработка пользовательского интерфейса интернет-магазина;
- разработка клиентской части системы управления контентом;
- разработка интерфейса личного кабинета пользователя;
- разработка веб-интерфейса системы бронирования;
- разработка интерфейса информационного портала;
- разработка клиентской части веб-сервиса.

Тема курсового проекта согласовывается и утверждается преподавателем.

5. Этапы выполнения курсового проекта

Работа над курсовым проектом включает несколько этапов.

1. Анализ технического задания

На данном этапе студент:

- изучает требования к информационному ресурсу;
- определяет функциональные возможности интерфейса;
- анализирует целевую аудиторию.

2. Проектирование пользовательского интерфейса

Включает:

- разработку структуры веб-страниц;
- создание прототипов интерфейса;
- проектирование навигации;
- разработку макета страниц.

3. Разработка клиентской части

На данном этапе выполняется:

- создание HTML-структуры веб-страниц;
- разработка стилей оформления с использованием CSS;
- программирование интерактивных элементов с использованием JavaScript;
- использование библиотек и фреймворков.

4. Реализация взаимодействия с сервером

Включает:

- отправку запросов к серверу;
- получение и обработку данных;
- отображение данных в пользовательском интерфейсе.

5. Тестирование и отладка

Проводится:

- проверка корректности работы интерфейса;
- тестирование обработки пользовательских действий;
- исправление выявленных ошибок;
- оптимизация работы приложения.

6. Подготовка пояснительной записки

Студент оформляет документацию, содержащую описание разработанного проекта.

6. Структура пояснительной записки

Пояснительная записка должна включать следующие разделы:

1. Титульный лист
2. Задание на курсовой проект
3. Содержание
4. Введение
5. Анализ предметной области
6. Проектирование пользовательского интерфейса
7. Реализация клиентской части информационного ресурса
8. Тестирование и отладка программного кода
9. Заключение
10. Список используемых источников
11. Приложения (листинг программного кода, макеты интерфейса, схемы)

7. Требования к программному продукту

Разрабатываемая клиентская часть информационного ресурса должна:

- соответствовать техническому заданию;
- обеспечивать удобный пользовательский интерфейс;
- поддерживать интерактивное взаимодействие с пользователем;
- корректно отображаться в современных веб-браузерах;
- обеспечивать взаимодействие с серверной частью приложения.

Допускается использование следующих технологий:

- **HTML**
- **CSS**
- **JavaScript**
- **TypeScript**
- **React**
- **Angular**
- **Vue.js**
- **Bootstrap**
- **Tailwind CSS**

8. Критерии оценки курсового проекта

Критерии	Максимальный балл
Соответствие техническому заданию	20
Качество интерфейса	20
Качество программного кода	15
Реализация функциональности	15
Использование библиотек и фреймворков	10
Тестирование и отладка	10
Оформление пояснительной записки	5
Защита проекта	5

9. Защита курсового проекта

Защита курсового проекта проводится в форме презентации разработанного информационного ресурса.

Студент должен:

- представить разработанный интерфейс;
- объяснить структуру и функциональность приложения;
- продемонстрировать работу основных функций;
- ответить на вопросы преподавателя.

Примерные вопросы к экзамену по МДКн 03.02 Верстка страниц

1. Понятие веб-верстки и её роль в разработке веб-приложений.
2. Структура HTML-документа.
3. Основные теги HTML и их назначение.

4. Текстовые элементы HTML.
5. Списки в HTML и их виды.
6. Гиперссылки и их атрибуты.
7. Работа с изображениями в HTML.
8. Таблицы в HTML: структура и основные элементы.
9. Формы HTML и их назначение.
10. Основные элементы форм.
11. Понятие CSS и его назначение.
12. Способы подключения CSS к HTML-документу.
13. Селекторы CSS и их виды.
14. Каскадность и приоритет стилей.
15. Блочная модель CSS.
16. Свойства отступов и границ элементов.
17. Работа с цветом и фоном в CSS.
18. Оформление текста с помощью CSS.
19. Позиционирование элементов на странице.
20. Использование display в CSS.
21. Основы Flexbox.
22. Основные свойства Flexbox.
23. Основы CSS Grid.
24. Создание сетки с помощью Grid.
25. Понятие адаптивной верстки.
26. Использование медиа-запросов.
27. Кроссбраузерная верстка.
28. Методология БЭМ.
29. Оптимизация CSS-кода.
30. Основные ошибки при верстке веб-страниц.

Примерные задачи к экзамену по МДКн 03.02 Верстка страниц

1. Создать HTML-страницу с базовой структурой документа.
2. Создать страницу с использованием заголовков, абзацев и списков.
3. Добавить изображения и оформить их с помощью CSS.
4. Создать страницу с гиперссылками.
5. Разработать таблицу с данными и оформить её стилями.
6. Создать HTML-форму для ввода данных пользователя.
7. Оформить форму с использованием CSS.
8. Создать страницу с подключением внешнего CSS-файла.
9. Использовать различные типы CSS-селекторов.
10. Оформить текст с помощью CSS-свойств.
11. Создать блоковую структуру страницы.
12. Настроить отступы элементов с использованием margin и padding.
13. Создать навигационное меню сайта.
14. Оформить страницу с использованием фоновых изображений.
15. Создать карточки контента (например, карточки товаров).
16. Сверстать шапку сайта (header).
17. Сверстать подвал сайта (footer).
18. Создать двухколоночный макет страницы.

19. Реализовать макет страницы с использованием Flexbox.
20. Разработать макет страницы с использованием CSS Grid.
21. Сверстать страницу по заданному макету.
22. Реализовать адаптивную верстку страницы.
23. Добавить медиа-запросы для мобильных устройств.
24. Реализовать адаптивное меню сайта.
25. Выполнить выравнивание элементов с использованием Flexbox.
26. Создать сеточную структуру сайта с использованием Grid.
27. Проверить страницу на кроссбраузерность.
28. Оптимизировать CSS-код страницы.
29. Исправить ошибки верстки в готовом HTML-файле.
30. Сверстать простую страницу сайта по предложенному дизайну.

Примерные вопросы к экзамену по МДКн 03.03 Разработка клиентской части информационных ресурсов

1. Назначение клиентской части веб-приложения.
2. Роль языка JavaScript в веб-разработке.
3. Основные типы данных в JavaScript.
4. Переменные и способы их объявления.
5. Арифметические и логические операторы в JavaScript.
6. Условные конструкции в JavaScript.
7. Циклы и их виды.
8. Функции в JavaScript и их назначение.
9. Область видимости переменных.
10. Массивы и методы работы с массивами.
11. Объекты в JavaScript.
12. Структура и назначение DOM.
13. Методы доступа к элементам DOM.
14. Изменение содержимого HTML-элементов с помощью JavaScript.
15. Изменение стилей элементов страницы.
16. События в JavaScript.
17. Обработчики событий.
18. Работа с формами на стороне клиента.
19. Проверка (валидация) данных форм.
20. Асинхронное программирование в JavaScript.
21. Promise и их использование.
22. Использование async/await.
23. Fetch API и его назначение.
24. Работа с JSON-данными.
25. Получение данных с сервера.
26. LocalStorage и SessionStorage.
27. Создание интерактивных элементов интерфейса.
28. Отладка JavaScript-кода.
29. Оптимизация клиентского кода.
30. Основные ошибки при разработке клиентской части веб-приложений.

Примерные задачи к экзамену по МДКн 03.03 Разработка клиентской части информационных ресурсов

1. Подключить JavaScript-файл к HTML-странице.
2. Создать программу для вывода текста на страницу.
3. Реализовать программу с использованием переменных и арифметических операций.
4. Реализовать условную конструкцию для проверки значения переменной.
5. Создать программу с использованием цикла.
6. Написать функцию, выполняющую вычисление.
7. Создать массив и выполнить операции над его элементами.
8. Создать объект и вывести его свойства на страницу.
9. Получить элемент страницы через DOM.
10. Изменить текст элемента страницы через JavaScript.
11. Изменить стиль элемента страницы.
12. Создать новый элемент DOM и добавить его на страницу.
13. Удалить элемент со страницы.
14. Реализовать обработчик события клика.
15. Реализовать обработку событий клавиатуры.
16. Получить данные из HTML-формы.
17. Реализовать проверку данных формы.
18. Реализовать динамическое добавление элементов списка.
19. Создать интерактивную кнопку.
20. Реализовать простую анимацию элемента.
21. Сохранить данные пользователя в LocalStorage.
22. Получить данные из LocalStorage и вывести на страницу.
23. Отправить запрос к API с использованием Fetch.
24. Обработать полученные JSON-данные.
25. Вывести данные из API на страницу.
26. Реализовать динамическое обновление данных страницы.
27. Реализовать простой интерактивный калькулятор.
28. Реализовать список задач (ToDo list).
29. Найти и исправить ошибки в JavaScript-коде.
30. Разработать простое клиентское веб-приложение с использованием JavaScript.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕУЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

Рабочая программа профессионального модуля

ПМн.04 «Разработка веб приложения на стороне сервера»

для специальности

09.02.09 «Веб-разработка»

Екатеринбург

2026

Рабочая программа ПМн.04 «Разработка веб приложения на стороне сервера» разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.09 «Веб-разработка».

Организация-разработчик: ГАПОУ СО «Екатеринбургский монтажный колледж»

Разработчик Софьина Н.А. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», ВКК; Кузьмин В.П. - преподаватель ГАПОУ СО «Екатеринбургский монтажный колледж», 1КК;,

Рассмотрена методическим объединением Информационных технологий

Руководитель МО _____Софьина Н.А.

УТВЕРЖДАЮ

Заместитель директора по учебной работе

_____ Л.С.Хоринова

«13» марта 2026

СОДЕРЖАНИЕ

1. **ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
2. **СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
3. **УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**
4. **КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ**

1. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
«ПМн.04 Разработка веб приложения на стороне сервера»

1.1. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающихся должен освоить основной вид деятельности ВД.3 «Разработка веб приложения на стороне сервера» и соответствующие ему общие компетенции и профессиональные компетенции:

1.1.1. Перечень общих компетенций

Код	*Наименование общих компетенций*
ОК 01	*Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;*
ОК 02	*Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;*
ОК 05	*Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;*
ОК 09	*Пользоваться профессиональной документацией на государственном и иностранном языках*

1.1.2. Перечень профессиональных компетенций

Код	*Наименование видов деятельности и профессиональных компетенций*
ВД.3	Разработка веб приложения на стороне сервера
ПК 3.1.	Администрировать среды и платформы разработки информационных ресурсов
ПК 3.2.	Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков
ПК 3.3.	Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Владеть навыками	В разработке веб приложения на стороне сервера
Уметь	• применять выбранные языки программирования для написания программного кода; • использовать выбранную среду программирования и средства системы управления базами данных; • применять инструменты для тестирования программных модулей; • использовать возможности имеющейся программной архитектуры информационного ресурса. • использовать серверную инфраструктуру и средства виртуализации • владеть методами поиска информации по специальности, уметь выбирать необходимые технические средства и системы при решении конкретных задач и проблем
Знать	• отраслевую нормативную техническую документацию; • особенности выбранной среды программирования; • сетевые протоколы и основ web-технологий; • основы информационной безопасности web-ресурсов. • методы повышения читаемости программного кода; • синтаксис выбранного языка программирования, особенностей программирования на этом языке, стандартных библиотек языка программирования;

1.2. Количество часов, отводимое на освоение профессионального модуля

Всего часов: 588;

в том числе в форме практической подготовки: 204 часов;

Из них:

- на освоение МДКн.04.01. Администрирование сред и платформ разработки информационных ресурсов: 84 часа;

в том числе в форме практической подготовки: 40 часов

- на освоение МДКн.04.02. Разработка кода информационных ресурсов: 132 часа;

в том числе в форме практической подготовки: 74 часов;

- на освоение МДКн.04.03. Разработка информационных ресурсов с использованием программных платформ: 156 часов;

в том числе в форме практической подготовки: 90 часов;

практики, в том числе учебная: 36 часов;

производственная: 180 часов.

Промежуточная аттестация _____.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1. Структура профессионального модуля

Коды профессиональных общих компетенций	Наименования разделов профессионального модуля	Всего, час.	В т.ч. в форме практической.	Объем профессионального модуля, ак. час.						
				Обучение по МДК					Практики	
				Всего	В том числе					
					Лабораторных. и практических. занятий	Курсовых работ (проектов)	Самостоятельная работа	Промежуточная аттестация	Учебная	Производственная
1	2	3	4	5	6	7	8	9	10	11
ПК 3.1 ОК 01 ОК 02 ОК 05 ОК 09	Раздел 1. Администрирование сред и платформ разработки информационных ресурсов	**84**	40	**84**	40	X	14	X	**36**	**X**
ПК 3.2 ОК 01 ОК 02 ОК 05 ОК 09	Раздел 2. Разработка кода информационных ресурсов	**132**	74	**132**	74	X	24			**X**
ПК 3.2 ПК 3.3 ОК 01 ОК 02 ОК 05 ОК 09	Раздел 3. Разработка информационных ресурсов с использованием программных платформ	**156**	90	**156**	90	30	30			**X**
ПК 3.1 ПК 3.2 ПК 3.3 ОК 01 ОК 02 ОК 05 ОК 09	Производственная практика (по профилю специальности), часов	**108**	X							**180**
	Промежуточная аттестация	**X**	X							
	Всего:	**588**	**204**	**588**	**204**	**30**	**68**	**X**	**36**	**180**

2.2. Тематический план и содержание профессионального модуля (ПМ)

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная учебная работа обучающихся, курсовой проект	Объем, акад. ч / в том числе в форме практической подготовки, акад ч
1	**2**	3
Раздел 1. Администрирование сред и платформ разработки информационных ресурсов		
МДКн.04.01. Администрирование сред и платформ разработки информационных ресурсов		
Тема 1.1. Администрирование сред и платформ разработки информационных ресурсов	**Содержание**	**30**
	Понятие и назначение сред разработки информационных ресурсов	2
	Архитектура среды разработки веб-приложений	2
	Основные компоненты платформ разработки информационных ресурсов	2
	Операционные системы, используемые для серверной разработки	2
	Основы администрирования серверных операционных систем	2
	Среды разработки и их назначение (IDE, редакторы кода)	2
	Установка и настройка среды разработки	2
	Платформы выполнения серверных приложений	2
	Веб-серверы и их роль в инфраструктуре веб-разработки	2
	Установка и конфигурирование веб-сервера	2
	Инструменты управления зависимостями и пакетами	2
	Системы контроля версий в процессе разработки	2

	Основы безопасности серверных сред разработки	2
	Управление конфигурациями и настройками среды разработки	2
	Резервное копирование и восстановление серверных сред	2
	В том числе практических занятий и лабораторных работ	**40**
	Установка и настройка среды разработки для веб-приложений	2
	Настройка редактора кода и инструментов разработчика	2
	Установка серверной платформы выполнения приложений	2
	Настройка переменных окружения среды разработки	2
	Установка и настройка локального веб-сервера	2
	Конфигурирование Apache / Nginx	2
	Развертывание тестового веб-проекта	2
	Настройка структуры каталогов веб-проекта	2
	Установка и настройка системы управления базами данных	2
	Настройка подключения веб-приложения к базе данных	2
	Установка системы контроля версий Git	2
	Создание локального репозитория проекта	2
	Работа с удалённым репозиторием (GitHub / GitLab)	2
	Настройка совместной работы над проектом	2
	Настройка прав доступа к серверной среде	2
	Конфигурирование параметров веб-сервера	2

	Настройка журналирования и мониторинга	2
	Настройка параметров безопасности серверной среды	2
	Создание резервной копии проекта	2
	Восстановление среды разработки из резервной копии	2
Тематика самостоятельной учебной работы при изучении раздела № 1 1. Изучение документации среды разработки 2. Анализ современных платформ серверной разработки 3. Изучение архитектуры веб-сервера 4. Изучение документации системы Git 5. Подготовка схемы серверной инфраструктуры веб-проекта 6. Изучение методов обеспечения безопасности серверной среды **7.** Подготовка отчета по настройке среды разработки		**14**
Раздел 2. Разработка кода информационных ресурсов		
МДКн.04.02. Разработка кода информационных ресурсов		
Тема 2.1. Виды и назначения серверных языков программирования, среды разработки.	**Содержание**	**18**
	Понятие серверного программирования и его роль в разработке информационных ресурсов	2
	Архитектура клиент-серверных приложений	2
	Виды серверных языков программирования	2
	Назначение и особенности серверных языков программирования	2
	Популярные серверные языки программирования (PHP, Python, JavaScript, Java)	2
	Сравнительный анализ серверных языков программирования	2
	Среды разработки серверных приложений	2

	Инструменты разработки и тестирования серверного кода	2
	Управление зависимостями и пакетами в серверных проектах	2
	В том числе практических и лабораторных занятий	**36**
	Установка среды разработки для серверного программирования	2
	Настройка проекта серверного приложения	2
	Создание первого серверного приложения	2
	Работа с HTTP-запросами	2
	Обработка параметров запроса	2
	Формирование HTTP-ответов	2
	Работа с маршрутами приложения	2
	Разработка серверной логики приложения	2
	Работа с формами и отправкой данных	2
	Обработка пользовательского ввода	2
	Реализация механизма сессий	2
	Работа с cookie	2
	Подключение базы данных к серверному приложению	2
	Выполнение запросов к базе данных	2
	Добавление данных в базу	2
	Извлечение данных из базы данных	2
	Обновление данных в базе	2

	Удаление данных из базы	2
Тема 2.2. Программирование на стороне сервера	**Содержание**	**16**
	Основы разработки серверного кода	2
	Структура серверного приложения	2
	Обработка HTTP-запросов на стороне сервера	2
	Формирование HTTP-ответов	2
	Работа с параметрами запроса и формами	2
	Работа с сессиями и cookie	2
	Взаимодействие серверного приложения с базами данных	2
	Подключение базы данных к серверному приложению	2
	В том числе практических и лабораторных занятий	**38**
	Разработка API для веб-приложения	2
	Работа с JSON-данными	2
	Реализация обработки ошибок	2
	Логирование работы приложения	2
	Тестирование серверного приложения	2
	Оптимизация серверного кода	2
	Организация структуры серверного проекта	2
	Разработка серверного модуля обработки данных	2
	Реализация авторизации пользователей	2

	Реализация аутентификации пользователей	2
	Разработка REST API	2
	Обработка асинхронных запросов	2
	Использование сторонних библиотек	2
	Интеграция серверного приложения с внешними сервисами	2
	Подготовка серверного приложения к развертыванию	2
	Тестирование и отладка серверного приложения	2
	Итоговая практическая работа по разработке серверного приложения	2
	Анализ и оптимизация разработанного серверного приложения	2
	Подготовка отчета по практическим работам	2
Тематика самостоятельной учебной работы при изучении раздела №2 1. Изучение документации серверных языков программирования 2. Анализ архитектуры серверных веб-приложений 3. Изучение методов обработки HTTP-запросов 4. Изучение способов взаимодействия серверного приложения с базами данных 5. Анализ технологий разработки REST API **6.** Подготовка отчета по разработке серверного приложения		**24**
Раздел 3. Разработка информационных ресурсов с использованием программных платформ		
МДКн.04.03. Разработка информационных ресурсов с использованием программных платформ		
Тема 3.1. Разработка информационных ресурсов с использованием фреймворков и библиотек	**Содержание**	**36**
	Понятие программных платформ разработки информационных ресурсов	2
	Назначение и виды фреймворков и библиотек	2

	Архитектура современных веб-фреймворков	2
	Принципы работы фреймворков MVC	2
	Использование библиотек в разработке веб-приложений	2
	Преимущества использования фреймворков в разработке	2
	Обзор популярных серверных фреймворков	2
	Обзор популярных клиентских фреймворков	2
	Структура проекта, разработанного с использованием фреймворка	2
	Настройка окружения для работы с фреймворками	2
	Организация маршрутизации в веб-приложениях	2
	Работа с шаблонизаторами	2
	Работа с контроллерами и моделями	2
	Интеграция библиотек в веб-приложение	2
	Обеспечение безопасности веб-приложений	2
	Тестирование веб-приложений, разработанных с использованием фреймворков	2
	Оптимизация производительности веб-приложений	2
	Подготовка веб-приложений к развертыванию	2
	В том числе практических и лабораторных занятий	**90**
	Установка и настройка программной платформы разработки	2
	Создание проекта с использованием фреймворка	2
	Настройка структуры проекта	2

	Настройка маршрутизации в веб-приложении	2
	Реализация контроллеров	2
	Реализация моделей данных	2
	Работа с шаблонизаторами	2
	Реализация пользовательского интерфейса	2
	Подключение библиотек к проекту	2
	Использование сторонних библиотек	2
	Разработка функциональных модулей приложения	2
	Реализация обработки пользовательских запросов	2
	Работа с формами и вводом данных	2
	Реализация валидации данных	2
	Интеграция веб-приложения с базой данных	2
	Реализация CRUD-операций	2
	Реализация механизма авторизации пользователей	2
	Реализация механизма аутентификации пользователей	2
	Разработка REST API с использованием фреймворка	2
	Работа с JSON-данными	2
	Обработка ошибок приложения	2
	Реализация логирования	2
	Тестирование функциональности веб-приложения	2

	Отладка веб-приложения	2
	Оптимизация структуры проекта	2
	Оптимизация производительности веб-приложения	2
	Подготовка веб-приложения к развертыванию	2
	Настройка конфигурации приложения	2
	Развертывание веб-приложения на сервере	2
	Настройка серверной среды для работы приложения	2
	Интеграция приложения с внешними сервисами	2
	Использование API сторонних сервисов	2
	Настройка безопасности веб-приложения	2
	Реализация защиты от распространенных веб-уязвимостей	2
	Тестирование безопасности приложения	2
	Подготовка технической документации проекта	2
	Итоговая практическая работа по разработке веб-приложения	2
	Тестирование итогового проекта	2
	Анализ разработанного веб-приложения	2
	Оптимизация итогового проекта	2
	Подготовка отчета по практическим работам	2
	Защита разработанного веб-проекта	2
	Итоговая доработка веб-проекта	2

	Анализ результатов разработки	2
	Подготовка итогового отчета	2
Тематика самостоятельной учебной работы при изучении раздела №3 1. Изучение документации популярных веб-фреймворков 2. Анализ архитектуры веб-приложений, разработанных с использованием фреймворков 3. Изучение библиотек для разработки веб-интерфейсов 4. Анализ технологий разработки REST API **5.** Подготовка отчета по разработке веб-приложения		**30**
Курсовой проект		
Примерная тематика курсовых проектов 1. Разработка информационного веб-ресурса для управления задачами с использованием веб-фреймворка. 2. Разработка системы управления контентом (CMS) на основе серверного фреймворка. 3. Разработка веб-приложения для учета клиентов организации с использованием программной платформы. 4. Разработка информационной системы управления заказами интернет-магазина с использованием фреймворка. 5. Разработка веб-приложения для бронирования услуг с использованием программной платформы. 6. Разработка информационного портала организации с использованием современного веб-фреймворка. 7. Разработка системы управления пользователями веб-ресурса с использованием серверного фреймворка. 8. Разработка веб-приложения для публикации новостей с использованием программной платформы. 9. Разработка системы управления учебными материалами с использованием веб-фреймворка. 10. Разработка веб-приложения для учета заявок пользователей с использованием программной платформы. 11. Разработка REST API для информационного ресурса с использованием серверного фреймворка. 12. Разработка веб-приложения для управления событиями и мероприятиями с использованием фреймворка. 13. Разработка информационной системы учета товаров склада с использованием программной платформы. 14. Разработка веб-приложения для ведения блога с использованием фреймворка. 15. Разработка системы онлайн-регистрации пользователей с использованием веб-платформы. 16. Разработка веб-приложения для управления проектами с использованием программного фреймворка.		

17. Разработка информационного ресурса для публикации объявлений с использованием веб-фреймворка. 18. Разработка веб-приложения для ведения каталога продукции с использованием программной платформы. 19. Разработка системы управления заказами службы доставки с использованием веб-фреймворка. 20. Разработка веб-ресурса для организации онлайн-опросов с использованием программной платформы. 21. Разработка системы управления пользователями и ролями доступа на основе веб-фреймворка. 22. Разработка веб-приложения для учета посещаемости мероприятий с использованием программной платформы. 23. Разработка информационной системы учета заявок технической поддержки с использованием фреймворка. 24. Разработка веб-приложения для управления библиотечным фондом с использованием программной платформы. 25. Разработка веб-сервиса для публикации и хранения файлов с использованием фреймворка. 26. Разработка информационного ресурса для онлайн-обучения с использованием программной платформы. 27. Разработка веб-приложения для управления расписанием занятий с использованием фреймворка. 28. Разработка веб-приложения для учета сотрудников организации с использованием программной платформы. 29. Разработка веб-приложения для управления задачами команды с использованием фреймворка. **30.** Разработка информационной системы управления заявками клиентов с использованием программной платформы.	
Обязательные аудиторные учебные занятия по курсовому проекту 1. Выбор темы курсового проекта и постановка задачи разработки информационного ресурса 2. Анализ предметной области и формирование требований к информационному ресурсу 3. Разработка структуры информационного ресурса 4. Проектирование архитектуры веб-приложения 5. Выбор программной платформы, фреймворков и библиотек для разработки 6. Проектирование базы данных информационного ресурса 7. Разработка структуры проекта и настройка среды разработки 8. Реализация основных функциональных модулей информационного ресурса 9. Реализация взаимодействия с базой данных 10. Реализация пользовательского интерфейса информационного ресурса 11. Реализация механизмов обработки пользовательских запросов 12. Тестирование разработанного информационного ресурса	**30**

13. Подготовка пояснительной записки курсового проекта 14. Подготовка презентации курсового проекта 15. Защита курсового проекта	
Учебная практика Разработка веб приложения на стороне сервера **Виды работ** 1. Изучение технического задания на разработку серверной части веб-приложения. 2. Анализ предметной области и определение функциональных требований к серверной части приложения. 3. Выбор программной платформы, языка программирования и серверного фреймворка для разработки веб-приложения. 4. Настройка среды разработки и инструментов программирования для создания серверного приложения. 5. Проектирование архитектуры серверной части веб-приложения. 6. Разработка структуры серверного проекта и программных модулей. 7. Реализация серверной логики обработки клиентских запросов. 8. Разработка API для взаимодействия клиентской и серверной частей веб-приложения. 9. Реализация механизмов взаимодействия серверного приложения с базой данных. 10. Создание программных модулей для обработки, хранения и передачи данных. 11. Реализация функций регистрации, авторизации и аутентификации пользователей (при необходимости). 12. Реализация механизмов валидации и обработки пользовательских данных. 13. Проведение тестирования программных модулей серверной части приложения. 14. Выполнение отладки программного кода на уровне отдельных модулей. 15. Проверка межмодульного взаимодействия компонентов серверного приложения. 16. Анализ и устранение ошибок программного кода. 17. Оптимизация серверного кода и структуры приложения. 18. Подготовка документации по разработанному серверному приложению. **19.** Подготовка отчёта по результатам выполнения учебной практики.	**36**
Производственная практика **Виды работ** 1. Ознакомление со структурой организации, правилами внутреннего распорядка и техникой безопасности на предприятии. 2. Изучение используемых в организации информационных систем, программных платформ и технологий разработки.	**180**

3. Анализ технического задания на разработку или доработку серверной части веб-приложения. 4. Изучение архитектуры существующего программного продукта и структуры программного проекта. 5. Настройка рабочей среды разработки, системы управления версиями и инструментов командной разработки. 6. Разработка программных модулей серверной части веб-приложения в соответствии с техническим заданием. 7. Использование библиотек и фреймворков для реализации серверной логики приложения. 8. Реализация обработки HTTP-запросов и формирования ответов сервера. 9. Разработка и реализация API для взаимодействия серверной части с клиентскими приложениями. 10. Реализация взаимодействия серверного приложения с базой данных. 11. Разработка модулей обработки, хранения и передачи данных. 12. Реализация механизмов регистрации, авторизации и аутентификации пользователей. 13. Реализация механизмов проверки и обработки пользовательских данных (валидация данных). 14. Тестирование разработанных программных модулей серверной части приложения. 15. Выполнение отладки программного кода на уровне программных модулей. 16. Проверка межмодульного взаимодействия компонентов программного продукта. 17. Анализ и устранение ошибок в программном коде. 18. Использование средств логирования и диагностики работы серверного приложения. 19. Оптимизация программного кода и производительности серверного приложения. 20. Подготовка технической документации на разработанные программные модули. 21. Участие в командной разработке программного продукта и работе с системой контроля версий. 22. Подготовка отчёта по результатам прохождения производственной практики.	
Всего	**588**

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

3.1. Для реализации программы профессионального модуля должны быть предусмотрены следующие специальные помещения:

Лаборатория Разработки веб-приложений, оснащенная в соответствии с п. 6.1.2.3 Примерной образовательной программы по специальности 09.02.09 Веб-разработка.

Оснащенные базы практики в соответствии с п 6.1.2.4 примерной образовательной программы по специальности 09.02.09 Веб-разработка.

Реализация образовательной программы предполагает обязательную учебную и производственную практику.

Учебная практика реализуется в мастерских профессиональной образовательной организации и требует наличия оборудования, инструментов, расходных материалов, обеспечивающих выполнение всех видов работ, определенных содержанием программ профессиональных модулей, в том числе оборудования и инструментов, используемых при проведении чемпионатов профессионального мастерства и указанных в инфраструктурных листах конкурсной документации по компетенции «Веб-технологии» (или их аналогов).

Производственная практика реализуется в организациях, обеспечивающих деятельность обучающихся в профессиональной области информационных технологий.

Оборудование предприятий и технологическое оснащение рабочих мест производственной практики должно соответствовать содержанию профессиональной деятельности и дать возможность обучающемуся овладеть профессиональными компетенциями по всем видам деятельности, предусмотренными программой, с использованием современных технологий, материалов и оборудования.

3.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, для использования в образовательном процессе. При формировании библиотечного фонда образовательной организации выбирается не менее одного издания из перечисленных ниже печатных изданий и (или) электронных изданий в качестве основного, при этом список может быть дополнен новыми изданиями.

3.2.1. Основные печатные издания

1. Полуэктова, Н. Р. Разработка веб-приложений: учебное пособие для среднего профессионального образования / Н. Р. Полуэктова. — Москва: Издательство Юрайт, 2024. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-14744-5.

3.2.2. Основные электронные издания

1. Тузовский, А. Ф. Проектирование и разработка web-приложений: учебное пособие для среднего профессионального образования / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2024. — 218 с. — (Профессиональное образование). — ISBN 978-5-534-10017-4. — Текст: электронный// Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/495109. — Режим доступа: для авторизир. пользователей.

3.2.3. Дополнительные источники (при необходимости)

1. Заяц, А.М., Васильев, Н.П. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js: Учебное пособие.

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Раздел 1. Администрирование сред и платформ разработки информационных ресурсов		
ПК 3.1 Администрировать среды и платформы разработки информационных ресурсов	Демонстрация владения навыком администрирования сред и платформы разработки информационных ресурсов	Письменный контроль - ответы на поставленные вопросы, тестирование. Экспертная оценка выполнения практико-ориентированных заданий. Выполнение заданий в рамках группового проекта. Деловые игры. Выполнение и защита заданий самостоятельной внеаудиторной работы. Отчет по производственной практике. Отзывы и характеристики работодателей по итогам производственной практики
ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;	Обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач. Эффективность выбираемых и применяемых методов и способов решения профессиональных задач в области Веб-разработки Своевременность сдачи практических заданий, отчетов по практике. Рациональность распределения времени при выполнении практических работ с соблюдением норм и правил внутреннего распорядка	
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;	Способность самостоятельно и эффективно осуществлять сбор, обработку и интерпретацию информации для решения задач профессиональной деятельности. Широта использования различных источников информации, включая электронные	
ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;	Полнота и аргументированность изложения собственного мнения. Способность взаимодействовать с коллегами.	

ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках	Способность грамотно применять нормативно-правовую базу и профессиональную литературу для решения профессиональных задач. Умение проверять и правильно заполнять формы документации	
Раздел 2. Разработка кода информационных ресурсов		
ПК 3.2 Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков;	Демонстрация владения навыком создания программного кода на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков	Письменный контроль - ответы на поставленные вопросы, тестирование. Экспертная оценка выполнения практико-ориентированных заданий. Выполнение заданий в рамках группового проекта. Деловые игры. Выполнение курсовой работы. Выполнение и защита заданий самостоятельной внеаудиторной работы. Отчет по производственной практике. Отзывы и характеристики работодателей по итогам производственной практики
ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;	Обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач. Эффективность выбираемых и применяемых методов и способов решения профессиональных задач в области Веб-разработки Своевременность сдачи практических заданий, отчетов по практике. Рациональность распределения времени при выполнении практических работ с соблюдением норм и правил внутреннего распорядка	
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;	Способность самостоятельно и эффективно осуществлять сбор, обработку и интерпретацию информации для решения задач профессиональной деятельности. Широта использования различных источников информации, включая электронные	

ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;	Полнота и аргументированность изложения собственного мнения. Способность взаимодействовать с коллегами.	
ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках	Способность грамотно применять нормативно-правовую базу и профессиональную литературу для решения профессиональных задач. Умение проверять и правильно заполнять формы документации	
Раздел 3. Разработка информационных ресурсов с использованием программных платформ		
ПК 3.2 Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков;	Демонстрация владения навыком создавания программного кода на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков;	Письменный контроль - ответы на поставленные вопросы, тестирование. Экспертная оценка выполнения практико-ориентированных заданий. Выполнение заданий в рамках группового проекта. Деловые игры. Выполнение курсовой работы. Выполнение и защита заданий самостоятельной внеаудиторной работы. Отчет по производственной практике. Отзывы и характеристики работодателей по итогам производственной практики
ПК 3.3 Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением	Демонстрация владения навыком осуществления отладки программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением	
ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;	Обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач. Эффективность выбираемых и применяемых методов и способов решения профессиональных задач в области Веб-разработки Своевременность сдачи практических заданий, отчетов по практике. Рациональность распределения времени при выполнении практических работ с соблюдением	

	норм и правил внутреннего распорядка	
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;	Способность самостоятельно и эффективно осуществлять сбор, обработку и интерпретацию информации для решения задач профессиональной деятельности. Широта использования различных источников информации, включая электронные	
ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;	Полнота и аргументированность изложения собственного мнения. Способность взаимодействовать с коллегами.	
ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках	Способность грамотно применять нормативно-правовую базу и профессиональную литературу для решения профессиональных задач. Умение проверять и правильно заполнять формы документации	

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МДКн.04.01 АДМИНИСТРИРОВАНИЕ СРЕД И ПЛАТФОРМ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ РЕСУРСОВ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.09 «Веб-разработка»** и предназначен для проведения текущего контроля и промежуточной аттестации по дисциплине **МДК.04.01 «Администрирование сред и платформ разработки информационных ресурсов»**.

ФОС представляет собой комплекс теоретических вопросов и практических кейсов, имитирующих реальные задачи системного администратора и DevOps-инженера в контексте веб-индустрии.

2. Цель и задачи курса

Цель: Подготовка квалифицированного специалиста, способного самостоятельно проектировать, развертывать и поддерживать инфраструктуру для разработки и публикации веб-приложений.

Задачи:

- **Изучение архитектурных решений:** Освоение принципов работы веб-стеков (напр. LEMP/LAMP), контейнеризации и облачных платформ.
- **Администрирование ОС:** Овладение навыками работы в серверных операционных системах (преимущественно семейства Linux) через интерфейс командной строки (CLI).
- **Управление веб-службами:** Глубокая настройка веб-серверов (Nginx/Apache), серверов приложений и СУБД.
- **Автоматизация и контроль:** Внедрение систем контроля версий (Git) и инструментов управления зависимостями в рабочий процесс.
- **Обеспечение отказоустойчивости:** Настройка мониторинга, безопасности и систем резервного копирования.

3. Структура оценочных материалов

Содержание ФОС коррелирует с этапами жизненного цикла веб-разработки:

- **Модуль «Локальное окружение разработчика»:** Оценка умения подготавливать рабочее место, настраивать IDE, линтеры и переменные среды.
- **Модуль «Серверная инфраструктура»:** Контроль навыков развертывания веб-серверов, конфигурирования виртуальных хостов и настройки защищенного соединения (SSL/HTTPS).
- **Модуль «Управление данными и версиями»:** Практические задания по интеграции веб-приложений с базами данных и организации командной разработки через удаленные репозитории (GitHub/GitLab).
- **Модуль «Безопасность и эксплуатация»:** Проверка компетенций в области разграничения прав доступа, защиты серверных сред и аварийного восстановления систем.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** архитектуру веб-серверов; методы защиты сетевых соединений; принципы хранения и миграции данных; особенности различных серверных ОС.
- **Уметь:** настраивать проксирование запросов; управлять пакетами и зависимостями; делегировать права доступа; автоматизировать рутинные задачи администрирования.
- **Владеть:** навыками работы с консольными утилитами администрирования; инструментарием Git; методиками мониторинга логов и производительности.

5. Методические рекомендации по оцениванию

Оценивание строится на принципах **демонстрационного экзамена**:

1. **Теоретическая часть:** Оценивается владение терминологией и понимание логики работы сетевых протоколов и серверных служб.
2. **Практическая часть:** Оценивается по факту работоспособности настроенной среды. Ошибки в конфигурации, приводящие к неработоспособности веб-ресурса или уязвимости системы, являются критическими.

Краткое изложение планов хода лекционных занятий:

1. Понятие и назначение сред разработки информационных ресурсов

- **О чем занятие:** Роль специализированных инструментов в жизненном цикле программного обеспечения. Разница между простым текстовым редактором и профессиональной средой.
- **Ход занятия:**
  - Эволюция инструментов разработки: от блокнотов до облачных IDE.
  - Составные части среды: редакторы, компиляторы, отладчики и системы сборки.
  - Критерии выбора среды под конкретный технологический стек.
  - Современные тренды: интеграция ИИ-ассистентов в процесс написания кода.

2. Архитектура среды разработки веб-приложений

- **О чем занятие:** Внутренняя структура веб-проекта. Взаимодействие локальной машины разработчика, тестовых серверов и рабочей среды.
- **Ход занятия:**
  - Клиент-серверная модель и базовые протоколы передачи данных.
  - Принципы архитектуры веб-приложений (многослойность, разделение ответственности).
  - Различия между монолитной и микросервисной архитектурами в контексте разработки.
  - Путь сетевого запроса: от ввода URL в браузере до обработки на сервере.

3. Основные компоненты платформ разработки

- **О чем занятие:** «Строительные блоки» современных платформ: языки программирования, библиотеки, СУБД и системы кеширования.
- **Ход занятия:**
  - Понятие технологического стека (на примере LAMP, MERN).
  - Роль внешних API и сторонних сервисов в разработке.
  - Инструменты для документирования и первичного тестирования систем.
  - Взаимодействие компонентов внутри платформы.

4. Операционные системы для серверной разработки

- **О чем занятие:** Почему Linux является стандартом для серверов. Обзор и сравнение популярных дистрибутивов.
- **Ход занятия:**
  - Классификация серверных ОС: семейства Debian, RHEL и возможности Windows Server.
  - Понятие ядра ОС, системных оболочек (Shell) и файловой иерархии.
  - Сравнение подходов виртуализации и контейнеризации.
  - Выбор ОС в зависимости от требований проекта.

5. Основы администрирования серверных ОС

- **О чем занятие:** Базовые навыки работы в консоли сервера. Управление правами доступа и контроль системных ресурсов.

- **Ход занятия:**
    - Основные команды терминала для работы с файлами и поиска информации.
    - Управление пользователями, группами и правами доступа к данным.
    - Мониторинг загрузки процессора, оперативной памяти и дискового пространства.
    - Методы анализа системных журналов (логов) для поиска ошибок.

6. Среды разработки и их назначение (IDE, редакторы кода)

- **О чем занятие:** Анализ рынка инструментов разработки. Сравнение функциональности мощных IDE и легких текстовых редакторов.
- **Ход занятия:**
    - Ключевые функции IDE: статический анализ, рефакторинг, навигация по коду.
    - Экосистема расширений (на примере VS Code).
    - Оптимизация рабочего пространства: горячие клавиши и сниппеты.
    - Специфические инструменты для фронтенд- и бэкенд-разработчиков.

7. Установка и настройка среды разработки

- **О чем занятие:** Пошаговое развертывание рабочего окружения на локальном компьютере.
- **Ход занятия:**
    - Установка интерпретаторов и сред исполнения (Node.js, Python и др.).
    - Настройка переменных окружения и работа с системным путем (PATH).
    - Конфигурация внутреннего терминала и визуального стиля редактора.
    - Проверка корректности установки инструментов.

8. Платформы выполнения серверных приложений

- **О чем занятие:** Технологии, обеспечивающие работу кода на стороне сервера. Среды исполнения и контейнеры.
- **Ход занятия:**
    - Отличия среды выполнения (Runtime) от скомпилированного кода.
    - Принципы изоляции приложений для обеспечения стабильности.
    - Введение в концепцию Docker-контейнеров.
    - Управление жизненным циклом серверных процессов.

9. Веб-серверы и их роль в инфраструктуре

- **О чем занятие:** Механизмы обработки HTTP-запросов. Назначение Nginx и Apache в современной веб-разработке.
- **Ход занятия:**
    - Принципы обслуживания статического и динамического контента.
    - Использование веб-сервера как обратного прокси и балансировщика нагрузки.
    - Сравнение архитектурных подходов популярных веб-серверов.
    - Роль веб-сервера в обеспечении безопасности приложения.

10. Установка и конфигурирование веб-сервера

- **О чем занятие:** Практическая настройка веб-сервера для запуска реального сайта.

- **Ход занятия:**
    - Структура конфигурационных файлов и работа с виртуальными хостами.
    - Настройка защищенного соединения (HTTPS) и SSL-сертификатов.
    - Параметры оптимизации: сжатие данных и правила кэширования.
    - Диагностика типичных ошибок конфигурации.

11. Инструменты управления зависимостями и пакетами

- **О чем занятие:** Как эффективно управлять сторонними библиотеками в проекте. Использование NPM, Composer или Pip.
- **Ход занятия:**
    - Семантическое версионирование и контроль версий через lock-файлы.
    - Автоматизация обновления библиотек и аудит безопасности.
    - Работа с локальными и удаленными репозиториями пакетов.
    - Создание и публикация собственных модулей.

12. Системы контроля версий в процессе разработки

- **О чем занятие:** Использование Git для отслеживания изменений и совместной работы над кодом.
- **Ход занятия:**
    - Основные операции: создание коммитов, работа с индексацией и ветками.
    - Стратегии слияния веток и разрешение конфликтов в коде.
    - Рабочий процесс в команде: Pull Requests и Code Review.
    - Интеграция с облачными платформами (GitHub, GitLab).

13. Основы безопасности серверных сред разработки

- **О чем занятие:** Защита инфраструктуры и данных от несанкционированного доступа.
- **Ход занятия:**
    - Безопасный удаленный доступ по протоколу SSH.
    - Настройка сетевых экранов (Firewall) и фильтрация трафика.
    - Методы безопасного хранения паролей и ключей доступа.
    - Регулярное обновление системы и закрытие уязвимостей.

14. Управление конфигурациями и настройками среды

- **О чем занятие:** Автоматизация настройки серверов и принцип «Инфраструктура как код» (IaC).
- **Ход занятия:**
    - Проблемы ручного администрирования и методы их решения.
    - Обзор инструментов автоматизации (Ansible, Terraform).
    - Синхронизация настроек между средами разработки и промышленной эксплуатации.
    - Создание сценариев для автоматического развертывания систем.

15. Резервное копирование и восстановление серверных сред

- **О чем занятие:** Обеспечение сохранности данных и непрерывности работы при критических сбоях.
- **Ход занятия:**

- Разработка стратегии бэкапов: что, как часто и куда копировать.
- Автоматизация создания резервных копий баз данных и файлов.
- Проверка целостности архивов и регламент восстановления.
- План действий в случае полной потери данных (Disaster Recovery).

Краткое изложение планов хода практических занятий и примерные задания:

1. Установка и настройка среды разработки для веб-приложений

Ход занятия: Загрузка и инсталляция базового языка/платформы (Node.js, PHP или Python). Проверка работоспособности интерпретатора через командную строку.

Задание: Установить выбранную платформу и вывести в консоль версию ПО, используя системную команду.

2. Настройка редактора кода и инструментов разработчика

Ход занятия: Установка VS Code или аналогичного редактора. Настройка базовых плагинов для подсветки синтаксиса и проверки ошибок (Linter).

Задание: Сконфигурировать файл настроек редактора так, чтобы код автоматически форматировался при сохранении файла.

3. Установка серверной платформы выполнения приложений

Ход занятия: Развертывание среды для запуска кода (например, Docker Desktop или локальный стек серверов). Проверка статуса активных служб.

Задание: Запустить платформу и подтвердить доступность сервиса через системный монитор или браузер по адресу `localhost`.

4. Настройка переменных окружения среды разработки

Ход занятия: Работа с системными переменными (PATH). Создание и использование файлов конфигурации окружения (`.env`).

Задание: Прописать путь к установленному инструменту в переменные системы и создать переменную `APP_MODE` со значением `development`.

5. Установка и настройка локального веб-сервера

Ход занятия: Установка Nginx или Apache. Запуск службы и проверка стандартных портов (80, 8080).

Задание: Запустить веб-сервер и добиться отображения стандартной приветственной страницы в браузере.

6. Конфигурирование Apache / Nginx

Ход занятия: Редактирование основного файла конфигурации. Изменение порта прослушивания и указание пути к корневой папке сайта.

Задание: Изменить стандартный порт сервера на `8000` и перезапустить службу без ошибок.

7. Развертывание тестового веб-проекта

Ход занятия: Копирование файлов сайта в рабочую директорию сервера. Настройка прав доступа для чтения файлов веб-сервером.

Задание: Разместить простой HTML-проект на локальном сервере так, чтобы он открывался по локальному IP-адресу.

8. Настройка структуры каталогов веб-проекта

Ход занятия: Проектирование архитектуры папок проекта (разделение на исходники `src`, публичные файлы `public` и ассеты).

Задание: Создать иерархию папок для нового проекта в соответствии со стандартом выбранного фреймворка или языка.

9. Установка и настройка системы управления базами данных

Ход занятия: Инсталляция СУБД (MySQL/PostgreSQL). Первичная авторизация и установка пароля администратора.

Задание: Создать пустую базу данных под названием `test_db` через консоль или графический интерфейс.

10. Настройка подключения веб-приложения к базе данных

Ход занятия: Конфигурирование параметров подключения (хост, порт, логин, пароль) в коде приложения. Проверка стабильности соединения.

Задание: Реализовать простейший скрипт, который выводит сообщение «Соединение установлено» при успешном подключении к базе.

11. Установка системы контроля версий Git

Ход занятия: Инсталляция Git. Идентификация пользователя (настройка `user.name` и `user.email`).

Задание: Задать глобальные настройки пользователя и проверить их командой `git config --list`.

12. Создание локального репозитория проекта

Ход занятия: Инициализация Git в папке проекта. Формирование файла `.gitignore` для исключения системного мусора.

Задание: Сделать первый коммит (initial commit), содержащий только базовую структуру проекта.

13. Работа с удалённым репозиторием (GitHub / GitLab)

Ход занятия: Создание пустого репозитория на платформе. Связывание локальной папки с удаленным адресом (remote).

Задание: Отправить (Push) локальную ветку в облачный репозиторий.

14. Настройка совместной работы над проектом

Ход занятия: Создание новых веток для задач (feature-branches). Моделирование ситуации конфликта при слиянии веток.

Задание: Создать ветку, внести изменения в тот же файл, что и в основной ветке, и вручную разрешить возникший конфликт при слиянии.

15. Настройка прав доступа к серверной среде

Ход занятия: Работа с владельцами файлов и группами. Ограничение прав на выполнение скриптов в публичных папках.

Задание: Установить права `644` на конфигурационный файл приложения, чтобы его нельзя было случайно исполнить.

16. Конфигурирование параметров веб-сервера

Ход занятия: Настройка лимитов памяти, времени выполнения запроса и максимального размера передаваемых данных.

Задание: Ограничить в конфигурации размер загружаемого файла до `10 мегабайт`.

17. Настройка журналирования и мониторинга

Ход занятия: Определение путей для хранения логов доступа и ошибок. Анализ журналов в реальном времени.

Задание: Найти в файле логов запись о последнем успешном запросе к серверу.

18. Настройка параметров безопасности серверной среды

Ход занятия: Отключение ненужных модулей сервера. Скрытие информации о версии ПО в заголовках ответов.

Задание: Настроить сервер так, чтобы он не передавал заголовок `X-Powered-By` или `Server` с точной версией.

19. Создание резервной копии проекта

Ход занятия: Создание полного архива файлов и экспорт дампа базы данных (SQL-файл).

Задание: Сформировать архив бэкапа, включающий в себя как файлы кода, так и актуальное состояние базы данных.

20. Восстановление среды разработки из резервной копии

Ход занятия: Удаление текущей версии проекта. Процесс развертывания из архива и импорт данных в СУБД.

Задание: Полностью восстановить работоспособность проекта после удаления его исходных файлов, используя только созданный ранее бэкап.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
МДКн.04.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ РЕСУРСОВ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.09 «Веб-разработка»** и предназначен для проведения текущего контроля и промежуточной аттестации по дисциплине **МДК.04.02 «Разработка кода информационных ресурсов»**.

ФОС представляет собой комплекс теоретических вопросов и практических заданий, направленных на оценку навыков написания серверного кода, работы с базами данных и создания прикладной логики веб-приложений (Back-end разработка).

2. Цель и задачи курса

Цель: Подготовка квалифицированного специалиста (Back-end разработчика), способного проектировать программную логику, реализовывать алгоритмы обработки данных и обеспечивать взаимодействие веб-приложения с серверными ресурсами.

Задачи:

- **Освоение серверных технологий:** Изучение синтаксиса и возможностей современных языков программирования (Node.js, Python или PHP).
- **Проектирование API:** Овладение навыками создания RESTful-сервисов для взаимодействия с клиентской частью приложения.
- **Работа с данными:** Изучение методов интеграции приложений с базами данных (SQL/NoSQL) и принципов построения запросов.
- **Безопасность кода:** Внедрение механизмов аутентификации, авторизации и защиты от распространенных веб-уязвимостей.
- **Тестирование и отладка:** Овладение инструментами поиска ошибок и написания автоматизированных тестов для серверных модулей.

3. Структура оценочных материалов

Содержание ФОС коррелирует с этапами разработки серверной части веб-ресурса:

- **Модуль «Основы серверной разработки»:** Оценка умения работать с HTTP-протоколом, маршрутизацией и обработкой параметров запросов.
- **Модуль «Бизнес-логика и авторизация»:** Контроль навыков реализации пользовательских сценариев, управления сессиями и разграничения прав доступа.
- **Модуль «Интеграция с данными»:** Практические задания на выполнение CRUD-операций (создание, чтение, обновление, удаление) в базе данных.
- **Модуль «Оптимизация и тестирование»:** Проверка компетенций в области анализа производительности кода, обработки асинхронных операций и покрытия кода тестами.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** архитектуру серверных приложений; принципы работы REST API; методы хранения и защиты данных; форматы обмена данными (JSON, XML).
- **Уметь:** реализовывать серверную логику на выбранном языке программирования; проектировать структуру базы данных; обрабатывать асинхронные запросы; интегрировать внешние библиотеки и сервисы.
- **Владеть:** инструментами отладки серверного кода; навыками написания безопасных SQL-запросов (или использования ORM); методиками разработки модульных и масштабируемых приложений.

5. Методические рекомендации по оцениванию

Оценивание строится на принципах **функциональной проверки кода**:

- **Теоретическая часть:** Оценивается знание алгоритмов, понимание архитектурных паттернов и чистота владения терминологией бэкенд-разработки.
- **Практическая часть:** Оценивается по факту корректного выполнения бизнес-логики. Критическими ошибками считаются: неработающие маршруты (эндпоинты), уязвимости (инъекции), некорректная обработка данных в БД или отсутствие обработки исключительных ситуаций (ошибок).

Тема 2.1. Виды и назначения серверных языков программирования, среды разработки.

Краткое изложение планов хода лекционных занятий:

1. Понятие серверного программирования и его роль в разработке информационных ресурсов

О чем занятие: Суть бэкенд-разработки. Место серверного кода в обеспечении логики приложения, работы с данными и безопасности ресурсов.

Ход занятия:

- Понятие Front-end и Back-end: зоны ответственности.
- Жизненный цикл веб-запроса и роль серверного сценария в его обработке.
- Задачи, решаемые на стороне сервера: аутентификация, валидация данных, работа с файловой системой.
- Влияние качества серверного кода на производительность и масштабируемость системы.

2. Архитектура клиент-серверных приложений

О чем занятие: Механизмы взаимодействия между пользователем и сервером. Разбор различных подходов к организации обмена данными.

Ход занятия:

- Протокол HTTP/HTTPS как основа взаимодействия в вебе.
- Традиционная модель (Server-Side Rendering) против современной (SPA + API).
- Понятие REST API и JSON как универсального языка общения компонентов.
- Состояние приложения: Stateless-архитектура и использование сессий/токенов.

3. Виды серверных языков программирования

О чем занятие: Классификация инструментов разработки. Различия между интерпретируемыми, компилируемыми и скриптовыми языками.

Ход занятия:

- История развития серверных языков (от CGI-скриптов до современных сред).
- Компилируемые языки против интерпретируемых: баланс скорости и гибкости.
- Типизация данных (статическая и динамическая) и её влияние на надежность серверного кода.
- Понятие многопоточности и асинхронности в контексте обработки запросов.

4. Назначение и особенности серверных языков программирования

О чем занятие: Специфические функции языков, делающие их пригодными для работы на сервере: сетевые библиотеки, драйверы БД и безопасность.

Ход занятия:

- Работа с потоками ввода-вывода (I/O) и файловыми дескрипторами.
- Встроенные механизмы обеспечения безопасности и защиты от типичных атак (SQL Injection, XSS).
- Экосистема фреймворков и их роль в ускорении разработки.
- Поддержка протоколов (WebSockets, gRPC) на уровне языка.

5. Популярные серверные языки программирования (PHP, Python, JavaScript, Java)

О чем занятие: Обзор лидеров рынка веб-разработки. Области применения каждого языка в зависимости от типа проекта.

Ход занятия:

- **PHP:** Классика веба, простота развертывания и популярность в CMS.
- **Python (Django/Flask):** Чистота кода, мощные библиотеки и работа с данными.
- **JavaScript (Node.js):** Единый язык для фронтенда и бэкенда, высокая скорость событийной модели.
- **Java (Spring):** Стандарт для высоконагруженных корпоративных систем (Enterprise).

6. Сравнительный анализ серверных языков программирования

О чем занятие: Объективное сопоставление инструментов по ключевым метрикам бизнеса и разработки.

Ход занятия:

- Сравнение производительности (бенчмарки обработки запросов).
- Порог вхождения и скорость разработки (Time-to-market).
- Доступность хостинга и сложность настройки серверной среды.
- Рынок труда: востребованность специалистов и стоимость поддержки кода.

7. Среды разработки серверных приложений

О чем занятие: Инструментарий бэкенд-разработчика. Настройка рабочего пространства для эффективного написания и запуска серверного кода.

Ход занятия:

- Специализированные IDE против универсальных редакторов (например, PyCharm vs VS Code).
- Инструменты удаленной разработки (Remote Development через SSH).
- Использование локальных серверов и контейнеров для эмуляции среды продакшена.
- Интеграция сред разработки с системами контроля версий.

8. Инструменты разработки и тестирования серверного кода

О чем занятие: Контроль качества бэкенда. Использование отладчиков, профайлеров и средств тестирования API.

Ход занятия:

- Методы отладки кода (Debugging): точки остановки, инспекция переменных.
- Тестирование API с помощью Postman или Insomnia.
- Понятие Unit-тестирования и интеграционных тестов для серверной логики.
- Профайлинг: поиск «узких мест» в производительности алгоритмов.

9. Управление зависимостями и пакетами в серверных проектах

О чем занятие: Механизмы подключения внешних библиотек. Работа с пакетными менеджерами и разрешение конфликтов версий.

Ход занятия:

- Роль пакетных менеджеров (NPM, Composer, Pip, Maven) в современной разработке.
- Семантическое версионирование и файлы блокировки (lock-файлы).
- Управление виртуальными окружениями для изоляции проектов друг от друга.
- Безопасность цепочки поставок: проверка сторонних пакетов на уязвимости.

Краткое изложение планов хода практических занятий и примерные задания:

1. Установка среды разработки для серверного программирования

Ход занятия: Выбор и установка серверного стека (например, Node.js или Python). Настройка путей в системных переменных и проверка доступности компилятора/интерпретатора.

Задание: Установить среду выполнения и выполнить команду проверки версии, подтвердив успешную установку в отчете.

2. Настройка проекта серверного приложения

Ход занятия: Инициализация структуры проекта (npm init или venv). Установка базового веб-фреймворка (Express, Flask или Fastify). Создание файла конфигурации.

Задание: Создать рабочую директорию проекта и инициализировать в ней менеджер пакетов, установив одну внешнюю зависимость (библиотеку).

3. Создание первого серверного приложения

Ход занятия: Написание минимального исполняемого кода для запуска сервера. Назначение порта (например, 3000) и запуск «прослушивания» входящих подключений.

Задание: Написать скрипт, который при запуске выводит в консоль сообщение «Сервер запущен на порту X» и возвращает строку «Hello World» в браузере.

4. Работа с HTTP-запросами

Ход занятия: Изучение структуры HTTP-запроса (Method, Headers, Body). Использование инструментов для отправки запросов (Postman/Insomnia).

Задание: С помощью Postman отправить запросы разных типов (GET, POST) на локальный сервер и зафиксировать их получение в логах сервера.

5. Обработка параметров запроса

Ход занятия: Работа с динамическими данными. Извлечение данных из строки запроса (Query parameters) и путей (URL parameters).

Задание: Реализовать обработчик, который принимает имя пользователя из URL (например, `/user/Ivan`) и выводит персональное приветствие на страницу.

6. Формирование HTTP-ответов

Ход занятия: Управление статус-кодами (200, 404, 500). Настройка заголовков ответа (Content-Type). Отправка данных в форматах HTML и JSON.

Задание: Настроить сервер так, чтобы при обращении к несуществующей странице он возвращал корректный статус-код 404 и оформленную страницу ошибки.

7. Работа с маршрутами приложения

Ход занятия: Создание системы маршрутизации (Routing). Разделение логики приложения по разным URL-адресам.

Задание: Создать три разных маршрута (`/`, `/about`, `/contacts`), каждый из которых должен возвращать уникальный контент.

8. Разработка серверной логики приложения

Ход занятия: Написание функций-контроллеров. Обработка условий и выполнение математических или логических операций на стороне сервера перед отправкой ответа.

Задание: Реализовать простейший калькулятор, который получает два числа через параметры запроса и возвращает результат их сложения.

9. Работа с формами и отправкой данных

Ход занятия: Создание HTML-формы. Настройка атрибутов `action` и `method`. Прием данных на сервере через тело запроса (Request Body).

Задание: Разработать форму обратной связи и настроить сервер на прием данных из этой формы с последующим выводом их в консоль.

10. Обработка пользовательского ввода

Ход занятия: Фильтрация и валидация данных (проверка на пустые поля, корректность e-mail). Обеспечение базовой безопасности ввода.

Задание: Написать программный фильтр, который проверяет длину пароля в форме и возвращает ошибку, если он короче 8 символов.

11. Реализация механизма сессий

Ход занятия: Подключение модуля сессий. Хранение временных данных о пользователе на стороне сервера.

Задание: Реализовать счетчик посещений страницы для конкретного пользователя, данные о котором сохраняются в сессии.

12. Работа с cookie

Ход занятия: Установка, чтение и удаление куки-файлов в браузере через серверные заголовки. Настройка срока жизни cookie (Expires/Max-Age).

Задание: Создать функционал «темной темы», где выбор пользователя сохраняется в cookie и применяется при перезагрузке страницы.

13. Подключение базы данных к серверному приложению

Ход занятия: Установка драйвера БД. Настройка строки подключения (Connection String). Проверка успешности соединения.

Задание: Подключить приложение к локальной базе данных и вывести в консоль статус «Connected», если база доступна.

14. Выполнение запросов к базе данных

Ход занятия: Изучение синтаксиса SQL-запросов или методов ORM. Выполнение простых команд через серверный код.

Задание: Написать функцию, которая отправляет запрос на получение системного времени или версии из базы данных.

15. Добавление данных в базу

Ход занятия: Реализация операции Create (INSERT). Передача данных из формы в таблицу базы данных.

Задание: Создать страницу регистрации, данные с которой (логин и почта) будут успешно записываться в таблицу базы данных.

16. Извлечение данных из базы данных

Ход занятия: Реализация операции Read (SELECT). Получение списка записей и их отображение в виде HTML-списка или таблицы.

Задание: Вывести на экран список всех зарегистрированных пользователей, хранящихся в базе данных.

17. Обновление данных в базе

Ход занятия: Реализация операции Update (UPDATE). Поиск записи по идентификатору (ID) и изменение отдельных полей.

Задание: Создать страницу профиля, где пользователь может изменить свою электронную почту в базе данных.

18. Удаление данных из базы

Ход занятия: Реализация операции Delete (DELETE). Удаление записей по ID. Меры предосторожности при удалении данных.

Задание: Добавить кнопку «Удалить аккаунт», которая при нажатии безвозвратно стирает запись пользователя из таблицы базы данных.

Тема 2.2. Программирование на стороне сервера.

Краткое изложение планов хода лекционных занятий:

1. Основы разработки серверного кода

О чем занятие: Базовые принципы создания бэкенд-логики. Понятие серверного скрипта, его жизненный цикл и роль в обработке данных.

Ход занятия:

- Различие между статическим контентом и динамической генерацией страниц на сервере.
- Принципы работы интерпретаторов и сред выполнения (Node.js, Python, PHP).
- Синтаксические особенности языков, ориентированных на серверную разработку.
- Безопасность выполнения серверного кода: изоляция процессов.

2. Структура серверного приложения

О чем занятие: Архитектурная организация кода. Как разделить логику, данные и интерфейс для удобного администрирования.

Ход занятия:

- Точка входа в приложение (Entry point) и инициализация сервера.
- Паттерн MVC (Model-View-Controller) применительно к серверной части.
- Слои приложения: роутинг, контроллеры, сервисы и доступ к данным.
- Конфигурационные файлы проекта и управление зависимостями.

3. Обработка HTTP-запросов на стороне сервера

О чем занятие: Глубокое изучение того, как сервер «понимает» клиента. Разбор анатомии входящего трафика.

Ход занятия:

- Структура HTTP-пакета: стартовая строка, заголовки и тело запроса.
- Методы передачи данных (GET, POST, PUT, DELETE) и их семантика.
- Парсинг (анализ) входящих данных различных типов (JSON, XML, Form-data).
- Промежуточное ПО (Middleware) для предварительной обработки запросов.

4. Формирование HTTP-ответов

О чем занятие: Правила «общения» сервера с браузером. Как возвращать данные так, чтобы клиент их корректно интерпретировал.

Ход занятия:

- Коды состояния HTTP (2xx, 3xx, 4xx, 5xx) и их правильное использование.
- Типы контента (MIME-types) и управление заголовком Content-Type.
- Перенаправления (Redirects) и кэширование на стороне клиента.
- Передача бинарных данных и потоков (Streaming).

5. Работа с параметрами запроса и формами

О чем занятие: Механизмы получения данных от пользователя. Безопасный прием информации через URL и HTML-формы.

Ход занятия:

- Параметры строки запроса (Query Strings) и параметры пути (Path params).
- Кодирование данных форм (application/x-www-form-urlencoded и multipart/form-data).
- Валидация и санитизация (очистка) пользовательского ввода для предотвращения инъекций.
- Загрузка файлов на сервер: приём, проверка и сохранение.

6. Работа с сессиями и cookie

О чем занятие: Решение проблемы «отсутствия состояния» (stateless) протокола HTTP. Идентификация пользователя между запросами.

Ход занятия:

- Назначение Cookie: установка заголовка Set-Cookie и параметры безопасности (HttpOnly, Secure).
- Механизм сессий на сервере: хранение идентификаторов и данных состояния.
- Сравнение хранения данных в Cookie vs LocalStorage vs Session.
- Проблемы безопасности: перехват сессий и CSRF-атаки.

7. Взаимодействие серверного приложения с базами данных

О чем занятие: Теоретические основы связки «код — данные». Как приложение хранит и извлекает информацию.

Ход занятия:

- Типы СУБД для веб-проектов: реляционные (SQL) и нереляционные (NoSQL).
- Архитектурные подходы: работа через «чистый» SQL vs использование ORM (Object-Relational Mapping).
- Понятие пула соединений (Connection Pool) для оптимизации нагрузки на сервер.
- Транзакции и обеспечение целостности данных при выполнении запросов.

8. Подключение базы данных к серверному приложению

О чем занятие: Практические аспекты интеграции. Конфигурирование доступа и проверка стабильности связи.

Ход занятия:

- Строки подключения (Connection Strings) и хранение учетных данных в переменных окружения.
- Использование драйверов и адаптеров для конкретных языков программирования.
- Обработка ошибок подключения и механизмы повторных попыток (Retry).
- Логирование запросов к базе данных для целей отладки и мониторинга производительности.

Краткое изложение планов хода практических занятий и примерные задания:

1. Разработка API для веб-приложения

Ход занятия: Проектирование интерфейса взаимодействия (Endpoints). Определение структуры URL и методов (GET, POST) для обмена данными с фронтендом.

Задание: Создать набор эндпоинтов для получения и отправки данных о товарах или пользователях.

2. Работа с JSON-данными

Ход занятия: Изучение формата JSON. Сериализация (преобразование объекта в строку) и десериализация (чтение строки в объект) данных на сервере.

Задание: Написать обработчик, который принимает JSON-объект от клиента и возвращает преобразованный ответ в том же формате.

3. Реализация обработки ошибок

Ход занятия: Использование блоков `try...catch`. Создание глобального обработчика исключений для предотвращения падения сервера при сбоях.

Задание: Настроить сервер так, чтобы при любой внутренней ошибке клиент получал аккуратный ответ со статусом 500 и пояснением, а не пустую страницу.

4. Логирование работы приложения

Ход занятия: Подключение библиотек для ведения журналов (например, Morgan или Winston). Запись событий (запросы, ошибки, запуск сервера) в консоль и файлы.

Задание: Настроить запись всех входящих запросов в файл `access.log` с указанием времени и метода запроса.

5. Тестирование серверного приложения

Ход занятия: Знакомство с автоматизированным тестированием. Написание простых тестов для проверки доступности основных маршрутов (маршрутизации).

Задание: Написать тест, который проверяет, что главная страница сервера возвращает статус-код 200.

6. Оптимизация серверного кода

Ход занятия: Анализ производительности алгоритмов. Устранение избыточных запросов к базе данных и оптимизация циклов.

Задание: Провести рефакторинг функции обработки массива данных, сократив количество используемой памяти или время выполнения.

7. Организация структуры серверного проекта

Ход занятия: Применение паттерна «Разделение ответственности». Вынос роутов, контроллеров и моделей в отдельные директории.

Задание: Перенести логику из одного основного файла в модульную структуру (папки `routes/`, `controllers/`).

8. Разработка серверного модуля обработки данных

Ход занятия: Создание изолированной логики (сервиса) для сложных вычислений или преобразования данных (например, парсинг текста или обработка изображений).

Задание: Написать модуль, который принимает массив чисел и возвращает объект с их статистическими показателями (среднее, максимум и т.д.).

9. Реализация авторизации пользователей

Ход занятия: Проверка прав доступа. Разграничение контента для обычных пользователей и администраторов на основе их ролей.

Задание: Создать маршрут `/admin`, доступ к которому разрешен только пользователям с соответствующим флагом в базе данных.

10. Реализация аутентификации пользователей

Ход занятия: Процесс входа в систему. Сравнение хешированных паролей. Использование токенов (JWT) или сессий для подтверждения личности.

Задание: Реализовать функцию Login, которая проверяет логин/пароль и выдает временный доступ к защищенным ресурсам.

11. Разработка REST API

Ход занятия: Приведение API к стандартам REST. Использование правильных HTTP-методов (PUT для обновления, DELETE для удаления) и статус-кодов.

Задание: Реализовать полный цикл управления ресурсом «Заметки» (создание, чтение, обновление, удаление) по канонам REST.

12. Обработка асинхронных запросов

Ход занятия: Работа с асинхронностью (Async/Await, Promises). Обеспечение стабильной работы сервера при ожидании ответов от БД или внешних API.

Задание: Написать асинхронную функцию запроса к базе данных, которая корректно обрабатывает ожидание ответа без блокировки сервера.

13. Использование сторонних библиотек

Ход занятия: Поиск и установка пакетов через менеджер зависимостей (npm/pip) для решения специфических задач (например, работа с датами через Moment.js).

Задание: Установить библиотеку для работы с датами и реализовать вывод текущего времени в специфическом формате на сервере.

14. Интеграция серверного приложения с внешними сервисами

Ход занятия: Взаимодействие между серверами. Отправка запросов к сторонним API (например, получение курса валют или погоды).

Задание: Реализовать эндпоинт, который запрашивает данные с публичного API и пересылает их клиенту.

15. Подготовка серверного приложения к развертыванию

Ход занятия: Проверка безопасности и переменных окружения. Сборка проекта, очистка от отладочных логов и подготовка файла запуска (например, `ecosystem.config.js`).

Задание: Создать `.env.example` файл и проверить готовность проекта к запуску в режиме `production`.

16. Тестирование и отладка серверного приложения

Ход занятия: Комплексная проверка всех модулей. Использование встроенных инструментов отладки (Debugger) для поиска скрытых ошибок.

Задание: Найти и исправить три намеренно допущенные ошибки в предоставленном программном коде.

17. Итоговая практическая работа по разработке серверного приложения

Ход занятия: Самостоятельное создание полноценного веб-сервиса (например, мини-блога или планировщика задач) с нуля до рабочего протокола.

Задание: Разработать и запустить серверное приложение, включающее роутинг, работу с БД и аутентификацию.

18. Анализ и оптимизация разработанного серверного приложения

Ход занятия: Оценка написанного кода на соответствие стандартам. Измерение времени отклика сервера под нагрузкой.

Задание: Выявить наиболее медленный эндпоинт в итоговой работе и предложить (или реализовать) способ его ускорения.

19. Подготовка отчета по практическим работам

Ход занятия: Систематизация полученных результатов. Описание структуры проекта, использованных технологий и скриншоты работающих эндпоинтов.

Задание: Оформить отчетную документацию, включающую пояснительную записку и листинги ключевых модулей приложения.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ СВЕРДЛОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СВЕРДЛОВСКОЙ ОБЛАСТИ «ЕКАТЕРИНБУРГСКИЙ МОНТАЖНЫЙ КОЛЛЕДЖ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МДКн.04.03 РАЗРАБОТКА ИНФОРМАЦИОННЫХ РЕСУРСОВ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНЫХ ПЛАТФОРМ

для специальности
09.02.09 «Веб-разработка»

Екатеринбург
2026

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Назначение ФОС

Данный Фонд оценочных средств разработан в соответствии с требованиями ФГОС СПО по специальности **09.02.09 «Веб-разработка»** и предназначен для проведения текущего контроля и промежуточной аттестации по дисциплине **МДК.04.03 «Разработка информационных ресурсов с использованием программных платформ»**.

ФОС представляет собой комплекс теоретических вопросов и практических кейсов, направленных на оценку навыков использования современных высокоуровневых программных платформ (фреймворков) для создания масштабируемых, защищенных и производительных информационных ресурсов.

2. Цель и задачи курса

Цель: Подготовка специалиста, способного эффективно использовать экосистемы современных программных платформ для быстрой разработки и развертывания сложных веб-проектов в соответствии с промышленными стандартами.

Задачи:

- **Освоение программных платформ:** Изучение архитектуры, структуры и возможностей современных веб-фреймворков (Laravel, Django, Express и др.).
- **Компонентная разработка:** Овладение навыками модульного программирования и интеграции сторонних библиотек через менеджеры зависимостей.
- **Реализация комплексных систем:** Разработка полноценных информационных ресурсов, включающих API, системы авторизации и механизмы взаимодействия с внешними сервисами.
- **Обеспечение качества и безопасности:** Применение встроенных инструментов платформ для защиты данных, тестирования кода и логирования событий.
- **Развертывание и эксплуатация:** Подготовка веб-приложения к работе на реальном сервере и настройка производственной среды.

3. Структура оценочных материалов

Содержание ФОС коррелирует с этапами создания ресурса на базе программных платформ:

- **Модуль «Архитектура и среда платформы»:** Оценка умения настраивать окружение, инициализировать проекты и конфигурировать структуру фреймворка.
- **Модуль «Функциональная разработка»:** Контроль навыков реализации бизнес-логики через контроллеры, модели и шаблонизаторы платформы.
- **Модуль «Интеграция и API»:** Практические задания по работе с JSON, сторонними библиотеками и разработке программных интерфейсов.
- **Модуль «Безопасность, оптимизация и деплой»:** Проверка компетенций в области защиты от веб-уязвимостей, повышения производительности и развертывания проекта на сервере.

4. Планируемые результаты освоения

В результате освоения курса обучающийся должен демонстрировать следующие результаты:

- **Знать:** архитектурные паттерны фреймворков (MVC и др.); принципы управления зависимостями; методы обеспечения безопасности на уровне платформы; способы оптимизации производительности.
- **Уметь:** разворачивать проекты на базе программных платформ; интегрировать API внешних сервисов; автоматизировать рутинные задачи с помощью CLI-инструментов; настраивать серверную среду.
- **Владеть:** навыками разработки в рамках экосистемы выбранного фреймворка; инструментарием автоматизированного тестирования и отладки; методиками подготовки отчетной и технической документации.

5. Методические рекомендации по оцениванию

Оценивание строится на принципах **архитектурной точности и работоспособности**:

- **Теоретическая часть:** Оценивается понимание принципов работы платформы, знание паттернов проектирования и умение обосновать выбор конкретных библиотек или инструментов.
- **Практическая часть:** Оценивается по факту реализации полноценного функционала. Критическими ошибками считаются: нарушение архитектурной структуры фреймворка, неработоспособность ключевых модулей, отсутствие защиты данных или невозможность развертывания проекта из-за ошибок конфигурации.

Краткое изложение планов хода лекционных занятий:

1. Понятие программных платформ разработки информационных ресурсов

- **О чем занятие:** Платформа как комплексная среда для создания ПО. Отличие платформенной разработки от использования отдельных инструментов.
- **Ход занятия:**
  - Определение программной платформы и её составляющих (языки, API, среды исполнения).
  - Взаимодействие платформы с операционной системой и аппаратным обеспечением.
  - Жизненный цикл платформ и критерии их выбора для бизнеса.
  - Понятие экосистемы разработчика.

2. Назначение и виды фреймворков и библиотек

- **О чем занятие:** Инструменты повторного использования кода. Фундаментальные различия в управлении потоком выполнения программы.
- **Ход занятия:**
  - Библиотека как вспомогательный набор функций (вызывается разработчиком).
  - Фреймворк как управляющая структура (вызывает код разработчика — Inversion of Control).
  - Классификация: микрофреймворки, полнофункциональные (Full-stack) каркасы.
  - Критерии надежности стороннего кода: лицензия, поддержка, сообщество.

3. Архитектура современных веб-фреймворков

- **О чем занятие:** Внутреннее устройство профессиональных платформ. Слои абстракции и механизмы обработки данных.
- **Ход занятия:**
  - Обработка запроса: от входного скрипта до формирования ответа.
  - Слой промежуточного ПО (Middleware) и его задачи.
  - Сервис-контейнеры и автоматическое внедрение зависимостей (Dependency Injection).
  - Концепция «Соглашение превыше конфигурации» (Convention over Configuration).

4. Принципы работы фреймворков MVC

- **О чем занятие:** Классический паттерн проектирования веб-интерфейсов. Разделение данных, логики и визуального представления.
- **Ход занятия:**
  - **Model:** управление состоянием и бизнес-логикой.
  - **View:** логика отображения и шаблонизация.
  - **Controller:** обработка ввода и координация между Моделью и Видом.
  - Преимущества MVC: параллельная разработка и упрощенное тестирование.

5. Использование библиотек в разработке веб-приложений

- **О чем занятие:** Практические аспекты расширения базовых возможностей платформы за счет узкоспециализированных решений.
- **Ход занятия:**
    - Пакетные менеджеры (NPM, Composer, Pip) как основа современной разработки.
    - Семантическое версионирование и контроль обновлений.
    - Использование библиотек для решения типовых задач: работа с изображениями, PDF, сложной математикой.
    - Анализ безопасности подключаемых пакетов.

6. Преимущества использования фреймворков в разработке

- **О чем занятие:** Почему промышленная разработка невозможна без использования готовых каркасов. Экономический и технический аспекты.
- **Ход занятия:**
    - Стандартизация кода и упрощение командной работы.
    - Встроенные решения для типичных задач (авторизация, кэширование, почта).
    - Безопасность «из коробки» (защита от стандартных веб-атак).
    - Масштабируемость и легкость миграции на другие серверы.

7. Обзор популярных серверных фреймворков

- **О чем занятие:** Анализ рынка бэкенд-решений. Сравнение подходов разных языковых экосистем.
- **Ход занятия:**
    - **Laravel (PHP):** экосистема, синтаксис, ORM Eloquent.
    - **Django (Python):** «батарейки в комплекте», административная панель.
    - **Express.js / NestJS (Node.js):** асинхронность и высокая производительность.
    - **Spring Boot (Java) и ASP.NET Core:** решения для корпоративного сектора.

8. Обзор популярных клиентских фреймворков

- **О чем занятие:** Современный фронтенд. Перенос логики интерфейса на сторону клиента.
- **Ход занятия:**
    - **React:** компонентный подход и работа с состоянием.
    - **Vue.js:** легкость интеграции и прогрессивность.
    - **Angular:** мощное решение для крупных корпоративных систем.
    - Понятие SPA (Single Page Application) и взаимодействие с сервером через API.

9. Структура проекта, разработанного с использованием фреймворка

- **О чем занятие:** Типовая организация файлов и папок. Где искать ключевые элементы приложения.
- **Ход занятия:**
    - Расположение файлов конфигурации и переменных окружения (.env).
    - Директории ресурсов (Assets), маршрутов (Routes) и публичных файлов (Public).
    - Где хранится логика: контроллеры, модели и провайдеры.
    - Директории для хранения миграций и тестов.

10. Настройка окружения для работы с фреймворками

- **О чем занятие:** Подготовка локальной машины разработчика. Инструменты для запуска и отладки кода.
- **Ход занятия:**
  - Установка зависимостей (языки программирования, базы данных).
  - Настройка инструментов командной строки (CLI) фреймворков.
  - Использование локальных серверов разработки.
  - Конфигурирование доступа к БД через файлы окружения.

11. Организация маршрутизации в веб-приложениях

- **О чем занятие:** Связывание URL-адресов с программными методами. Управление навигационной логикой проекта.
- **Ход занятия:**
  - Определение статических и динамических маршрутов с параметрами.
  - Группировка маршрутов по функционалу (Admin, API, Web).
  - Использование посредников (Middleware) для защиты маршрутов.
  - Именованные маршруты и генерация URL.

12. Работа с шаблонизаторами

- **О чем занятие:** Эффективное построение интерфейсов. Механизмы наследования и вывода данных.
- **Ход занятия:**
  - Создание базовых макетов (Layouts) и вложенных секций.
  - Управляющие конструкции (циклы, условия) внутри HTML.
  - Безопасный вывод переменных и использование фильтров.
  - Компоненты шаблонов для повторного использования элементов дизайна.

13. Работа с контроллерами и моделями

- **О чем занятие:** Реализация взаимодействия между пользователем и данными. Жизненный цикл данных в приложении.
- **Ход занятия:**
  - Создание ресурсных контроллеров для обработки CRUD-запросов.
  - Определение моделей данных и настройка связей (Relationships).
  - Валидация входящих данных на уровне контроллера.
  - Работа с объектами запроса (Request) и ответа (Response).

14. Интеграция библиотек в веб-приложение

- **О чем занятие:** Методики расширения функционала платформы без нарушения её архитектуры.
- **Ход занятия:**
  - Регистрация сервис-провайдеров и использование фасадов.
  - Настройка конфигурационных файлов сторонних плагинов.
  - Использование менеджеров зависимостей для обновления библиотек.
  - Написание собственных адаптеров для внешних решений.

15. Обеспечение безопасности веб-приложений

- **О чем занятие:** Защита данных пользователей и инфраструктуры сервера. Предупреждение взломов.
- **Ход занятия:**
  - Автоматическая защита от CSRF (Cross-Site Request Forgery).
  - Экранирование вывода для предотвращения XSS-атак.
  - Хеширование паролей и безопасная аутентификация.
  - Настройка политики CORS и заголовков безопасности.

16. Тестирование веб-приложений, разработанных с использованием фреймворков

- **О чем занятие:** Автоматизированный контроль качества. Снижение рисков при внесении изменений.
- **Ход занятия:**
  - Различия между модульными (Unit) и функциональными (Feature) тестами.
  - Тестирование HTTP-запросов и ответов (статус-коды, JSON-структура).
  - Использование тестовых баз данных и фабрик данных.
  - Запуск тестов и интерпретация результатов в консоли.

17. Оптимизация производительности веб-приложений

- **О чем занятие:** Повышение скорости работы ресурса. Снижение нагрузки на сервер и базу данных.
- **Ход занятия:**
  - Оптимизация запросов к БД: жадная загрузка (Eager Loading) против ленивой.
  - Различные уровни кэширования: данные, маршруты, шаблоны.
  - Минификация статических файлов и использование CDN.
  - Очереди задач для тяжелых операций (отправка писем, ресайз фото).

18. Подготовка веб-приложений к развертыванию

- **О чем занятие:** Переход от разработки к запуску на реальном сервере (Production).
- **Ход занятия:**
  - Конфигурирование переменных окружения для боевого сервера.
  - Проверка безопасности и отключение режима отладки.
  - Компиляция ресурсов фронтенда (Build).
  - Создание скриптов для автоматического деплоя и выполнения миграций.

Краткое изложение планов хода практических занятий и примерные задания:

1. Установка и настройка программной платформы разработки

Ход занятия: Проверка системных требований (версия PHP/Python/Node.js). Установка глобальных менеджеров зависимостей и CLI-инструментов выбранной платформы.

Задание: Настроить локальное окружение и подтвердить установку платформы, выведя справочную информацию CLI в терминал.

2. Создание проекта с использованием фреймворка

Ход занятия: Развертывание базового шаблона приложения через командную строку. Настройка прав доступа к директориям и запуск встроенного сервера разработки.

Задание: Создать новый проект и добиться отображения стандартной страницы-приветствия фреймворка в браузере.

3. Настройка структуры проекта

Ход занятия: Работа с конфигурационными файлами и переменными окружения (`.env`). Определение структуры хранения статических файлов, стилей и скриптов.

Задание: Сконфигурировать базовые параметры приложения (название, режим отладки, URL) и создать иерархию папок для ресурсов.

4. Настройка маршрутизации в веб-приложении

Ход занятия: Создание карты URL-адресов. Использование именованных маршрутов и групп маршрутов для разделения логических зон сайта.

Задание: Реализовать систему маршрутов для многостраничного ресурса, включая обработку динамических параметров в URL.

5. Реализация контроллеров

- **Ход занятия:** Создание классов-контроллеров для обработки бизнес-логики. Организация методов для обработки различных типов запросов.
- **Задание:** Создать контроллер, который принимает запрос и возвращает данные в представление или текстовый ответ.

6. Реализация моделей данных

Ход занятия: Проектирование схемы базы данных. Создание моделей и файлов миграций. Настройка связей между таблицами.

Задание: Создать модель «Товар» (или аналогичную) и выполнить миграцию, сформировав таблицу в базе данных.

7. Работа с шаблонизаторами

Ход занятия: Использование механизмов наследования и секций шаблонизатора. Создание мастер-макета сайта.

Задание: Реализовать основной макет приложения с динамической подстановкой контента в зависимости от выбранного маршрута.

8. Реализация пользовательского интерфейса

Ход занятия: Передача данных из контроллера в представление. Использование циклов и условий шаблонизатора для вывода динамических списков.

Задание: Сверстать страницу каталога, которая выводит список объектов из базы данных в виде структурированной HTML-разметки.

9. Подключение библиотек к проекту

Ход занятия: Изучение реестров пакетов. Инициализация и регистрация стороннего кода в структуре программной платформы.

Задание: Подключить к проекту библиотеку для работы с внешними форматами (например, генерация PDF-отчетов) и настроить её вызов.

10. Использование сторонних библиотек

Ход занятия: Поиск и установка пакетов через менеджер зависимостей (npm/pip) для решения специфических задач (например, работа с датами через Moment.js/Carbon).

Задание: Установить библиотеку для работы с датами и реализовать вывод текущего времени в специфическом формате на сервере.

11. Разработка функциональных модулей приложения

Ход занятия: Проектирование изолированной бизнес-логики. Создание сервисных классов для выполнения сложных вычислений.

Задание: Разработать модуль «Расчет стоимости», который может быть использован в разных контроллерах приложения.

12. Реализация обработки пользовательских запросов

Ход занятия: Работа с объектом запроса (Request). Получение данных из заголовков, параметров строки запроса и тела POST-запроса.

Задание: Написать обработчик, который логирует IP-адрес и параметры запроса пользователя при посещении определенных страниц.

13. Работа с формами и вводом данных

Ход занятия: Создание HTML-форм средствами платформы. Настройка передачи данных на сервер и использование CSRF-защиты.

Задание: Реализовать форму обратной связи, данные из которой отправляются на сервер для дальнейшей обработки.

14. Реализация валидации данных

Ход занятия: Описание правил проверки (обязательность, формат, уникальность). Обработка ошибок ввода и возврат сообщений пользователю.

Задание: Настроить проверку полей формы регистрации (например, проверка формата e-mail и сложности пароля).

15. Интеграция веб-приложения с базой данных

Ход занятия: Настройка соединения с БД. Использование ORM для выполнения запросов без написания чистого SQL.

Задание: Реализовать выборку данных из базы с использованием фильтров, заданных пользователем через интерфейс.

16. Реализация CRUD-операций

Ход занятия: Настройка ресурсных маршрутов и методов контроллера для полного цикла управления записями (создание, чтение, правка, удаление).

Задание: Создать административный интерфейс для управления контентом (например, новостной лентой).

17. Реализация механизма аутентификации пользователей

Ход занятия: Использование встроенных инструментов платформы для входа и регистрации. Настройка защищенного хранения паролей.

Задание: Создать функционал авторизации, позволяющий пользователю входить в свою учетную запись.

18. Реализация механизма авторизации пользователей

Ход занятия: Проверка прав доступа к ресурсам. Разграничение контента для разных ролей (администратор, редактор, пользователь).

Задание: Закрыть доступ к функции удаления данных для всех пользователей, кроме администратора.

19. Разработка REST API с использованием фреймворка

Ход занятия: Проектирование API-эндпоинтов. Использование ресурсов для преобразования данных моделей в JSON.

Задание: Реализовать метод API, который возвращает список последних событий в формате JSON.

20. Работа с JSON-данными

Ход занятия: Парсинг входящих JSON-запросов и формирование структурированных ответов. Управление заголовками Content-Type.

Задание: Написать функцию, которая принимает JSON с внешнего сервиса и сохраняет полученные данные в базу данных приложения.

21. Обработка ошибок приложения

Ход занятия: Использование блоков перехвата исключений. Настройка кастомных страниц для стандартных ошибок (404, 403, 500).

Задание: Настроить систему так, чтобы при отсутствии запрашиваемого товара пользователю выводилась оформленная страница с предложением других товаров.

22. Реализация логирования

Ход занятия: Конфигурирование каналов логирования. Запись системных событий, ошибок и действий пользователей в файлы или базу данных.

Задание: Реализовать запись в лог всех попыток несанкционированного доступа к панели управления.

23. Тестирование функциональности веб-приложения

Ход занятия: Написание автоматизированных тестов для проверки работы основных функций приложения (Unit и Feature тесты).

Задание: Создать тест, который имитирует отправку формы и проверяет наличие новой записи в базе данных.

24. Отладка веб-приложения

Ход занятия: Использование инструментов дебаггинга (панели отладки, просмотр переменных в реальном времени). Анализ стека вызовов при ошибках.

Задание: Найти и исправить программную ошибку в предоставленном модуле, используя инструменты отладки платформы.

25. Оптимизация структуры проекта

Ход занятия: Рефакторинг кода. Вынос повторяющейся логики в отдельные модули и провайдеры. Приведение проекта к стандартам платформы.

Задание: Провести рефакторинг контроллера, сократив объем кода за счет использования моделей или сервисов.

26. Оптимизация производительности веб-приложения

Ход занятия: Использование механизмов кэширования (данных, маршрутов, конфигов). Оптимизация запросов к БД (избавление от N+1).

Задание: Настроить кэширование для главной страницы сайта и замерить изменение времени отклика сервера.

27. Подготовка веб-приложения к развертыванию

Ход занятия: Финализация настроек. Сборка фронтенд-ресурсов (production build). Проверка безопасности переменных окружения.

Задание: Подготовить проверочный список (чек-лист) готовности приложения к переносу на реальный сервер.

28. Настройка конфигурации приложения

Ход занятия: Управление настройками для разных окружений (Local, Testing, Production). Защита чувствительных данных.

Задание: Сформировать файл `.env.production` с настройками для боевого сервера.

29. Развертывание веб-приложения на сервере

Ход занятия: Перенос файлов на хостинг/сервер. Запуск миграций базы данных. Настройка автоматического запуска процессов.

Задание: Осуществить деплой проекта на виртуальный сервер и убедиться в его работоспособности по внешнему адресу.

30. Настройка серверной среды для работы приложения

Ход занятия: Конфигурирование веб-сервера (Nginx/Apache), установка необходимых расширений и служб на сервере.

Задание: Создать конфигурационный файл веб-сервера, обеспечивающий корректную работу маршрутизации фреймворка.

31. Интеграция приложения с внешними сервисами

Ход занятия: Изучение документации внешних систем. Использование HTTP-клиентов для взаимодействия с другими ресурсами.

Задание: Интегрировать в приложение внешнюю систему комментариев или виджет социальной сети.

32. Использование API сторонних сервисов

Ход занятия: Взаимодействие между серверами. Отправка запросов к сторонним API (например, получение курса валют или погоды).

Задание: Реализовать эндпоинт, который запрашивает данные с публичного API и пересылает их клиенту.

33. Настройка безопасности веб-приложения

Ход занятия: Настройка политики безопасности контента (CSP), безопасных куки и заголовков. Ограничение скорости запросов (Rate Limiting).

Задание: Реализовать ограничение на количество попыток входа в систему с одного IP-адреса.

34. Реализация защиты от распространенных веб-уязвимостей

Ход занятия: Проверка и усиление защиты от SQL-инъекций, XSS, CSRF и подделки параметров.

Задание: Провести аудит безопасности созданной формы и внедрить дополнительные механизмы фильтрации данных.

35. Тестирование безопасности приложения

Ход занятия: Использование инструментов сканирования уязвимостей. Ручное тестирование векторов атак.

Задание: Провести проверку приложения на устойчивость к XSS-атакам через поля пользовательского ввода.

36. Подготовка технической документации проекта

Ход занятия: Описание архитектуры, структуры базы данных и программных интерфейсов (API). Составление инструкций для пользователя.

Задание: Составить схему базы данных и описание основных методов API разработанного проекта.

37. Итоговая практическая работа по разработке веб-приложения

Ход занятия: Комплексная разработка проекта «с нуля» до рабочего состояния, включающая все изученные компоненты платформы.

Задание: Разработать веб-ресурс (например, систему бронирования или личный блог) по индивидуальному заданию.

38. Тестирование итогового проекта

Ход занятия: Проведение приемочного тестирования. Проверка соответствия реализованного функционала исходному заданию.

Задание: Написать и выполнить набор тестовых сценариев для проверки всех разделов итогового проекта.

39. Анализ разработанного веб-приложения

Ход занятия: Оценка качества кода, архитектурных решений и удобства интерфейса. Выявление слабых мест.

Задание: Подготовить краткий аналитический отчет с оценкой сильных и слабых сторон разработанного приложения.

40. Оптимизация итогового проекта

Ход занятия: Устранение выявленных в ходе анализа недостатков. Финальная шлифовка кода и повышение производительности.

Задание: Выполнить оптимизацию самого медленного запроса или интерфейсного решения в итоговом проекте.

41. Подготовка отчета по практическим работам

Ход занятия: Сбор и систематизация результатов всех выполненных заданий. Оформление пояснительной записки.

Задание: Оформить сводный отчет по дисциплине, включающий описания и результаты всех практических работ.

42. Защита разработанного веб-проекта

Ход занятия: Демонстрация работы приложения. Ответы на вопросы по реализации программного кода и использованным технологиям.

Задание: Провести презентацию итогового проекта, показав работу всех его модулей.

43. Итоговая доработка веб-проекта

Ход занятия: Внесение исправлений по результатам защиты и замечаниям экспертов.

Задание: Исправить выявленные в ходе защиты недочеты в программном коде проекта.

44. Анализ результатов разработки

Ход занятия: Сравнение достигнутых показателей с планируемыми. Оценка полученного опыта.

Задание: Написать заключение к проекту, отражающее основные сложности и найденные способы их решения.

45. Подготовка итогового отчета

Ход занятия: Окончательное оформление всей документации по курсу для сдачи на проверку.

Задание: Сформировать финальный пакет документов (код, документация, отчет) в электронном виде.

Курсовой проект

1. Общие положения

Курсовой проект является важной формой учебной деятельности студентов и направлен на закрепление теоретических знаний и формирование практических навыков разработки программных решений на стороне сервера.

Цель курсового проекта — приобретение студентами практического опыта разработки серверной части информационного ресурса с использованием современных языков программирования, библиотек и фреймворков.

В процессе выполнения курсового проекта студент должен продемонстрировать способность самостоятельно анализировать техническое задание, проектировать структуру программного решения, реализовывать программный код и выполнять его отладку.

2. Цели и задачи курсового проекта

Основной целью курсового проекта является формирование профессиональных компетенций в области разработки серверных программных решений.

Задачи курсового проекта:

- изучение предметной области разрабатываемого информационного ресурса;
- анализ и формализация требований технического задания;
- проектирование архитектуры программного решения;
- разработка программного кода на стороне сервера;
- использование библиотек и фреймворков программной платформы;
- организация взаимодействия между программными модулями;
- реализация взаимодействия с базой данных или внешними сервисами;
- проведение тестирования и отладки программного обеспечения;
- подготовка технической документации.

3. Формируемые профессиональные компетенции

В результате выполнения курсового проекта студент должен продемонстрировать следующие профессиональные компетенции:

ПК 1. Создавать программный код на стороне сервера в соответствии с техническим заданием (спецификацией) с использованием языков программирования, библиотек и фреймворков.

Студент должен уметь:

- разрабатывать серверные программные модули;
- использовать современные программные платформы;
- применять библиотеки и фреймворки;
- реализовывать обработку запросов клиентов;
- обеспечивать взаимодействие с базой данных.

ПК 2. Осуществлять отладку программного кода на стороне сервера на уровне программных модулей, межмодульных взаимодействий и взаимодействий с окружением.

Студент должен уметь:

- выявлять ошибки в программном коде;
- проводить тестирование модулей;

- анализировать межмодульное взаимодействие;
- выполнять диагностику взаимодействия с внешними системами;
- применять средства логирования и отладки.

4. Тематика курсовых проектов

Тематика курсовых проектов должна быть связана с разработкой серверной части информационных систем или веб-ресурсов.

Примеры тем:

- разработка серверной части интернет-магазина;
- разработка REST API для информационной системы;
- разработка системы управления контентом;
- разработка сервера обработки пользовательских данных;
- разработка системы регистрации и авторизации пользователей;
- разработка сервера для мобильного приложения.

Тема курсового проекта утверждается преподавателем.

5. Этапы выполнения курсового проекта

Работа над курсовым проектом выполняется поэтапно.

1. Анализ технического задания

На данном этапе студент:

- изучает требования к информационному ресурсу;
- определяет основные функции системы;
- формирует список функциональных требований.

2. Проектирование системы

Включает:

- разработку архитектуры приложения;
- проектирование структуры модулей;
- разработку схемы базы данных;
- описание взаимодействия компонентов системы.

3. Разработка программного кода

На данном этапе выполняется:

- создание серверного приложения;
- реализация основных модулей;
- реализация API или серверной логики;
- интеграция с базой данных.

4. Тестирование и отладка

Проводится:

- тестирование программных модулей;
- проверка взаимодействия компонентов;
- выявление и исправление ошибок;
- оптимизация работы системы.

5. Подготовка пояснительной записки

Студент оформляет документацию, описывающую:

- структуру системы;
- алгоритмы работы;
- программный код;
- результаты тестирования.

6. Структура пояснительной записки

Пояснительная записка должна содержать следующие разделы:

1. Титульный лист
2. Задание на курсовой проект
3. Содержание
4. Введение
5. Анализ предметной области
6. Проектирование системы
7. Реализация программного решения
8. Тестирование и отладка
9. Заключение
10. Список используемых источников
11. Приложения (листинг программного кода, схемы, диаграммы)

7. Требования к программному продукту

Разрабатываемое программное обеспечение должно:

- соответствовать техническому заданию;
- иметь модульную структуру;
- обеспечивать обработку пользовательских запросов;
- поддерживать взаимодействие с базой данных;
- корректно работать в программной среде.

Допускается использование следующих технологий:

- **Node.js**
- **Python**
- **PHP**
- **Java**
- **C#**
- серверные фреймворки (Django, Laravel, Spring, Express и др.).

8. Критерии оценки курсового проекта

Оценка курсового проекта осуществляется по следующим критериям:

Критерий	Максимальный балл
Соответствие техническому заданию	20
Качество программного кода	20
Архитектура и структура проекта	15
Реализация функциональности	15
Тестирование и отладка	10
Оформление пояснительной записки	10
Защита проекта	10

9. Защита курсового проекта

Защита курсового проекта проводится в форме презентации и демонстрации разработанного программного продукта.

Студент должен:

- представить разработанный информационный ресурс;
- объяснить архитектуру системы;
- продемонстрировать работу основных функций;
- ответить на вопросы преподавателя.

10. Рекомендуемая литература

1. Основы разработки веб-приложений.
2. Документация используемых программных платформ.
3. Руководства по выбранным фреймворкам.
4. Учебные пособия по серверному программированию.