

Рабочая программа учебной дисциплины **Б1.В.08 Разработка Web-приложений**, компонента основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** направленность (профиль) **«Информационные системы в экономике»**, направлена на обеспечение у обучающегося способности осуществлять профессиональную деятельность в соответствующей области и сферах профессиональной деятельности, в том числе на их практическую подготовку с учётом рабочей программы воспитания и календарного плана воспитательной работы Частном учреждении высшего образования **«Высшая школа предпринимательства (институт)»** (далее — **ЧУВО «ВШП»**).

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

Настоящая рабочая программа учебной дисциплины устанавливает требования к результатам обучения студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей и студентов направления подготовки 09.03.02 Информационные системы и технологии.

Программа учебной дисциплины разработана в соответствии с ФГОС ВО, утвержденного приказом Минобрнауки России от 19.09.2017 № 926 «Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриата по направлению подготовки 09.03.02 Информационные системы и технологии», основной профессиональной образовательной программой высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии, направленность (профиль) Информационные системы в экономике.

2. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Целью изучения дисциплины «Разработка Web-приложений» является формирование у студентов практических навыков и целостного представления о методах и подходах, используемых в web-разработке; формирование умения и навыков работы с web-приложениями; изучение программных средств web-разработки.

Для этого в рамках дисциплины решаются следующие задачи:

- изучить основы построения и функционирования сети Интернет;
- изучить эволюцию концепций Web;
- изучить основы HTML и CSS;
- изучить технологии разработки web-сайтов;
- изучить основы разработки интернет-приложений с помощью языка PHP.

3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина является компонентом части, формируемой участниками образовательных отношений Блока 1 основной профессиональной образовательной программы высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии, направленность (профиль) — Информационные системы в экономике.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ В РАМКАХ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций.

В результате освоения учебной дисциплины обучающийся должен демонстрировать следующие результаты обучения: УК-2, ПК-1, ПК-2, ПК-4.

Таблица 1. Результаты обучения

Код компетенции	Наименование компетенции	Индекс и наименование индикатора содержания компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Способен определять круг задач в рамках поставленной цели	Знать: - Методы и техники формулирования задач. - Принципы декомпозиции задач на подзадачи. Уметь: - Определять цели и задачи проекта. - Делить задачи на этапы и определять приоритеты. Владеть: - Навыками структурирования и организации задач. - Навыками работы с целями и задачами проекта.

		<p>УК-2.2 Способен выбирать оптимальные способы решения задач, исходя из правовых норм, ресурсов и ограничений</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы выбора оптимальных решений. - Правовые нормы и ограничения, влияющие на процесс разработки. <p>Уметь:</p> <ul style="list-style-type: none"> - Оценивать доступные ресурсы и ограничения проекта. - Выбирать и обосновывать оптимальные методы и инструменты для решения задач. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками анализа и оценки рисков при выборе решений. - Навыками работы с правовыми нормами и стандартами в области разработки программного обеспечения.
ПК-1	Разработка и отладка программного кода	<p>ПК-1.1 Способен разрабатывать программный код</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Основные принципы и методы разработки программного кода. - Синтаксис и семантику используемых языков программирования (Python, C/C++, JS). <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать корректный и эффективный программный код. - Использовать стандартные библиотеки и фреймворки для разработки. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками написания и документирования кода. - Навыками применения передовых практик программирования (SOLID, DRY).
		<p>ПК-1.2 Способен отлаживать программный код</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Основные методы и инструменты отладки программного кода. - Принципы и техники выявления и исправления ошибок. <p>Уметь:</p> <ul style="list-style-type: none"> - Использовать отладчики и инструменты профилирования для поиска ошибок. - Анализировать и исправлять ошибки в программном коде. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками эффективной отладки и тестирования кода. - Навыками применения различных техник отладки (пошаговая отладка, логирование).
ПК-2	Проверка работоспособности и рефакторинг кода программного обеспечения	<p>ПК-2.1 Способен проверять работоспособность программного кода</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы тестирования программного кода. - Основы юнит-тестирования и функционального тестирования. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать и выполнять тестовые сценарии для проверки работоспособности кода. - Использовать тестовые фреймворки и инструменты для автоматизированного тестирования. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками написания и выполнения тестов.

			<ul style="list-style-type: none"> - Навыками анализа результатов тестирования и устранения обнаруженных проблем.
		<p>ПК-2.2 Способен рефакторить программный код</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы рефакторинга программного кода. - Паттерны и антипаттерны проектирования кода. <p>Уметь:</p> <ul style="list-style-type: none"> - Проводить анализ и улучшение структуры кода. - Вносить изменения в код, сохраняя его функциональность. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками применения рефакторинга для улучшения читаемости и поддерживаемости кода. - Навыками использования инструментов для автоматизированного рефакторинга.
ПК-3	Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта	<p>ПК-3.1 Способен интегрировать программные модули и компоненты</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы интеграции программных модулей. - Основы межмодульного взаимодействия и совместимости. <p>Уметь:</p> <ul style="list-style-type: none"> - Планировать и проводить интеграцию модулей в общий проект. - Решать проблемы, возникающие при интеграции различных компонентов. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками использования инструментов и платформ для интеграции. - Навыками написания интеграционных тестов.
		<p>ПК-3.2 Способен проверять работоспособность выпусков программного продукта</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Методы и инструменты для проверки выпусков программного продукта. - Основы тестирования и валидации программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Проводить проверку и валидацию программного продукта. - Анализировать результаты тестирования и устранять выявленные дефекты. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками работы с инструментами для проверки и тестирования выпусков. - Навыками документирования результатов тестирования и предложений по улучшению.
ПК-4	Разработка требований и проектирование программного обеспечения	<p>ПК-4.1 Способен разрабатывать требования к программному обеспечению</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Методы и техники сбора и анализа требований. - Принципы написания требований и спецификаций. <p>Уметь:</p> <ul style="list-style-type: none"> - Собирать и документировать требования к программному обеспечению. - Анализировать и уточнять требования с заинтересованными сторонами. <p>Владеть:</p>

			<ul style="list-style-type: none"> - Навыками написания четких и измеримых требований. - Навыками работы с инструментами для управления требованиями.
		ПК-4.2 Способен проектировать программное обеспечение	Знать: <ul style="list-style-type: none"> - Принципы и методы проектирования программного обеспечения. - Основы архитектурных шаблонов и стилей. Уметь: <ul style="list-style-type: none"> - Разрабатывать архитектуру и дизайн программного обеспечения. - Создавать диаграммы и модели для визуализации архитектуры. Владеть: <ul style="list-style-type: none"> - Навыками применения архитектурных паттернов. - Навыками использования инструментов для проектирования и моделирования программного обеспечения.

5. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5.1. Объем дисциплины в зачетных единицах с указанием количества академических часов

Общая трудоемкость учебной дисциплины составляет 8 зачетных единиц, 288 часов, включая все формы контактной и самостоятельной работы обучающихся.

Объем дисциплины по учебному плану составляет – 8 зачётных единицы = 288 академических часов.

Контактная работа обучающегося (студенты) с научно-педагогическим работником организации (всего) - 96 академических часов,

в том числе:

Лекционные занятия (Лек.) - 38 академических часов,

Практические занятия (Пр.) - 56 академических часов,

Консультации (Конс.) - 2 академических часа.

Самостоятельная работа обучающегося (студента):

Самостоятельная работа (СР) - 147 академический час,

Текущий контроль успеваемости

и промежуточно-заочная аттестация обучающегося (студента):

Часы на контроль - 45 академических часов.

Таблица 2. Объём дисциплины

№ п/п	Раздел/тема дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (студентов), и трудоёмкость (в ак. часах)				Коды формируемых компетенций
		Виды учебных занятий по дисциплине			Самостоятельная работа	
		Лек.	Пр.	Конс.		
7 семестр / 4курс						

1	Тема 1. Введение в разработку Web-приложений	2	4	-	15	УК-2.1, УК-2.2
2	Тема 2. Основы HTML и CSS	4	4	-	15	ПК-1.1, ПК-4.2
3	Тема 3. JavaScript для разработки Web-приложений	4	4	-	15	ПК-1.1, ПК-1.2
4	Тема 4. Введение в клиент-серверную архитектуру	4	4	-	15	ПК-3.1, ПК-3.2
5	Тема 5. Основы работы с серверной частью на примере Node.js	4	4	-	15	ПК-1.1, ПК-3.1
6	Тема 6. Введение в базы данных	4	4	-	15	ПК-4.1, ПК-3.2
7	Тема 7. Работа с REST API	4	4	-	15	ПК-3.1, ПК-3.2
8	Тема 8. Введение в популярные фреймворки и библиотеки	4	4	-	15	ПК-1.1, ПК-4.2
9	Тема 9. Подготовка к зачету		10	-	15	УК-2.1, УК-2.2, ПК-1.1, ПК-1.2, ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2
8 семестр / 4 курс						
10	Тема 10. Продвинутое возможности JavaScript	1	2	-	2	ПК-1.1, ПК-1.2
11	Тема 11. Интерактивность и взаимодействие с пользователем	1	2	-	2	ПК-1.1, ПК-3.1
12	Тема 12. Безопасность Web-приложений	2	2	-	2	ПК-2.1, ПК-2.2
13	Тема 13. Оптимизация и производительность Web-приложений	2	2	-	2	ПК-2.2, ПК-3.1
14	Тема 14. Тестирование Web-приложений	1	2	-	2	ПК-2.1, ПК-3.2
15	Тема 15. Развертывание и публикация Web-приложений	1	2	-	2	ПК-3.1, ПК-3.2
16	Тема 16. Итоговое занятие и подготовка к экзамену		2	2	2	УК-2.1, УК-2.2, ПК-1.1, ПК-1.2, ПК-2.1, ПК-2.2, ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2
ИТОГО аудиторных часов/СР:		7-8 семестры/ 4 курс		96 ак. часов		147 ак. часа
Часы на контроль		7-8 семестры/ 4 курс		9 ак. час (форма промежуточной аттестации – зачет с оценкой – 7 семестр) 36 ак. час (форма промежуточной аттестации – экзамен – 8 семестр)		
ВСЕГО ак. часов:		7-8 семестры/ 4 курс		288 академических часов		

5.2. Тематическое содержание дисциплины

* количество академических часов и виды занятий представлены в таблице № 2.

7 семестр

Тема 1: Введение в разработку Web-приложений

Изучение истории и развития Web-технологий, обзор современных стандартов и протоколов (HTML, CSS, HTTP, HTTPS). Основные отличия между Web-приложениями и традиционными приложениями, преимущества и вызовы.

Тема 2: Основы HTML и CSS

Изучение основ языка разметки HTML (структура документа, основные теги и атрибуты). Основы каскадных таблиц стилей (CSS) (синтаксис, селекторы, свойства). Введение в современные возможности HTML5 и CSS3.

Тема 3: JavaScript для разработки Web-приложений

Изучение основ языка программирования JavaScript (синтаксис, типы данных, операторы, функции, объекты). Введение в манипуляцию DOM, обработка событий и основное взаимодействие с HTML и CSS через JavaScript.

Тема 4: Введение в клиент-серверную архитектуру

Понятие клиент-серверной архитектуры, HTTP-запросы и ответы, методы GET и POST. Основы работы с AJAX для асинхронного взаимодействия с сервером.

Тема 5: Основы работы с серверной частью на примере Node.js

Изучение основ Node.js (установка, настройка, модули, создание простого сервера). Введение в Express.js для создания RESTful API, маршрутизация и обработка запросов.

Тема 6: Введение в базы данных

Изучение основ реляционных баз данных (MySQL, PostgreSQL) и нереляционных баз данных (MongoDB). Основные операции CRUD (Create, Read, Update, Delete), подключение к базе данных из Node.js.

Тема 7: Работа с REST API

Создание и использование REST API для взаимодействия между клиентом и сервером. Формирование и обработка запросов, работа с JSON-данными.

Тема 8: Введение в популярные фреймворки и библиотеки

Обзор популярных клиентских фреймворков (React, Angular, Vue.js). Основы работы с Vue.js: компоненты, состояние, жизненный цикл компонентов.

Тема 9: Подготовка к зачету

Повторение пройденного материала, решение типовых задач и вопросов для подготовки к зачету.

8 семестр

Тема 10: Продвинутое возможности JavaScript

Изучение продвинутых возможностей JavaScript (ES6+, промисы, async/await, модули). Работа с асинхронным кодом, улучшенные подходы к обработке данных.

Тема 11: Интерактивность и взаимодействие с пользователем

Использование клиентских библиотек для повышения интерактивности (jQuery, D3.js). Введение в анимации с CSS и JavaScript, работа с формами и валидация пользовательского ввода.

Тема 12: Безопасность Web-приложений

Основы безопасности Web-приложений (XSS, CSRF, SQL-инъекции). Методы защиты данных, аутентификация и авторизация пользователей.

Тема 13: Оптимизация и производительность Web-приложений

Основные принципы оптимизации Web-приложений (сжатие файлов, кеширование, минимизация запросов). Введение в инструменты для анализа производительности (Lighthouse, WebPageTest).

Тема 14: Тестирование Web-приложений

Изучение основ тестирования (юнит-тесты, интеграционные тесты, e2e тестирование). Введение в популярные инструменты для тестирования (Jest, Mocha, Cypress).

Тема 15: Развертывание и публикация Web-приложений

Подготовка приложения к развертыванию, создание и настройка серверного окружения. Публикация приложения на популярных хостингах и платформах (Heroku, Vercel, AWS).

Тема 16: Итоговое занятие и подготовка к экзамену

Повторение и обобщение пройденного материала, решение типовых задач и вопросов для подготовки к экзамену, обсуждение практических аспектов разработки и публикации Web-приложений.

5.2.1 Содержание практических занятий

Таблица 3

№ п/п	Наименование темы (раздела) дисциплины	Содержание практического занятия
7 семестр		
1	Практическое занятие 1: Введение в разработку Web-приложений	Задание: Установить и настроить среду разработки (Visual Studio Code, WebStorm или аналогичную). Создать простую HTML-страницу и применить к ней CSS-стили. Цель: Ознакомиться с инструментами разработки и базовыми элементами HTML и CSS.
2	Практическое занятие 2: Основы HTML и CSS	Задание: Создать многостраничный сайт с использованием различных HTML-тегов и CSS-стилей (таблицы, списки, формы, фреймы). Цель: Закрепить знания о структуре HTML-документа и основах стилизации с помощью CSS.
3	Практическое занятие 3: Введение в JavaScript	Задание: Написать несколько скриптов на JavaScript, которые взаимодействуют с HTML-элементами (изменение стилей, добавление и удаление элементов, обработка событий). Цель: Научиться использовать JavaScript для динамического изменения контента на веб-странице.
4	Практическое занятие 4: Работа с DOM и событиями	Задание: Создать интерактивную страницу, используя DOM-манипуляции и обработчики событий (нажатия кнопок, ввод текста и т.д.). Цель: Освоить методы работы с DOM и обработки событий в JavaScript.
5	Практическое занятие 5: Введение в клиент-серверную архитектуру	Задание: Написать простой сервер на Node.js, который отвечает на HTTP-запросы. Создать клиентскую часть, которая взаимодействует с сервером через AJAX. Цель: Понять основные принципы клиент-серверного взаимодействия.
6	Практическое занятие 6: Основы работы с базами данных	Задание: Установить и настроить реляционную базу данных (MySQL). Создать таблицу, выполнить операции CRUD (Create, Read, Update, Delete) через Node.js. Цель: Научиться работать с базами данных и выполнять основные операции с данными.
7	Практическое занятие 7: Работа с REST API	Задание: Создать RESTful API на базе Express.js для выполнения CRUD операций. Реализовать взаимодействие клиентской части с API. Цель: Освоить разработку и использование REST API для веб-приложений.
8	Практическое занятие 8: Введение в Vue.js	Задание: Создать простое одностраничное приложение на Vue.js. Изучить компоненты, состояние и пропсы.

		Цель: Ознакомиться с основными концепциями Vue.js и их применением.
9	Практическое занятие 9: Подготовка к зачету	Задание: Решение типовых задач и вопросов для закрепления пройденного материала. Цель: Повторение и обобщение изученного материала для успешной сдачи зачета.
8 семестр		
10	Практическое занятие 10: Продвинутое возможности JavaScript	Задание: Написать несколько скриптов с использованием новых возможностей ES6+ (стрелочные функции, промисы, async/await, модули). Цель: Освоить продвинутое возможности языка JavaScript для более эффективного программирования.
11	Практическое занятие 11: Интерактивность и взаимодействие с пользователем	Задание: Создать интерактивное приложение с использованием библиотеки jQuery. Реализовать анимации и динамическое изменение контента. Цель: Научиться использовать библиотеки для повышения интерактивности веб-приложений.
12	Практическое занятие 12: Безопасность Web-приложений	Задание: Реализовать основные методы защиты веб-приложений от XSS, CSRF, и SQL-инъекций. Настроить аутентификацию и авторизацию пользователей. Цель: Освоить методы обеспечения безопасности в веб-приложениях.
13	Практическое занятие 13: Оптимизация и производительность	Задание: Провести оптимизацию веб-страницы (сжатие файлов, кеширование, минимизация запросов). Использовать инструменты для анализа производительности. Цель: Научиться улучшать производительность веб-приложений.
14	Практическое занятие 14: Тестирование Web-приложений	Задание: Написать юнит-тесты и интеграционные тесты для веб-приложения с использованием Jest и Cypress. Цель: Освоить методы тестирования веб-приложений для обеспечения их корректной работы.
15	Практическое занятие 15: Развертывание и публикация	Задание: Подготовить веб-приложение к развертыванию, создать и настроить серверное окружение, опубликовать приложение на Heroku или Vercel. Цель: Научиться процессу развертывания и публикации веб-приложений.
16	Практическое занятие 16: Итоговое занятие и подготовка к экзамену	Задание: Решение типовых задач и вопросов для закрепления пройденного материала, обсуждение практических аспектов разработки и публикации веб-приложений. Цель: Повторение и обобщение пройденного материала для успешной сдачи экзамена.

5.2.2 Содержание самостоятельной работы

Таблица 4

№ п/п	Наименование темы (раздела) дисциплины	Содержание самостоятельной работы	Форма контроля
7 семестр			
1	Самостоятельная работа 1: Введение в разработку Web-приложений	Задание: Прочитать рекомендованные материалы по истории и развитию Web-технологий, подготовить краткий обзор. Цель: Понять исторический контекст и эволюцию веб-технологий.	Подготовка реферата.
2	Самостоятельная работа 2: Основы HTML и CSS	Задание: Изучить материалы по основам HTML и CSS, выполнить дополнительные упражнения, создавая различные элементы веб-страниц. Цель: Освоить основы HTML и CSS для создания структурированных и стилизованных веб-страниц.	Ответы на контрольные вопросы, выполнение практического задания.
3	Самостоятельная работа 3: JavaScript для разработки Web-приложений	Задание: Изучить материалы по основам JavaScript, выполнить упражнения по манипуляции DOM и обработке событий. Цель: Научиться использовать JavaScript для добавления интерактивности в веб-страницы.	Выполнение практического задания.
4	Самостоятельная работа 4: Клиент-серверная архитектура	Задание: Прочитать материалы по клиент-серверной архитектуре, выполнить практическое задание по созданию простого HTTP-сервера. Цель: Понять основные принципы клиент-серверного взаимодействия.	Выполнение практического задания.

5	Самостоятельная работа 5: Основы работы с серверной частью на примере Node.js	Задание: Изучить материалы по Node.js и Express.js, написать простое приложение с использованием этих технологий. Цель: Освоить основные концепции и инструменты для разработки серверной части веб-приложений.	Выполнение практического задания.
6	Самостоятельная работа 6: Работа с базами данных	Задание: Прочитать материалы по реляционным и нереляционным базам данных, выполнить упражнения по выполнению операций CRUD. Цель: Научиться работать с различными типами баз данных и выполнять основные операции с данными.	Выполнение практического задания.
7	Самостоятельная работа 7: Работа с REST API	Задание: Изучить материалы по созданию и использованию REST API, выполнить упражнения по созданию RESTful сервиса. Цель: Освоить разработку REST API для взаимодействия клиентской и серверной частей веб-приложений.	Выполнение практического задания.
8	Самостоятельная работа 8: Введение в Vue.js	Задание: Прочитать материалы по Vue.js, создать простое приложение на Vue.js. Цель: Освоить основы работы с Vue.js для создания динамических веб-приложений.	Выполнение практического задания.
9	Самостоятельная работа 9: Подготовка к зачету	Задание: Решение типовых задач и вопросов для закрепления пройденного материала. Цель: Повторение и обобщение изученного материала для успешной сдачи зачета.	Тест.
8 семестр			
10	Самостоятельная работа 10: Продвинутое возможности JavaScript	Задание: Изучить материалы по продвинутым возможностям JavaScript (ES6+), выполнить упражнения по использованию новых возможностей. Цель: Освоить продвинутое возможности JavaScript для более эффективного программирования.	Выполнение практического задания.
11	Самостоятельная работа 11: Интерактивность и взаимодействие с пользователем	Задание: Прочитать материалы по использованию jQuery и анимациям, выполнить упражнения по созданию интерактивных элементов. Цель: Научиться использовать библиотеки для повышения интерактивности веб-приложений.	Выполнение практического задания.
12	Самостоятельная работа 12: Безопасность Web-приложений	Задание: Изучить материалы по безопасности веб-приложений, выполнить упражнения по защите от XSS, CSRF и SQL-инъекций. Цель: Освоить методы обеспечения безопасности в веб-приложениях.	Выполнение практического задания.
13	Самостоятельная работа 13: Оптимизация и производительность	Задание: Прочитать материалы по оптимизации веб-приложений, выполнить упражнения по сжатию файлов и настройке кеширования. Цель: Научиться улучшать производительность веб-приложений.	Выполнение практического задания.
14	Самостоятельная работа 14: Тестирование Web-приложений	Задание: Изучить материалы по тестированию веб-приложений, выполнить упражнения по написанию юнит-тестов и интеграционных тестов. Цель: Освоить методы тестирования веб-приложений для обеспечения их корректной работы.	Выполнение практического задания.
15	Самостоятельная работа 15: Развертывание и публикация	Задание: Изучить процесс развертывания веб-приложений, подготовить и опубликовать приложение на Heroku или Vercel. Цель: Научиться процессу развертывания и публикации веб-приложений.	Выполнение практического задания.
16	Самостоятельная работа 16: Итоговое занятие и подготовка к экзамену	Задание: Решение типовых задач и вопросов для закрепления пройденного материала, обсуждение практических аспектов разработки и публикации веб-приложений. Цель: Повторение и обобщение пройденного материала для успешной сдачи экзамена.	Экзамен.

6.Оценочные материалы по дисциплине

Оценочные материалы находятся в документе «Оценочные материалы по дисциплине «Разработка Web-приложений».

7.Методические материалы для обучающихся по освоению дисциплины (модуля)

А) Рекомендации обучающемуся (студенту) по работе с конспектом после лекции

Какими бы замечательными качествами в области методики ни обладал лектор, какое бы большое значение на занятиях ни уделял лекции слушатель, глубокое понимание материала достигается только путем самостоятельной работы над ним. Самостоятельную работу следует начинать с доработки конспекта, желательно в тот же день, пока время не стерло содержание лекции из памяти (через 10 часов после лекции в памяти остается не более 30-40 % материала). С целью доработки необходимо в первую очередь прочитать записи, восстановить текст в памяти, а также исправить описки, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Далее прочитать материал по рекомендуемой литературе, разрешая в ходе чтения, возникшие ранее затруднения, вопросы, а также дополнения и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. В ходе доработки конспекта углубляются, расширяются и закрепляются знания, а также дополняется, исправляется и совершенствуется конспект. Подготовленный конспект и рекомендуемая литература используется при подготовке к практическому занятию. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений и формул, к решению примеров, задач, к ответам на вопросы, предложенные в конце лекции преподавателем или помещенные в рекомендуемой литературе. Примеры, задачи, вопросы по теме являются материалом самоконтроля. Непременным условием глубокого усвоения учебного материала является знание основ, на которых строится изложение материала. Обычно преподаватель напоминает, какой ранее изученный материал и в какой степени требуется подготовить к очередному занятию. Эта рекомендация, как и требование систематической и серьезной работы над всем лекционным курсом, подлежит безусловному выполнению. Потери логической связи как внутри темы, так и между ними приводит к негативным последствиям: материал учебной дисциплины перестает основательно восприниматься, а творческий труд подменяется утомленным переписыванием. Обращение к ранее изученному материалу не только помогает восстановить в памяти известные положения, выводы, но и приводит разрозненные знания в систему, углубляет и расширяет их. Каждый возврат к старому материалу позволяет найти в нем что-то новое, переосмыслить его с иных позиций, определить для него наиболее подходящее место в уже имеющейся системе знаний. Неоднократное обращение к пройденному материалу является наиболее рациональной формой приобретения и закрепления знаний. Очень полезным, но, к сожалению, еще мало используемым в практике самостоятельной работы, является предварительное ознакомление с учебным материалом. Даже краткое, беглое знакомство с материалом очередной лекции дает многое. Обучающиеся (студенты) получают общее представление о её содержании и структуре, о главных и второстепенных вопросах, о терминах и определениях. Все это облегчает работу на лекции и делает ее целеустремленной.

Б) Рекомендации обучающемуся (студенту) по подготовке к занятиям семинарского типа

Обучающийся (студент) должен чётко уяснить, что именно с лекции начинается его подготовка к лабораторному/ практическому/ семинарскому/ методическому/ клиническому практическому занятию. Вместе с тем, лекция лишь организует мыслительную деятельность, но не обеспечивает глубину усвоения программного материала. При подготовке к такому виду занятий можно выделить 2 этапа:

1-й - организационный,

2-й - закрепление и углубление теоретических знаний.

На первом этапе обучающийся (студент) планирует свою самостоятельную работу, которая включает:

- уяснение задания на самостоятельную работу;
- подбор рекомендованной литературы;
- составление плана работы, в котором определяются основные пункты предстоящей подготовки.

Составление плана дисциплинирует и повышает организованность в работе. Второй этап включает непосредственную подготовку обучающегося (студента) к занятию. Начинать надо с изучения рекомендованной литературы. Необходимо помнить, что на лекции обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы обучающийся (студент) должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. Заканчивать подготовку следует составлением плана (перечня основных пунктов) по изучаемому материалу (вопросу). Такой план позволяет составить концентрированное, сжатое представление по изучаемым вопросам. В процессе подготовки к семинарскому занятию рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретает практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю. Идя на консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения. В начале семинарского занятия обучающиеся (студента) под руководством преподавателя более глубоко осмысливают теоретические положения по теме занятия, раскрывают и объясняют основные явления и факты. В процессе творческого обсуждения и дискуссии вырабатываются умения и навыки использовать приобретенные знания для решения практических задач.

В) Рекомендации по самостоятельной работе обучающегося (студента) над изучаемым материалом

Успешное освоение данного курса базируется на рациональном сочетании нескольких видов учебной деятельности - лекций, семинарских занятий, самостоятельной работы. При этом самостоятельную работу следует рассматривать одним из главных звеньев полноценного высшего образования, на которую отводится значительная часть учебного времени.

Самостоятельная работа студентов складывается из следующих составляющих:

- работа с основной и дополнительной литературой, с материалами интернета и конспектами лекций;
- внеаудиторная подготовка к контрольным работам, выполнение докладов, рефератов и курсовых работ;
- выполнение самостоятельных практических работ;
- подготовка к экзаменам (зачетам) непосредственно перед ними.

Для правильной организации работы необходимо учитывать порядок изучения разделов курса, находящихся в строгой логической последовательности. Поэтому хорошее усвоение одной части дисциплины является предпосылкой для успешного перехода к следующей. Задания, проблемные вопросы, предложенные для изучения дисциплины, в том числе и для самостоятельного выполнения, носят междисциплинарный характер и базируются, прежде всего, на причинно-следственных связях между компонентами окружающего нас мира. В течение семестра, необходимо подготовить рефераты (проекты) с использованием рекомендуемой основной и дополнительной литературы и сдать рефераты для проверки преподавателю. Важным составляющим в изучении данного курса является решение ситуационных задач и работа над проблемно-аналитическими заданиями, что предполагает знание соответствующей научной терминологии и т.д.

Для лучшего запоминания материала целесообразно использовать индивидуальные особенности и разные виды памяти: зрительную, слуховую, ассоциативную. Успешному

запоминанию также способствует приведение ярких свидетельств и наглядных примеров. Учебный материал должен постоянно повторяться и закрепляться.

При выполнении докладов, творческих, информационных, исследовательских проектов особое внимание следует обращать на подбор источников информации и методику работы с ними.

Для успешной сдачи экзамена (зачета) рекомендуется соблюдать следующие правила:

1. Подготовка к экзамену (зачету) должна проводиться систематически, в течение всего семестра.
2. Интенсивная подготовка должна начинаться не позднее, чем за месяц до экзамена.
3. Время непосредственно перед экзаменом (зачетом) лучше использовать таким образом, чтобы оставить последний день свободным для повторения курса в целом, для систематизации материала и доработки отдельных вопросов.

На экзамене высокую оценку получают студенты, использующие данные, полученные в процессе выполнения самостоятельных работ, а также использующие собственные выводы на основе изученного материала.

Учитывая значительный объем теоретического материала, студентам рекомендуется регулярное посещение и подробное конспектирование лекций.

8. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники и учебные пособия, иная учебная литература, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

а) для слабовидящих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- задания для выполнения, а также инструкция о порядке проведения промежуточной аттестации оформляются увеличенным шрифтом;

- задания для выполнения на промежуточной аттестации зачитываются ассистентом;

- письменные задания выполняются на бумаге, надиктовываются ассистенту;

- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;

- студенту для выполнения задания при необходимости предоставляется увеличивающее устройство;

в) для глухих и слабослышащих:

- на промежуточной аттестации присутствует ассистент, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (он помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе записывая под диктовку);

- промежуточно-заочная аттестация проводится в письменной форме;

- обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости поступающим предоставляется звукоусиливающая аппаратура индивидуального пользования;

- по желанию студента промежуточно-заочная аттестация может проводиться в письменной форме;

д) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
- по желанию студента промежуточно-заочная аттестация проводится в устной форме.

Примечание:

а) Для обучающегося (бакалавра), осваивающего учебную дисциплину, обязательный компонент основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии (направленность (профиль) «Информационные системы в экономике»), форма обучения — очно-заочная), одобренной на заседании Учёного совета образовательной организации, утверждённой ректором Частного образовательного учреждения высшего образования «Высшая школа предпринимательства», по индивидуальному учебному плану (при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра)), Институт:**

- разрабатывает, согласовывает с участниками образовательных отношений и утверждает в установленном порядке согласно соответствующему локальному нормативному акту **индивидуальный учебный план** конкретного обучающегося (бакалавра) (*учебный план, обеспечивающий освоение конкретной основной образовательной программы высшего образования на основе индивидуализации её содержания с учётом особенностей и образовательных потребностей конкретного обучающегося (бакалавра)*);

- устанавливает для конкретного обучающегося (бакалавра) по индивидуальному учебному плану **одинаковые дидактические единицы** — элементы содержания учебного материала, изложенного в виде утверждённой в установленном образовательной организацией порядке согласно соответствующему локальному нормативному акту рабочей программы учебной дисциплины, обязательного компонента разработанной и реализуемой Институтом основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** (направленность (профиль) «Информационные системы в экономике»), форма обучения — очно-заочная), как и для обучающего (бакалавра), осваивающего основную образовательную программу высшего образования в учебной группе;

- определяет в индивидуальном учебном плане конкретного обучающегося (бакалавра) **объём учебной дисциплины** с указанием количества академических часов/ ЗЕТ, выделенных на его контактную работу (групповую и (или) индивидуальную работу) с руководящими и (или) научно-педагогическими работниками, реализующими основную образовательную программу высшего образования;

- определяет в индивидуальном учебном плане конкретного обучающегося (бакалавра) количество академических часов/ ЗЕТ по учебной дисциплине, выделенных на его самостоятельную работу (*при необходимости*).

б) Для обучающегося (бакалавра) с ограниченными возможностями здоровья и инвалида, осваивающего учебную дисциплину, обязательный компонент основной профессиональной образовательной программы высшего образования — программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии (направленность (профиль) «Информационные системы в экономике»), форма обучения — очно-заочная), одобренной на заседании Учёного совета образовательной организации, утверждённой ректором Частного образовательного учреждения высшего образования «Высшая школа предпринимательства», (*при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий)*), Институт:**

- разрабатывает, согласовывает с участниками образовательных отношений и утверждает в установленном порядке согласно соответствующему локальному нормативному акту **индивидуальный учебный план** конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья/ инвалида (*при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии*

(нозологий)) (учебный план, обеспечивающий освоение конкретной основной образовательной программы высшего образования на основе индивидуализации её содержания с учётом особенностей и образовательных потребностей конкретного обучающегося (бакалавра));

- устанавливает для конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья содержание образования (**одинаковые дидактические единицы** — элементы содержания учебного материала, как и для обучающегося (бакалавра), осваивающего основную образовательную программу высшего образования в учебной группе) и условия организации обучения, изложенного в виде утверждённой в установленном Институте порядке согласно соответствующему локальному нормативному акту рабочей программы учебной дисциплины, обязательного компонента разработанной и реализуемой им адаптированной основной профессиональной образовательной программы высшего образования - программы бакалавриата по направлению подготовки **09.03.02 Информационные системы и технологии** (направленность (профиль) «Информационные системы в экономике»), форма обучения — очно-заочная), а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида (для конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья/ инвалида (при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий));

- определяет в индивидуальном учебном плане конкретного обучающегося бакалавра) с ограниченными возможностями здоровья/ инвалида (при наличии факта зачисления такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий)) **объём учебной дисциплины** с указанием количества академических часов/ ЗЕТ, выделенных на его контактную работу (групповую и (или) индивидуальную работу) с руководящими и (или) научно-педагогическими работниками, реализующими основную образовательную программу высшего образования;

- определяет в индивидуальном учебном плане конкретного обучающегося (бакалавра) с ограниченными возможностями здоровья/ инвалида (при наличии факта зачисления в образовательную организацию такого обучающегося (бакалавра) с учётом конкретной (конкретных) нозологии (нозологий)) количество академических часов/ ЗЕТ по учебной дисциплине, выделенных на его самостоятельную работу (при необходимости).

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

9.1 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Дукарт, Дж. Программирование на JavaScript / Д. Дукарт. - 2-е изд. - Санкт-Петербург: Питер, 2019. - 800 с. - ISBN 978-5-4461-1234-9.

2. Резиг, Дж. Разработка динамических веб-приложений / Дж. Резиг, Б. Бибо. - 3-е изд. - Москва: Вильямс, 2018. - 912 с. - ISBN 978-5-8459-2096-7.

Дополнительная литература:

1. Евсеев Д.А., Web-дизайн в примерах и задачах [Электронный ресурс] : учебное пособие / Д.А. Евсеев, В.В. Трофимов. - М. : КноРус, 2022. - 263 с. - ISBN 978-5-406-09190-6. - Режим доступа: <https://book.ru/book/942676>

9.2 Используемое программное обеспечение (комплект лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства в соответствии с п.4.3.2. ФГОС ВО 09.03.02):

1. Microsoft Windows 11 Pro или аналогичная ОС, включая дистрибутивы Linux, например Debian, Ubuntu, OpenSuse, в том числе отечественного производства, например ОС Astra Linux Common Edition (Разработчик: АО «НПО РусБИТех»), ОС «РОСА» (Разработчик: «НТЦ ИТ

РОСА»).

2. Microsoft Office 365 или аналогичный офисный пакет, например OpenOffice, LibreOffice, ONLYOFFICE, в том числе отечественного производства, например МойОфис (Разработчик: ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»).

3. Adobe Reader или аналогичный просмотрщик PDF, например Okular, Foxit Reader, в том числе отечественного производства, например Окуляр ГОСТ (Разработчик: ООО «Лаборатория 50»).

4. Google Chrome или аналогичный веб-браузер, например Microsoft Edge, Mozilla Firefox, в том числе отечественного производства, например Яндекс.Браузер (Разработчик: ООО «ЯНДЕКС»).

5. Microsoft Visual Studio Code или аналогичная IDE, например Sublime Text, Eclipse, в том числе отечественного производства

6. PyCharm / IntelliJ IDEA / CLion / WebStorm либо аналогичная IDE полного стека, в том числе отечественного производства

7. MySQL CE 8.0 / MySQL Workbench или аналогичные СУБД, например MS SQL, PostgreSQL, в том числе отечественного производства

8. Android Studio или аналогичная IDE для разработки мобильных приложений, в том числе отечественного производства

9. Figma или аналогичное ПО для подготовки макетов, например Penpot, Lunacy, в том числе отечественного производства.

9.3 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля) (в соответствии с п.4.3.4. ФГОС ВО 09.03.02)

1. Электронно-библиотечная система BOOK.RU [Электронный ресурс]. - Режим доступа: <https://book.ru/>

2. Современный учебник JavaScript [Электронный ресурс]. - Режим доступа: <https://learn.javascript.ru/>

9.4 Базы данных, информационно-справочные и поисковые системы (в соответствии с п.4.3.4. ФГОС ВО 09.03.02)

1. КонсультантПлюс: справочно-поисковая система [Электронный ресурс]. - <http://www.consultant.ru>

2. Мировая цифровая библиотека: <http://wdl.org/ru>

3. Научная электронная библиотека «Scopus»: <https://www.scopus.com>

4. Научная электронная библиотека ScienceDirect: <http://www.sciencedirect.com>

5. Научная электронная библиотека «eLIBRARY»: <https://elibrary.ru>

6. Портал «Гуманитарное образование» <http://www.humanities.edu.ru>

7. Федеральный портал «Российское образование» <http://www.edu.ru>

8. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов» <http://school-collection.edu.ru>

9. Поисковые системы Yandex, Rambler и др.

10. Электронная библиотека Российской Государственной Библиотеки (РГБ): <http://elibrary.rsl.ru>

11. Электронно-библиотечная система <http://www.sciteclibrary.ru>

10. Материально-техническое обеспечение дисциплины

Помещения для проведения всех видов работы, предусмотренных учебным планом, укомплектованы необходимым оборудованием и техническими средствами обучения.

Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для	Адрес (местоположение) помещений для проведения всех видов учебной деятельности, предусмотренной учебным
---	---

<p>самостоятельной работы, с указанием перечня основного оборудования, учебно-наглядных пособий и используемого программного обеспечения</p>	<p>планом (в случае реализации образовательной программы в сетевой форме дополнительно указывается наименование организации, с которой заключен договор)</p>
<p>Специализированная многофункциональная учебная аудитория для проведения учебных занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающихся, с перечнем основного оборудования:</p> <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>
<p>Специализированная многофункциональная учебная аудитория для проведения учебных занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, в том числе, для организации практической подготовки обучающихся, с перечнем основного оборудования:</p> <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>
<p>Помещение для самостоятельной работы обучающихся с перечнем основного оборудования:</p> <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>

<ul style="list-style-type: none"> - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	
<p>Помещение для практических занятий на персональных компьютерах:</p> <ul style="list-style-type: none"> - Столы для обучающихся; - Стулья для обучающихся; - Стол педагогического работника; - Стул педагогического работника; - Компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Ноутбуки с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду лицензиата; - Маркерная или меловая доска; - Проектор. 	<p>170001, Россия, город Тверь, улица Спартака, дом 26а</p>

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**



**Частное учреждение высшего образования
«Высшая школа предпринимательства (институт)»
(ЧУВО «ВШП»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
по дисциплине
Б1.В.08 «Разработка Web-приложений»**

**Направление подготовки: 09.03.02 Информационные системы и технологии
Направленность (профиль) программы бакалавриата
«Информационные системы в экономике»**

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ В РАМКАХ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций.

В результате освоения учебной дисциплины обучающийся должен демонстрировать следующие результаты обучения: УК-2, ПК-1, ПК-2, ПК-4.

Код компетенции	Наименование компетенции	Индекс и наименование индикатора содержания компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Способен определять круг задач в рамках поставленной цели	Знать: <ul style="list-style-type: none"> - Методы и техники формулирования задач. - Принципы декомпозиции задач на подзадачи. Уметь: <ul style="list-style-type: none"> - Определять цели и задачи проекта. - Делить задачи на этапы и определять приоритеты. Владеть: <ul style="list-style-type: none"> - Навыками структурирования и организации задач. - Навыками работы с целями и задачами проекта.
		УК-2.2 Способен выбирать оптимальные способы решения задач, исходя из правовых норм, ресурсов и ограничений	Знать: <ul style="list-style-type: none"> - Принципы и методы выбора оптимальных решений. - Правовые нормы и ограничения, влияющие на процесс разработки. Уметь: <ul style="list-style-type: none"> - Оценивать доступные ресурсы и ограничения проекта. - Выбирать и обосновывать оптимальные методы и инструменты для решения задач. Владеть: <ul style="list-style-type: none"> - Навыками анализа и оценки рисков при выборе решений. - Навыками работы с правовыми нормами и стандартами в области разработки программного обеспечения.
ПК-1	Разработка и отладка программного кода	ПК-1.1 Способен разрабатывать программный код	Знать: <ul style="list-style-type: none"> - Основные принципы и методы разработки программного кода. - Синтаксис и семантику используемых языков программирования (Python, C/C++, JS). Уметь: <ul style="list-style-type: none"> - Разрабатывать корректный и эффективный программный код. - Использовать стандартные библиотеки и фреймворки для разработки. Владеть: <ul style="list-style-type: none"> - Навыками написания и документирования кода. - Навыками применения передовых практик программирования (SOLID, DRY).

		ПК-1.2 Способен отлаживать программный код	<p>Знать:</p> <ul style="list-style-type: none"> - Основные методы и инструменты отладки программного кода. - Принципы и техники выявления и исправления ошибок. <p>Уметь:</p> <ul style="list-style-type: none"> - Использовать отладчики и инструменты профилирования для поиска ошибок. - Анализировать и исправлять ошибки в программном коде. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками эффективной отладки и тестирования кода. - Навыками применения различных техник отладки (пошаговая отладка, логирование).
ПК-2	Проверка работоспособности и рефакторинг кода программного обеспечения	ПК-2.1 Способен проверять работоспособность программного кода	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы тестирования программного кода. - Основы юнит-тестирования и функционального тестирования. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать и выполнять тестовые сценарии для проверки работоспособности кода. - Использовать тестовые фреймворки и инструменты для автоматизированного тестирования. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками написания и выполнения тестов. - Навыками анализа результатов тестирования и устранения обнаруженных проблем.
		ПК-2.2 Способен рефакторить программный код	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы рефакторинга программного кода. - Паттерны и антипаттерны проектирования кода. <p>Уметь:</p> <ul style="list-style-type: none"> - Проводить анализ и улучшение структуры кода. - Вносить изменения в код, сохраняя его функциональность. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками применения рефакторинга для улучшения читаемости и поддерживаемости кода. - Навыками использования инструментов для автоматизированного рефакторинга.
ПК-3	Интеграция программных модулей и компонентов и проверка работоспособности выпусков программного продукта	ПК-3.1 Способен интегрировать программные модули и компоненты	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы интеграции программных модулей. - Основы межмодульного взаимодействия и совместимости. <p>Уметь:</p> <ul style="list-style-type: none"> - Планировать и проводить интеграцию модулей в общий проект. - Решать проблемы, возникающие при интеграции различных компонентов.

			<p>Владеть:</p> <ul style="list-style-type: none"> - Навыками использования инструментов и платформ для интеграции. - Навыками написания интеграционных тестов.
		<p>ПК-3.2 Способен проверять работоспособность выпусков программного продукта</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Методы и инструменты для проверки выпусков программного продукта. - Основы тестирования и валидации программного обеспечения. <p>Уметь:</p> <ul style="list-style-type: none"> - Проводить проверку и валидацию программного продукта. - Анализировать результаты тестирования и устранять выявленные дефекты. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками работы с инструментами для проверки и тестирования выпусков. - Навыками документирования результатов тестирования и предложений по улучшению.
ПК-4	Разработка требований и проектирование программного обеспечения	<p>ПК-4.1 Способен разрабатывать требования к программному обеспечению</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Методы и техники сбора и анализа требований. - Принципы написания требований и спецификаций. <p>Уметь:</p> <ul style="list-style-type: none"> - Собирать и документировать требования к программному обеспечению. - Анализировать и уточнять требования с заинтересованными сторонами. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками написания четких и измеримых требований. - Навыками работы с инструментами для управления требованиями.
		<p>ПК-4.2 Способен проектировать программное обеспечение</p>	<p>Знать:</p> <ul style="list-style-type: none"> - Принципы и методы проектирования программного обеспечения. - Основы архитектурных шаблонов и стилей. <p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать архитектуру и дизайн программного обеспечения. - Создавать диаграммы и модели для визуализации архитектуры. <p>Владеть:</p> <ul style="list-style-type: none"> - Навыками применения архитектурных паттернов. - Навыками использования инструментов для проектирования и моделирования программного обеспечения.

КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

(признак, на основании которого, проводится оценка по выбранному показателю)

<i>Показатель оценивания компетенций</i>	<i>Результат обучения</i>	<i>Критерии оценивания компетенций</i>
Высокий уровень (отлично)	Знать	Обучающийся продемонстрировал: глубокие исчерпывающие знания и понимание учебного материала; содержательные, полные, правильные и конкретные ответы на все вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.
	Уметь	Обучающийся продемонстрировал: понимание учебного материала; умение свободно решать практические задания (ситуационные задачи), которые следует выполнить или описание результата, который нужно получить и др.; логически последовательные, содержательные, полные, правильные и конкретные ответы (решения) на все поставленные задания (вопросы), включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.
	Владеть	Обучающийся продемонстрировал: понимание учебного материала; умение свободно решать комплексные практические задания (решения задач по нестандартным ситуациям); логически последовательные, полные, правильные и конкретные ответы в ходе защиты задания, включая дополнительные уточняющие вопросы (задания); свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины.
Средний уровень (хорошо)	Знать	Обучающийся продемонстрировал: твердые и достаточно полные знания учебного материала; правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном устранении замечаний по отдельным вопросам; достаточное владение литературой, рекомендованной учебной программой дисциплины
	Уметь	Обучающийся продемонстрировал: понимание учебного материала; логически последовательные, правильные и конкретные ответы (решения) на основные задания (вопросы), включая дополнительные; устранение замечаний по отдельным элементам задания (вопроса); владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины
	Владеть	Обучающийся продемонстрировал: понимание учебного материала; продемонстрировал логически последовательные, достаточно полные, правильные ответы, включая дополнительные; самостоятельно устранил замечания по отдельным элементам задания (вопроса); владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины
Достаточный уровень (удовлетворительно)	Знать	Обучающийся продемонстрировал: твердые знания и понимание основного учебного материала; правильные, без грубых ошибок, ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины
	Уметь	Обучающийся продемонстрировал: понимание основного учебного материала; правильные, без грубых ошибок, ответы (решения) на основные задания (вопросы), включая дополнительные, устранение, при наводящих вопросах

		преподавателя, замечаний по отдельным элементам задания (вопроса); недостаточное полное владение литературой, рекомендованной учебной программой дисциплины
	Владеть	Обучающийся понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы при устранении неточностей и ошибок в решениях в ходе защиты задания (проекта, портфолио) при наводящих вопросах преподавателя; недостаточно полное владение литературой, рекомендованной учебной программой дисциплины

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

При проведении промежуточной аттестации в ЧУВО «ВШП» используются традиционные формы аттестации:

Форма промежуточной аттестации	Шкала оценивания
ЗАЧЕТ	«зачтено», «незачтено»
ЭКЗАМЕН	«отлично», «хорошо», «удовлетворительно», «неудовлетворительно»

КРИТЕРИИ И ПРОЦЕДУРЫ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ

Для оценивания результатов обучения в виде **ЗНАНИЙ** используются следующие процедуры и технологии:

- тестирование.

Для оценивания результатов обучения в виде **УМЕНИЙ и ВЛАДЕНИЙ** используются следующие процедуры и технологии:

- устный или письменный ответ на вопрос.
- практические задания, включающие одну или несколько задач (вопросов) в виде краткой формулировки действий (комплекса действий), которые следует выполнить, или описать результат, который нужно получить.

Критерии оценивания результата обучения по дисциплине (модулю)

Результат обучения по дисциплине (модулю)	ШКАЛА ОЦЕНИВАНИЯ				Процедуры оценивания
	«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»	
<u>УК-2</u> <u>ПК-1</u> <u>ПК-2</u> <u>ПК-4</u> Знать:	Обучаемый продемонстрировал: глубокие исчерпывающие знания и понимание учебного материала; содержательные, полные, правильные и конкретные ответы на все вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной	Обучаемый продемонстрировал: твердые и достаточно полные знания учебного материала; правильное понимание сущности и взаимосвязи рассматриваемых процессов и явлений; последовательные, правильные, конкретные ответы на поставленные вопросы при свободном	Обучаемый продемонстрировал: твердые знания и понимание основного учебного материала; правильные, без грубых ошибок, ответы на поставленные вопросы при устранении неточностей и несущественных ошибок в освещении отдельных положений при наводящих вопросах преподавателя; недостаточно	Обучаемый продемонстрировал: неправильные ответы на основные вопросы; грубые ошибки в ответах; непонимание сущности излагаемых вопросов; неуверенные и неточные ответы на дополнительные вопросы; не владеет основной литературой, рекомендованной учебной программой дисциплины.	Тестовые задания

	учебной программой дисциплины.	устранении замечаний по отдельным вопросам; достаточное владение литературой.	полное владение литературой, рекомендованной учебной программой дисциплины.		
<u>УК-2</u> , <u>ПК-1</u> , <u>ПК-2</u> , <u>ПК-4</u> Уметь:	Обучаемый продемонстрировал: понимание учебного материала, содержательные, полные, правильные и конкретные ответы на все поставленные вопросы, включая дополнительные; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание учебного материала; логически последовательные и конкретные ответы на основные задания/вопросы, включая дополнительные; устранение замечаний по отдельным элементам задания; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание основного учебного материала; правильные, без грубых ошибок, ответы на основные вопросы, включая дополнительные, при устранении, при наводящих вопросах преподавателя, замечаний по отдельным элементам задания; недостаточное владение литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: непонимание основного учебного материала; не дал правильные ответы на основные вопросы, включая дополнительные; не устранил, при наводящих вопросах преподавателя, замечания и грубые ошибки по вопросу; не владеет основной учебной литературой, рекомендованной учебной программой дисциплины	Вопросы Практические задания
<u>УК-2</u> , <u>ПК-1</u> , <u>ПК-2</u> , <u>ПК-4</u> Владеть:	Обучаемый продемонстрировал: понимание учебного материала; правильные и конкретные ответы, включая дополнительные и уточняющие вопросы; свободное владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание учебного материала; продемонстрировал логически последовательные, достаточно полные, верные ответы; самостоятельно устранил замечания по отдельным элементам; владение основной и дополнительной литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: понимание основного учебного материала; без грубых ошибок дал ответы на поставленные вопросы, в том числе при наводящих вопросах преподавателя; недостаточно владение литературой, рекомендованной учебной программой дисциплины	Обучаемый продемонстрировал: непонимание основного учебного материала; дал неправильные ответы на поставленные вопросы; не владеет основной учебной литературой, рекомендованной учебной программой дисциплины	Вопросы Практические задания

1. Оценочные материалы для самостоятельной работы обучающихся (студентов)

1.1 Реферат

7 семестр

Реферат позволит студентам углубиться в различные аспекты разработки Web-приложений, а также понять их важность и применение в современных информационных системах. Реферат покрывает компетенции УК-2.1, УК-2.2, ПК-1.1, ПК-1.2, ПК-4.1, ПК-4.2.

Примерная тематика рефератов:

- 1. История и эволюция Web-технологий**
 - Исследование развития веб-технологий, таких как HTML, CSS, JavaScript, и их влияние на рынок и пользователей.
- 2. Основы HTML и CSS в контексте разработки Web-приложений**
 - Анализ ключевых аспектов HTML и CSS и их применения в создании структурированных и стилизованных веб-страниц.
- 3. Проектирование пользовательского интерфейса в Web-приложениях**
 - Изучение принципов и лучших практик проектирования пользовательского интерфейса для веб-приложений.
- 4. Клиент-серверная архитектура Web-приложений**
 - Исследование клиент-серверной архитектуры, протоколов HTTP и HTTPS, и их роли в разработке веб-приложений.
- 5. Использование JavaScript для создания интерактивных Web-приложений**
 - Анализ возможностей JavaScript для создания динамических и интерактивных элементов на веб-страницах.
- 6. Node.js и серверная часть Web-приложений**
 - Обзор возможностей и применения Node.js и Express.js для разработки серверной части веб-приложений.

1.2 Реферат

8 семестр

Реферат позволит студентам углубиться в различные аспекты разработки Web-приложений, а также понять их важность и применение в современных информационных системах. Реферат покрывает компетенции УК-2.1, УК-2.2, ПК-1.1, ПК-1.2, ПК-4.1, ПК-4.2.

Примерная тематика рефератов:

- 1. Работа с базами данных в Web-приложениях**
 - Исследование различных методов хранения данных в веб-приложениях, включая реляционные и нереляционные базы данных.
- 2. Создание и использование REST API в Web-приложениях**
 - Изучение принципов разработки REST API и их применения для взаимодействия между клиентом и сервером.
- 3. Vue.js и его применение в разработке Web-приложений**
 - Анализ преимуществ использования Vue.js для создания динамических веб-приложений.
- 4. Обеспечение безопасности Web-приложений**
 - Обзор методов защиты веб-приложений от XSS, CSRF и SQL-инъекций, а также практики аутентификации и авторизации.
- 5. Оптимизация и производительность Web-приложений**
 - Исследование методов оптимизации производительности веб-приложений и инструментов для анализа производительности.
- 6. Тестирование и отладка Web-приложений**

- Обзор методов тестирования и отладки веб-приложений, включая юнит-тестирование и интеграционные тесты.
7. **Процесс развертывания и публикации Web-приложений**
- Исследование этапов подготовки и развертывания веб-приложений на различных хостингах и платформах.

Цель написания рефератов: Углубить понимание и критическое осмысление различных аспектов разработки Web-приложений, развивая аналитические и научные навыки студентов.

Структура реферата:

1. **Введение**
 - Краткое описание темы и целей реферата.
 - Актуальность темы.
2. **Основная часть**
 - Теоретические основы темы.
 - История и эволюция (если применимо).
 - Применение в современной разработке веб-приложений.
 - Примеры и кейсы.
 - Проблемы и вызовы.
 - Перспективы и тенденции развития.
3. **Заключение**
 - Выводы по результатам исследования.
 - Значение изученных аспектов для разработки веб-приложений.
4. **Список использованных источников**
 - Перечень использованной литературы и интернет-ресурсов.

Критерии оценивания:

1. **Структура и логика изложения (20%)**
 - Четкая структура работы (введение, основная часть, заключение).
 - Логичность и последовательность изложения материала.
2. **Содержание (40%)**
 - Полнота раскрытия темы.
 - Описание основных этапов развития разработки веб-приложений.
 - Анализ современных тенденций.
 - Примеры применения веб-разработки в информационных системах.
3. **Аналитическая часть (20%)**
 - Глубина анализа роли разработки веб-приложений в информационных системах.
 - Наличие собственных выводов и оценок.
4. **Оформление (10%)**
 - Соответствие требованиям к оформлению рефератов (шрифт, отступы, заголовки и т.д.).
 - Корректное оформление ссылок и списка литературы.
5. **Язык и стиль (10%)**
 - Грамотность и точность изложения.
 - Научный стиль текста.

Требования к объему:

Объем реферата должен составлять 10-15 страниц печатного текста (шрифт Times New Roman, размер 12, интервал 1.5, поля 2 см со всех сторон).

2. Оценочные материалы для оценки текущей аттестации обучающихся (студентов)

2.1 Тестовые задания для текущего контроля успеваемости в виде ЗНАНИЙ

В тестовом задании вопросы, которые имеют закрытый характер.

Правильные ответы выделены знаком +.

7 семестр

1. Что представляет собой HTML? (УК-2.1)
 - Язык программирования.
 - Язык разметки для создания веб-страниц. +
 - Стандарт сетевого протокола.
 - Графический редактор.
2. Какой тег используется для создания гиперссылки в HTML? (ПК-4.1)
 - <link>
 - <a> +
 - <href>
 - <hyperlink>
3. Какое из следующих утверждений правильно описывает CSS? (УК-2.2)
 - Язык программирования.
 - Язык разметки.
 - Язык для описания стилей элементов веб-страницы. +
 - Язык запросов к базе данных.
4. Какой из следующих селекторов CSS выбирает элемент по его идентификатору? (ПК-4.2)
 - .class
 - #id +
 - element
 - *
5. Какой метод HTTP используется для отправки данных на сервер? (ПК-1.1)
 - GET
 - POST +
 - PUT
 - DELETE
6. Что такое DOM в контексте JavaScript? (УК-2.1, ПК-2.1)
 - Система управления базами данных.
 - Объектная модель документа, представляющая структуру HTML или XML документа. +
 - Система обработки запросов на сервере.
 - Язык запросов к базе данных.
7. Какой метод JavaScript используется для добавления элемента в конец массива? (ПК-1.2)
 - push() +
 - pop()
 - shift()
 - unshift()
8. Что делает функция JSON.stringify() в JavaScript? (ПК-2.2)
 - Преобразует объект JavaScript в строку JSON. +
 - Преобразует строку JSON в объект JavaScript.
 - Сжимает данные для отправки на сервер.
 - Распаковывает данные, полученные с сервера.
9. Что такое Node.js? (ПК-3.1)
 - Фреймворк для создания клиентских приложений.
 - Среда выполнения JavaScript вне браузера. +
 - Библиотека для работы с графикой.
 - Приложение для создания анимаций.

10. Какой модуль в Node.js используется для создания HTTP-сервера? (ПК-3.2)
 - fs
 - http +
 - url
 - path
11. Какой из следующих запросов используется для получения данных с сервера? (УК-2.2)
 - POST
 - GET +
 - PUT
 - DELETE
12. Что из перечисленного является основным компонентом Vue.js? (ПК-4.2)
 - Directive
 - Component +
 - Module
 - Service
13. Какой из методов Vue.js используется для создания реактивных данных? (ПК-1.1)
 - data()
 - methods
 - computed
 - ref() +
14. Какое из следующих утверждений правильно описывает REST API? (ПК-4.1)
 - Протокол передачи файлов.
 - Архитектурный стиль для создания веб-сервисов. +
 - Язык разметки для создания веб-страниц.
 - Библиотека для работы с графикой.
15. Какие меры могут быть приняты для защиты веб-приложений от XSS-атак? (УК-2.1, ПК-2.1)
 - Использование HTTPS.
 - Валидация и экранирование ввода пользователя. +
 - Шифрование данных.
 - Использование сложных паролей.
16. Что такое CORS? (УК-2.2)
 - Протокол передачи данных.
 - Механизм для разрешения запросов между различными источниками. +
 - Язык программирования.
 - Система управления базами данных.
17. Какой атрибут HTML используется для задания пути к внешнему файлу CSS? (ПК-4.1)
 - src
 - href +
 - link
 - style
18. Что такое middleware в Express.js? (ПК-3.1, ПК-3.2)
 - Компонент для управления базами данных.
 - Функция, обрабатывающая запросы и ответы в приложении. +
 - Библиотека для работы с графикой.
 - Приложение для тестирования.
19. Какой метод JavaScript используется для удаления последнего элемента из массива? (ПК-1.2)
 - pop() +
 - push()
 - shift()
 - unshift()

20. Что такое Vuex в контексте Vue.js? (ПК-4.2)
- Модуль для работы с формами.
 - Хранилище состояния для управления данными в приложении. +
 - Библиотека для анимации.
 - Плагин для интеграции с базами данных.
21. Какой метод Vue.js используется для отслеживания изменений данных? (ПК-1.1, ПК-2.1)
- watch() +
 - computed
 - methods
 - data()
22. Какое из следующих утверждений правильно описывает принцип работы AJAX? (УК-2.2, ПК-1.1)
- Метод асинхронного взаимодействия клиента с сервером. +
 - Протокол передачи файлов.
 - Средство для создания анимаций.
 - Язык программирования.
23. Какой метод используется для удаления свойства объекта в JavaScript? (ПК-1.2, ПК-2.2)
- delete +
 - remove
 - unset
 - clear
24. Какие из перечисленных мер могут улучшить производительность веб-приложений? (УК-2.1, ПК-3.2)
- Минимизация количества запросов.
 - Использование кеширования.
 - Оптимизация изображений.
 - Все перечисленные меры. +

Критерии оценки результатов теста

- 1. "Неудовлетворительно" (0-39%)**
 - Студент ответил правильно на менее 40% вопросов.
 - Значительные пробелы в знаниях по большинству тем.
 - Неправильное понимание ключевых понятий и принципов.
 - Неспособность применить теоретические знания на практике.
- 2. "Удовлетворительно" (40-59%)**
 - Студент ответил правильно на 40-59% вопросов.
 - Основные понятия и принципы поняты частично, есть ошибки в ответах.
 - Знания по большинству тем на базовом уровне, недостаточная глубина понимания.
 - Частичная способность применять теоретические знания на практике, нужны дополнительные разъяснения.
- 3. "Хорошо" (60-79%)**
 - Студент ответил правильно на 60-79% вопросов.
 - Хорошее понимание ключевых понятий и принципов, незначительные ошибки.
 - Знания по всем темам на достаточном уровне, однако есть некоторые пробелы.
 - Способность применять теоретические знания на практике, но требуется улучшение точности и уверенности.
- 4. "Отлично" (80-100%)**
 - Студент ответил правильно на 80-100% вопросов.
 - Полное и правильное понимание всех ключевых понятий и принципов.
 - Глубокие знания по всем темам, минимальные или отсутствующие ошибки.
 - Высокий уровень способности применять теоретические знания на практике, демонстрация уверенности и точности в ответах.

2.2 Тестовые задания для текущего контроля успеваемости в виде ЗНАНИЙ

В тестовом задании вопросы, которые имеют закрытый характер.

Правильные ответы выделены знаком +.

8 семестр

1. Что такое ECMAScript? (УК-2.1)
 - Версия Java.
 - Спецификация стандарта для языка JavaScript. +
 - Протокол передачи данных.
 - Библиотека для работы с графикой.
2. Какой оператор в JavaScript используется для создания промиса? (ПК-1.1)
 - `new Promise` +
 - `new Await`
 - `new Async`
 - `new Defer`
3. Что такое Vue Router? (ПК-4.2)
 - Хранилище данных.
 - Маршрутизатор для создания одностраничных приложений на Vue.js. +
 - Библиотека для работы с API.
 - Инструмент для тестирования.
4. Какое из следующих утверждений правильно описывает компонент Vue.js? (ПК-1.1, ПК-2.1)
 - Компонент Vue.js нельзя переиспользовать.
 - Компонент Vue.js представляет собой переиспользуемый блок кода, который инкапсулирует HTML, CSS и JavaScript. +
 - Компонент Vue.js используется только для стилей.
 - Компонент Vue.js используется только для данных.
5. Какое свойство CSS отвечает за прозрачность элемента? (УК-2.2)
 - `visibility`
 - `opacity` +
 - `display`
 - `transparency`
6. Что такое WebSocket? (ПК-3.1)
 - Протокол для одностороннего обмена данными.
 - Протокол для двустороннего обмена данными в режиме реального времени. +
 - Метод асинхронного запроса.
 - Язык разметки.
7. Какой метод JavaScript используется для создания асинхронных функций? (ПК-2.2)
 - `async` +
 - `await`
 - `defer`
 - `promise`
8. Что такое unit-тестирование? (ПК-2.1, ПК-2.2)
 - Тестирование всей системы в целом.
 - Тестирование отдельных модулей или компонентов кода. +
 - Тестирование пользовательского интерфейса.
 - Тестирование базы данных.
9. Какой инструмент используется для автоматического тестирования Vue.js компонентов? (ПК-4.1)
 - Jest +

- Mocha
 - Jasmine
 - Chai
10. Что такое CI/CD? (ПК-3.2)
- Инструменты для работы с базами данных.
 - Методология, включающая непрерывную интеграцию и непрерывное развертывание. +
 - Протокол передачи данных.
 - Средство для управления версиями.
11. Какой тег HTML используется для внедрения JavaScript кода в страницу? (УК-2.2)
- `<style>`
 - `<link>`
 - `<script>` +
 - `<meta>`
12. Что из перечисленного является примером асинхронного взаимодействия в JavaScript? (ПК-2.2)
- `setTimeout()` +
 - `JSON.parse()`
 - `Math.random()`
 - `Date.now()`
13. Какой метод JavaScript используется для удаления первого элемента из массива? (ПК-1.2)
- `shift()` +
 - `unshift()`
 - `pop()`
 - `push()`
14. Какое свойство CSS используется для изменения цвета текста? (УК-2.2)
- `font-color`
 - `color` +
 - `text-color`
 - `background-color`
15. Что такое JSON? (ПК-4.1)
- Язык программирования.
 - Формат обмена данными, основанный на JavaScript. +
 - Протокол передачи файлов.
 - Библиотека для работы с графикой.
16. Какой инструмент используется для проверки производительности веб-страниц? (ПК-3.2)
- Lighthouse +
 - Vuex
 - Express
 - Webpack
17. Что такое E2E-тестирование? (ПК-2.1, ПК-2.2)
- Тестирование отдельных модулей.
 - Тестирование всей системы от начала до конца. +
 - Тестирование только серверной части.
 - Тестирование пользовательского интерфейса.
18. Какой метод JavaScript используется для создания нового массива из всех элементов, которые прошли проверку, реализованную в предоставленной функции? (ПК-1.2)
- `filter()` +
 - `map()`

- reduce()
 - forEach()
19. Что такое пакетный менеджер npm? (ПК-4.1)
- Приложение для управления базами данных.
 - Пакетный менеджер для JavaScript, позволяющий управлять зависимостями проекта. +
 - Библиотека для создания графиков.
 - Фреймворк для тестирования.
20. Какое из следующих утверждений правильно описывает async/await? (ПК-2.2)
- Конструкции для работы с синхронным кодом.
 - Конструкции для упрощения работы с промисами и асинхронным кодом. +
 - Метод работы с XML.
 - Функции для управления DOM.

Критерии оценки результатов теста

5. **"Неудовлетворительно" (0-39%)**
- Студент ответил правильно на менее 40% вопросов.
 - Значительные пробелы в знаниях по большинству тем.
 - Неправильное понимание ключевых понятий и принципов.
 - Неспособность применить теоретические знания на практике.
6. **"Удовлетворительно" (40-59%)**
- Студент ответил правильно на 40-59% вопросов.
 - Основные понятия и принципы поняты частично, есть ошибки в ответах.
 - Знания по большинству тем на базовом уровне, недостаточная глубина понимания.
 - Частичная способность применять теоретические знания на практике, нужны дополнительные разъяснения.
7. **"Хорошо" (60-79%)**
- Студент ответил правильно на 60-79% вопросов.
 - Хорошее понимание ключевых понятий и принципов, незначительные ошибки.
 - Знания по всем темам на достаточном уровне, однако есть некоторые пробелы.
 - Способность применять теоретические знания на практике, но требуется улучшение точности и уверенности.
8. **"Отлично" (80-100%)**
- Студент ответил правильно на 80-100% вопросов.
 - Полное и правильное понимание всех ключевых понятий и принципов.
 - Глубокие знания по всем темам, минимальные или отсутствующие ошибки.
 - Высокий уровень способности применять теоретические знания на практике, демонстрация уверенности и точности в ответах.

2.3 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ

7 семестр

1. **Что представляет собой HTML? (УК-2.1)**
- Правильный ответ: HTML (HyperText Markup Language) - это язык разметки, используемый для создания и структурирования веб-страниц. Он определяет структуру веб-документов с помощью различных тегов и атрибутов.
2. **Какой тег используется для создания гиперссылки в HTML? (ПК-4.1)**
- Правильный ответ: Тег <a> используется для создания гиперссылок в HTML. Он позволяет ссылаться на другие веб-страницы или ресурсы, используя атрибут href.
3. **Какое из следующих утверждений правильно описывает CSS? (УК-2.2)**
- Правильный ответ: CSS (Cascading Style Sheets) - это язык, используемый для описания внешнего вида и форматирования HTML-документов. Он позволяет

- управлять стилями элементов, такими как цвета, шрифты и расположение на странице.
4. **Какой из следующих селекторов CSS выбирает элемент по его идентификатору? (ПК-4.2)**
 - Правильный ответ: Селектор #id используется для выбора элемента по его идентификатору в CSS. Идентификатор задается атрибутом id в HTML.
 5. **Какой метод HTTP используется для отправки данных на сервер? (ПК-1.1)**
 - Правильный ответ: Метод POST используется для отправки данных на сервер в HTTP. Он обычно используется при отправке форм или данных, требующих обработки на сервере.
 6. **Что такое DOM в контексте JavaScript? (УК-2.1, ПК-2.1)**
 - Правильный ответ: DOM (Document Object Model) - это объектная модель документа, представляющая структуру HTML или XML-документа в виде дерева объектов. DOM позволяет JavaScript взаимодействовать с содержимым и структурой веб-страницы.
 7. **Какой метод JavaScript используется для добавления элемента в конец массива? (ПК-1.2)**
 - Правильный ответ: Метод push() используется для добавления одного или нескольких элементов в конец массива в JavaScript.
 8. **Что делает функция JSON.stringify() в JavaScript? (ПК-2.2)**
 - Правильный ответ: Функция JSON.stringify() преобразует объект JavaScript в строку JSON, что позволяет передавать данные между клиентом и сервером в формате JSON.
 9. **Что такое Node.js? (ПК-3.1)**
 - Правильный ответ: Node.js - это среда выполнения JavaScript вне браузера, основанная на движке V8. Она позволяет создавать серверные приложения с использованием JavaScript.
 10. **Какой модуль в Node.js используется для создания HTTP-сервера? (ПК-3.2)**
 - Правильный ответ: Модуль http используется для создания HTTP-сервера в Node.js. Он предоставляет функциональность для обработки HTTP-запросов и ответов.
 11. **Какой из следующих запросов используется для получения данных с сервера? (УК-2.2)**
 - Правильный ответ: Метод GET используется для получения данных с сервера в HTTP. Он обычно используется для запросов, которые не изменяют состояние сервера.
 12. **Что из перечисленного является основным компонентом Vue.js? (ПК-4.2)**
 - Правильный ответ: Основным компонентом Vue.js является компонент (Component). Компоненты в Vue.js инкапсулируют HTML, CSS и JavaScript, обеспечивая модульность и переиспользуемость кода.
 13. **Какой из методов Vue.js используется для создания реактивных данных? (ПК-1.1)**
 - Правильный ответ: Метод ref() используется для создания реактивных данных в Vue.js. Он позволяет отслеживать изменения данных и автоматически обновлять интерфейс.
 14. **Какое из следующих утверждений правильно описывает REST API? (ПК-4.1)**
 - Правильный ответ: REST API (Representational State Transfer Application Programming Interface) - это архитектурный стиль для создания веб-сервисов, который использует стандартные HTTP-методы для взаимодействия с ресурсами.
 15. **Какие меры могут быть приняты для защиты веб-приложений от XSS-атак? (УК-2.1, ПК-2.1)**
 - Правильный ответ: Валидация и экранирование ввода пользователя являются основными мерами защиты веб-приложений от XSS-атак. Они предотвращают внедрение вредоносного кода в веб-страницы.

16. **Что такое CORS? (УК-2.2)**
- Правильный ответ: CORS (Cross-Origin Resource Sharing) - это механизм, который позволяет веб-сайтам обращаться к ресурсам на других доменах, соблюдая определенные правила безопасности.
17. **Какой атрибут HTML используется для задания пути к внешнему файлу CSS? (ПК-4.1)**
- Правильный ответ: Атрибут href используется в теге <link> для задания пути к внешнему файлу CSS.
18. **Что такое middleware в Express.js? (ПК-3.1, ПК-3.2)**
- Правильный ответ: Middleware в Express.js - это функция, которая обрабатывает запросы и ответы в приложении, выполняя различные задачи, такие как аутентификация, логирование и обработка ошибок.
19. **Какой метод JavaScript используется для удаления последнего элемента из массива? (ПК-1.2)**
- Правильный ответ: Метод pop() используется для удаления последнего элемента из массива в JavaScript.
20. **Что такое Vuex в контексте Vue.js? (ПК-4.2)**
- Правильный ответ: Vuex - это хранилище состояния для управления данными в приложении на Vue.js. Оно обеспечивает централизованное управление состоянием и упрощает работу с данными в крупных приложениях.
21. **Какой метод Vue.js используется для отслеживания изменений данных? (ПК-1.1, ПК-2.1)**
- Правильный ответ: Метод watch() используется в Vue.js для отслеживания изменений данных и выполнения определенных действий при их изменении.
22. **Какое из следующих утверждений правильно описывает принцип работы AJAX? (УК-2.2, ПК-1.1)**
- Правильный ответ: AJAX (Asynchronous JavaScript and XML) - это метод асинхронного взаимодействия клиента с сервером, позволяющий обновлять содержимое веб-страницы без перезагрузки.
23. **Какой метод используется для удаления свойства объекта в JavaScript? (ПК-1.2, ПК-2.2)**
- Правильный ответ: Оператор delete используется для удаления свойства объекта в JavaScript.
24. **Какие из перечисленных мер могут улучшить производительность веб-приложений? (УК-2.1, ПК-3.2)**
- Правильный ответ: Минимизация количества запросов, использование кеширования и оптимизация изображений - все эти меры могут улучшить производительность веб-приложений.

Критерии оценки ответов на вопросы

- **"Отлично" (5 баллов)**
 - **Критерии:**
 - Полное и точное объяснение вопроса.
 - Ответ включает все ключевые аспекты и детали.
 - Примеры, если требуются, приведены и правильно объяснены.
 - Ответ демонстрирует глубокое понимание темы.
- **"Хорошо" (4 балла)**
 - **Критерии:**
 - Корректное объяснение вопроса.
 - Ответ охватывает основные аспекты, но может отсутствовать незначительная деталь или пример.
 - Демонстрируется хорошее, но не полное понимание темы.

- **"Удовлетворительно" (3 балла)**
 - **Критерии:**
 - Общее представление о вопросе.
 - Ответ включает основные аспекты, но содержит неточности или пропуски.
 - Примеры, если требуются, могут отсутствовать или быть неверно объяснены.
 - Демонстрируется базовое понимание темы.
- **"Неудовлетворительно" (2 балла)**
 - **Критерии:**
 - Некорректное или неполное объяснение вопроса.
 - Отсутствие ключевых аспектов и деталей.
 - Примеры, если требуются, отсутствуют или приведены неверные.
 - Ответ демонстрирует недостаточное понимание темы.

2.4 Вопросы для текущего контроля успеваемости в виде УМЕНИЙ 8 семестр

1. **Что такое ECMAScript? (УК-2.1)**
 - Правильный ответ: ECMAScript - это спецификация стандарта для языка JavaScript, определяющая его синтаксис и поведение.
2. **Какой оператор в JavaScript используется для создания промиса? (ПК-1.1)**
 - Правильный ответ: Оператор `new Promise` используется для создания нового промиса в JavaScript.
3. **Что такое Vue Router? (ПК-4.2)**
 - Правильный ответ: Vue Router - это маршрутизатор для создания одностраничных приложений на Vue.js, позволяющий управлять навигацией между различными компонентами.
4. **Какое из следующих утверждений правильно описывает компонент Vue.js? (ПК-1.1, ПК-2.1)**
 - Правильный ответ: Компонент Vue.js представляет собой переиспользуемый блок кода, который инкапсулирует HTML, CSS и JavaScript, обеспечивая модульность и переиспользуемость.
5. **Какое свойство CSS отвечает за прозрачность элемента? (УК-2.2)**
 - Правильный ответ: Свойство `opacity` в CSS отвечает за прозрачность элемента, задавая уровень его непрозрачности.
6. **Что такое WebSocket? (ПК-3.1)**
 - Правильный ответ: WebSocket - это протокол для двустороннего обмена данными в режиме реального времени между клиентом и сервером.
7. **Какой метод JavaScript используется для создания асинхронных функций? (ПК-2.2)**
 - Правильный ответ: Оператор `async` используется для создания асинхронных функций в JavaScript.
8. **Что такое unit-тестирование? (ПК-2.1, ПК-2.2)**
 - Правильный ответ: Unit-тестирование - это тестирование отдельных модулей или компонентов кода для проверки их правильной работы.
9. **Какой инструмент используется для автоматического тестирования Vue.js компонентов? (ПК-4.1)**
 - Правильный ответ: Jest - это инструмент для автоматического тестирования Vue.js компонентов, позволяющий писать и выполнять тесты.
10. **Что такое CI/CD? (ПК-3.2)**
 - Правильный ответ: CI/CD (Continuous Integration/Continuous Deployment) - это методология, включающая непрерывную интеграцию и непрерывное развертывание, обеспечивающая автоматизацию разработки и доставки программного обеспечения.

11. **Какой тег HTML используется для внедрения JavaScript кода в страницу? (УК-2.2)**
 - Правильный ответ: Тег `<script>` используется для внедрения JavaScript кода в HTML-страницу.
12. **Что из перечисленного является примером асинхронного взаимодействия в JavaScript? (ПК-2.2)**
 - Правильный ответ: Метод `setTimeout()` является примером асинхронного взаимодействия в JavaScript, позволяя выполнять код через определенный промежуток времени.
13. **Какой метод JavaScript используется для удаления первого элемента из массива? (ПК-1.2)**
 - Правильный ответ: Метод `shift()` используется для удаления первого элемента из массива в JavaScript.
14. **Какое свойство CSS используется для изменения цвета текста? (УК-2.2)**
 - Правильный ответ: Свойство `color` в CSS используется для изменения цвета текста.
15. **Что такое JSON? (ПК-4.1)**
 - Правильный ответ: JSON (JavaScript Object Notation) - это формат обмена данными, основанный на JavaScript, используемый для передачи структурированных данных.
16. **Какой инструмент используется для проверки производительности веб-страниц? (ПК-3.2)**
 - Правильный ответ: Lighthouse - это инструмент для проверки производительности веб-страниц, предоставляющий отчеты о производительности, доступности и SEO.
17. **Что такое E2E-тестирование? (ПК-2.1, ПК-2.2)**
 - Правильный ответ: E2E (End-to-End) тестирование - это тестирование всей системы от начала до конца, включая взаимодействие всех компонентов.
18. **Какой метод JavaScript используется для создания нового массива из всех элементов, которые прошли проверку, реализованную в предоставленной функции? (ПК-1.2)**
 - Правильный ответ: Метод `filter()` используется для создания нового массива из всех элементов, которые прошли проверку, реализованную в предоставленной функции.
19. **Что такое пакетный менеджер npm? (ПК-4.1)**
 - Правильный ответ: npm (Node Package Manager) - это пакетный менеджер для JavaScript, позволяющий управлять зависимостями проекта.
20. **Какое из следующих утверждений правильно описывает `async/await`? (ПК-2.2)**
 - Правильный ответ: Конструкции `async/await` упрощают работу с промисами и асинхронным кодом в JavaScript, делая его более читаемым и управляемым.

Критерии оценки ответов на вопросы

- **"Отлично" (5 баллов)**
 - **Критерии:**
 - Полное и точное объяснение вопроса.
 - Ответ включает все ключевые аспекты и детали.
 - Примеры, если требуются, приведены и правильно объяснены.
 - Ответ демонстрирует глубокое понимание темы.
- **"Хорошо" (4 балла)**
 - **Критерии:**
 - Корректное объяснение вопроса.
 - Ответ охватывает основные аспекты, но может отсутствовать незначительная деталь или пример.
 - Демонстрируется хорошее, но не полное понимание темы.
- **"Удовлетворительно" (3 балла)**
 - **Критерии:**
 - Общее представление о вопросе.
 - Ответ включает основные аспекты, но содержит неточности или пропуски.

- Примеры, если требуются, могут отсутствовать или быть неверно объяснены.
- Демонстрируется базовое понимание темы.
- **"Неудовлетворительно" (2 балла)**
 - **Критерии:**
 - Некорректное или неполное объяснение вопроса.
 - Отсутствие ключевых аспектов и деталей.
 - Примеры, если требуются, отсутствуют или приведены неверные.
 - Ответ демонстрирует недостаточное понимание темы.

2.5 Задачи на соответствие понятий для текущего контроля успеваемости в виде ВЛАДЕНИЙ

Правильные ответы расположены в таблицах друг напротив друга, во время тестирования предполагается что порядок данных в рамках каждого столбца будет случайным.

Задача 1. Соотнесите основные понятия HTML с их определениями (УК-2.1, ПК-4.1)

Понятие	Определение
A - Тег	1 - Основной строительный блок HTML-документа, указывающий, как элементы должны отображаться.
B - Атрибут	2 - Информация, добавляемая к тегу для уточнения его свойств.
C - Элемент	3 - Объединение тега и его содержимого.
D - Документ	4 - Полная структура HTML, содержащая все элементы и теги.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 2. Соотнесите основные понятия CSS с их определениями (УК-2.2, ПК-4.2)

Понятие	Определение
A - Селектор	1 - Часть CSS-правила, которая указывает, к каким элементам HTML применяются стили.
B - Свойство	2 - Характеристика, которая задается для элемента, например, цвет или шрифт.
C - Значение	3 - Конкретное значение, присваиваемое свойству, например, red для цвета.
D - Правило	4 - Комбинация селектора, свойства и значения.

Правильный ответ: A-1, B-2, C-3, D-4

Задача 3. Соотнесите основные понятия JavaScript с их определениями (ПК-1.1, ПК-2.2)

Понятие	Определение
A - Переменная	1 - Место в памяти, где хранится значение, которое может изменяться.

В - Функция	2 - Набор инструкций, который выполняет определенную задачу и может быть вызван.
С - Объект	3 - Коллекция свойств и методов, представляющая конкретную сущность.
D - Массив	4 - Упорядоченная коллекция элементов, которые могут быть доступны по индексу.

Правильный ответ: А-1, В-2, С-3, D-4

Задача 4. Соотнесите основные понятия Node.js с их определениями (ПК-3.1, ПК-3.2)

Понятие	Определение
А - Модуль	1 - Файл или набор файлов, которые можно импортировать в другие части программы.
В - Сервер	2 - Программа, которая отвечает на запросы клиентов и предоставляет ресурсы или услуги.
С - Middleware	3 - Функция, обрабатывающая запросы и ответы в приложении.
D - Callback	4 - Функция, переданная в другую функцию как аргумент и вызываемая после завершения задачи.

Правильный ответ: А-1, В-2, С-3, D-4

Задача 5. Соотнесите основные понятия REST API с их определениями (ПК-4.1, ПК-4.2)

Понятие	Определение
А - GET	1 - HTTP метод для получения данных с сервера.
В - POST	2 - HTTP метод для отправки данных на сервер.
С - PUT	3 - HTTP метод для обновления существующих данных на сервере.
D - DELETE	4 - HTTP метод для удаления данных с сервера.

Правильный ответ: А-1, В-2, С-3, D-4

Задача 6. Соотнесите основные понятия Vue.js с их определениями (ПК-1.1, ПК-4.2)

Понятие	Определение
А - Компонент	1 - Переиспользуемый блок кода, инкапсулирующий HTML, CSS и JavaScript.
В - Директива	2 - Специальный атрибут, который изменяет поведение элемента.
С - Data	3 - Объект, содержащий реактивные данные компонента.
D - Method	4 - Функция, определенная в компоненте, выполняющая действия.

Правильный ответ: А-1, В-2, С-3, D-4

Критерии оценки выполнения задач на соответствие понятий

- **Правильность соответствий:**
 - **Отлично (5):** Все соответствия выполнены правильно.
 - **Хорошо (4):** 1 ошибка в соответствиях.
 - **Удовлетворительно (3):** 2 ошибки в соответствиях.
 - **Неудовлетворительно (2):** 3 и более ошибок в соответствиях.

3. Оценочные материалы для проведения промежуточной аттестации обучающихся (студентов)

3.1 Вопросы для проведения промежуточной аттестации в форме ЗАЧЕТА С ОЦЕНКОЙ 7 семестр

1. **Что представляет собой HTML и для чего он используется? (УК-2.1)**
 - **Правильный ответ:** HTML (HyperText Markup Language) - это язык разметки, используемый для создания и структурирования веб-страниц. Он определяет структуру веб-документов с помощью различных тегов и атрибутов. HTML позволяет добавлять текст, изображения, ссылки, таблицы и другие элементы на веб-страницы, создавая основу для веб-документа.
2. **Какой тег используется для создания гиперссылки в HTML и как он работает? (ПК-4.1)**
 - **Правильный ответ:** Тег `<a>` используется для создания гиперссылок в HTML. Он позволяет ссылаться на другие веб-страницы или ресурсы, используя атрибут `href`, который указывает URL целевого ресурса. Например, `Example`. При клике на такую ссылку пользователь будет перенаправлен на указанный ресурс.
3. **Какое из следующих утверждений правильно описывает CSS и его применение? (УК-2.2)**
 - **Правильный ответ:** CSS (Cascading Style Sheets) - это язык, используемый для описания внешнего вида и форматирования HTML-документов. Он позволяет управлять стилями элементов, такими как цвета, шрифты, отступы, границы и расположение на странице. CSS отделяет структуру документа (HTML) от его представления, обеспечивая гибкость и возможности для более сложного и привлекательного дизайна.
4. **Какой из следующих селекторов CSS выбирает элемент по его идентификатору и как это применяется? (ПК-4.2)**
 - **Правильный ответ:** Селектор `#id` используется для выбора элемента по его идентификатору в CSS. Идентификатор задается атрибутом `id` в HTML. Например, чтобы выбрать элемент с `id="header"`, используется CSS-селектор `#header { ... }`. Это позволяет точно нацеливать стили на конкретные элементы.
5. **Какой метод HTTP используется для отправки данных на сервер и в каких случаях его применяют? (ПК-1.1)**
 - **Правильный ответ:** Метод POST используется для отправки данных на сервер в HTTP. Он обычно применяется при отправке форм или данных, требующих обработки на сервере. Например, POST-запрос может быть использован для отправки данных регистрации или авторизации пользователей, добавления новых записей в базу данных и т.д.
6. **Что такое DOM в контексте JavaScript и как он используется для взаимодействия с веб-страницей? (УК-2.1, ПК-2.1)**

- **Правильный ответ:** DOM (Document Object Model) - это объектная модель документа, представляющая структуру HTML или XML-документа в виде дерева объектов. DOM позволяет JavaScript взаимодействовать с содержимым и структурой веб-страницы, изменяя элементы, атрибуты, стили и события. Например, с помощью DOM можно добавлять новые элементы на страницу, изменять текст существующих элементов или обрабатывать пользовательские действия, такие как клики и ввод данных.
7. **Какой метод JavaScript используется для добавления элемента в конец массива и как он работает? (ПК-1.2)**
- **Правильный ответ:** Метод push() используется для добавления одного или нескольких элементов в конец массива в JavaScript. Например, если у нас есть массив чисел, этот метод добавит новый элемент в конец этого массива, увеличивая его длину.
8. **Что делает функция JSON.stringify() в JavaScript и в каких случаях она полезна? (ПК-2.2)**
- **Правильный ответ:** Функция JSON.stringify() преобразует объект JavaScript в строку JSON. Это полезно для передачи данных между клиентом и сервером в формате JSON. Например, перед отправкой объекта через AJAX-запрос, его можно преобразовать в строку JSON, чтобы сервер мог его правильно обработать.
9. **Что такое Node.js и какие возможности оно предоставляет для разработки серверных приложений? (ПК-3.1)**
- **Правильный ответ:** Node.js - это среда выполнения JavaScript вне браузера, основанная на движке V8. Она позволяет создавать серверные приложения с использованием JavaScript. Node.js предоставляет множество модулей для работы с HTTP, файловой системой, базами данных и другими системными ресурсами, что делает его мощным инструментом для разработки серверных приложений и веб-сервисов.
10. **Какой модуль в Node.js используется для создания HTTP-сервера и как это делается? (ПК-3.2)**
- **Правильный ответ:** Модуль http используется для создания HTTP-сервера в Node.js. Для создания сервера необходимо импортировать этот модуль и использовать его методы для обработки запросов и отправки ответов. Например, этот модуль позволяет создавать сервер, который будет слушать запросы на определенном порту и отправлять ответы клиентам.
11. **Какой из следующих запросов используется для получения данных с сервера и в каких ситуациях его применяют? (УК-2.2)**
- **Правильный ответ:** Метод GET используется для получения данных с сервера в HTTP. Он обычно применяется для запросов, которые не изменяют состояние сервера, таких как получение информации о пользователе, загрузка веб-страниц или получение данных из базы данных. GET-запросы могут кэшироваться и индексироваться поисковыми системами.
12. **Что из перечисленного является основным компонентом Vue.js и как он используется? (ПК-4.2)**
- **Правильный ответ:** Основным компонентом Vue.js является компонент (Component). Компоненты в Vue.js инкапсулируют HTML, CSS и JavaScript, обеспечивая модульность и переиспользуемость кода. Компоненты можно создавать и использовать для создания сложных интерфейсов из отдельных, независимых блоков.
13. **Какой из методов Vue.js используется для создания реактивных данных и как он работает? (ПК-1.1)**
- **Правильный ответ:** Метод ref() используется для создания реактивных данных в Vue.js. Он позволяет отслеживать изменения данных и автоматически обновлять

интерфейс при изменении этих данных. Например, использование `ref()` позволяет создавать переменные, которые автоматически обновляют связанные с ними элементы интерфейса при изменении их значений.

14. Какое из следующих утверждений правильно описывает REST API и его применение? (ПК-4.1)

- **Правильный ответ:** REST API (Representational State Transfer Application Programming Interface) - это архитектурный стиль для создания веб-сервисов, который использует стандартные HTTP-методы для взаимодействия с ресурсами. REST API позволяет создавать, читать, обновлять и удалять данные, используя методы GET, POST, PUT и DELETE. Это широко используемый подход для построения масштабируемых и гибких веб-сервисов.

15. Какие меры могут быть приняты для защиты веб-приложений от XSS-атак и почему это важно? (УК-2.1, ПК-2.1)

- **Правильный ответ:** Валидация и экранирование ввода пользователя являются основными мерами защиты веб-приложений от XSS-атак. Валидация проверяет корректность и безопасность введенных данных, а экранирование предотвращает выполнение вредоносного кода. Эти меры важны для предотвращения атак, при которых злоумышленник может внедрить и выполнить вредоносный скрипт на веб-странице, что может привести к утечке данных, краже учетных записей и другим проблемам безопасности.

16. Что такое CORS и как он используется для обеспечения безопасности веб-приложений? (УК-2.2)

- **Правильный ответ:** CORS (Cross-Origin Resource Sharing) - это механизм, который позволяет веб-сайтам обращаться к ресурсам на других доменах, соблюдая определенные правила безопасности. CORS добавляет заголовки HTTP, которые указывают, какие домены и методы запросов разрешены для взаимодействия с сервером. Это помогает предотвратить несанкционированный доступ к ресурсам и защищает веб-приложения от различных атак.

17. Какой атрибут HTML используется для задания пути к внешнему файлу CSS и как это применяется? (ПК-4.1)

- **Правильный ответ:** Атрибут `href` используется в теге `<link>` для задания пути к внешнему файлу CSS. Например, `<link rel="stylesheet" href="styles.css">` связывает HTML-документ с внешним CSS-файлом, что позволяет применить стили из этого файла ко всем элементам документа.

18. Что такое middleware в Express.js и как оно используется для обработки запросов? (ПК-3.1, ПК-3.2)

- **Правильный ответ:** Middleware в Express.js - это функция, которая обрабатывает запросы и ответы в приложении, выполняя различные задачи, такие как аутентификация, логирование и обработка ошибок. Middleware функции могут изменять запросы и ответы, а также передавать управление следующей middleware функции или завершать обработку запроса. Это позволяет гибко и последовательно обрабатывать HTTP-запросы в приложении.

19. Какой метод JavaScript используется для удаления последнего элемента из массива и как он работает? (ПК-1.2)

- **Правильный ответ:** Метод `pop()` используется для удаления последнего элемента из массива в JavaScript. Например, если у нас есть массив чисел, этот метод удалит последний элемент из этого массива, уменьшая его длину.

20. Что такое Vuex в контексте Vue.js и как оно используется для управления состоянием? (ПК-4.2)

- **Правильный ответ:** Vuex - это хранилище состояния для управления данными в приложении на Vue.js. Оно обеспечивает централизованное управление состоянием и упрощает работу с данными в крупных приложениях. Vuex позволяет определять

состояния, мутации, действия и геттеры для управления и изменения данных, обеспечивая реактивность и синхронизацию между различными компонентами приложения.

21. **Какой метод Vue.js используется для отслеживания изменений данных и как он работает? (ПК-1.1, ПК-2.1)**

- **Правильный ответ:** Метод watch() используется в Vue.js для отслеживания изменений данных и выполнения определенных действий при их изменении. Например, можно отслеживать изменение переменной и запускать определенную функцию каждый раз, когда эта переменная изменяется. Это полезно для выполнения дополнительных действий или вычислений на основе изменений данных.

22. **Какое из следующих утверждений правильно описывает принцип работы AJAX и его применение? (УК-2.2, ПК-1.1)**

- **Правильный ответ:** AJAX (Asynchronous JavaScript and XML) - это метод асинхронного взаимодействия клиента с сервером, позволяющий обновлять содержимое веб-страницы без перезагрузки. AJAX-запросы могут отправлять данные на сервер и получать ответы в фоновом режиме, что обеспечивает более плавный и быстрый пользовательский опыт. Например, AJAX используется для автозаполнения полей формы, загрузки новых данных без перезагрузки страницы и других задач.

23. **Какой метод используется для удаления свойства объекта в JavaScript и как он применяется? (ПК-1.2, ПК-2.2)**

- **Правильный ответ:** Оператор delete используется для удаления свойства объекта в JavaScript. Например, если у нас есть объект, вызов этого оператора удалит указанное свойство из объекта.

24. **Какие из перечисленных мер могут улучшить производительность веб-приложений и как они работают? (УК-2.1, ПК-3.2)**

- **Правильный ответ:** Минимизация количества запросов, использование кеширования и оптимизация изображений - все эти меры могут улучшить производительность веб-приложений. Минимизация количества запросов уменьшает задержки и время загрузки страниц. Кеширование позволяет сохранять часто используемые данные, уменьшая нагрузку на сервер. Оптимизация изображений снижает их размер, улучшая время загрузки страниц. Эти меры помогают создать более быстрые и эффективные веб-приложения.

Критерии оценивания для зачета (зачет/незачет)

1. **Понимание основных понятий и терминов**

- **Зачет:** Студент правильно использует основные понятия и термины, объясняет их в контексте.
- **Незачет:** Студент не знает или неправильно использует основные понятия и термины, не может объяснить их.

2. **Точность и полнота ответов на теоретические вопросы**

- **Зачет:** Ответы на большинство вопросов полные и точные, основные аспекты тем раскрыты.
- **Незачет:** Ответы неполные, содержат значительные ошибки или отсутствуют важные аспекты тем.

3. **Способность к применению знаний на практике**

- **Зачет:** Студент правильно решает практические задачи, выполняет SQL-запросы без ошибок.
- **Незачет:** Студент допускает ошибки при решении практических задач, не может правильно выполнить SQL-запросы.

4. **Качество выполнения практических заданий**

- **Зачет:** Практические задания выполнены корректно, все необходимые элементы присутствуют.
- **Незачет:** Практические задания выполнены некорректно, содержат ошибки или неполные.

Окончательное решение по зачету:

- **Зачет:** Студент демонстрирует понимание основных понятий, корректно отвечает на теоретические вопросы и выполняет практические задания.
- **Незачет:** Студент не демонстрирует понимания основных понятий, отвечает с ошибками на теоретические вопросы и/или не справляется с практическими заданиями.

3.2 Вопросы для проведения промежуточной аттестации в форме ЭКЗАМЕНА 8 семестр

- 1. Что представляет собой HTML и какие его основные функции? (УК-2.1)**
 - **Правильный ответ:** HTML (HyperText Markup Language) - это язык разметки, используемый для создания и структурирования веб-страниц. Он определяет структуру веб-документов с помощью различных тегов и атрибутов. Основные функции HTML включают создание текстового контента, заголовков, параграфов, списков, гиперссылок, изображений, таблиц и форм, которые составляют основу веб-страницы.
- 2. Какой тег используется для создания гиперссылки в HTML и как он работает? (ПК-4.1)**
 - **Правильный ответ:** Тег <a> используется для создания гиперссылок в HTML. Он позволяет ссылаться на другие веб-страницы или ресурсы, используя атрибут href, который указывает URL целевого ресурса. Например, при клике на такую ссылку пользователь будет перенаправлен на указанный ресурс. Гиперссылки могут также использоваться для перехода к другим частям той же страницы.
- 3. Что такое CSS и как он применяется для стилизации веб-страниц? (УК-2.2)**
 - **Правильный ответ:** CSS (Cascading Style Sheets) - это язык, используемый для описания внешнего вида и форматирования HTML-документов. Он позволяет управлять стилями элементов, такими как цвета, шрифты, отступы, границы и расположение на странице. CSS отделяет структуру документа (HTML) от его представления, обеспечивая гибкость и возможности для более сложного и привлекательного дизайна. Стили могут применяться к элементам через встроенные стили, внутренние стили в документе и внешние CSS-файлы.
- 4. Какой из селекторов CSS используется для выбора элемента по его идентификатору и как это работает? (ПК-4.2)**
 - **Правильный ответ:** Селектор #id используется для выбора элемента по его идентификатору в CSS. Идентификатор задается атрибутом id в HTML и должен быть уникальным для каждого элемента. Например, чтобы выбрать элемент с id="header", используется CSS-селектор #header. Это позволяет применять стили непосредственно к конкретному элементу, обеспечивая точное и специфическое стилизование.
- 5. Какой метод HTTP используется для отправки данных на сервер и в каких случаях его применяют? (ПК-1.1)**
 - **Правильный ответ:** Метод POST используется для отправки данных на сервер в HTTP. Он обычно применяется при отправке форм или данных, требующих обработки на сервере, таких как регистрация пользователей, вход в систему или отправка комментариев. POST-запросы отправляют данные в теле запроса, что позволяет передавать большие объемы информации и более безопасно работать с чувствительными данными.
- 6. Что такое DOM в контексте JavaScript и как он используется для взаимодействия с веб-страницей? (УК-2.1, ПК-2.1)**

- **Правильный ответ:** DOM (Document Object Model) - это объектная модель документа, представляющая структуру HTML или XML-документа в виде дерева объектов. DOM позволяет JavaScript взаимодействовать с содержимым и структурой веб-страницы, изменяя элементы, атрибуты, стили и события. Например, с помощью DOM можно добавлять новые элементы на страницу, изменять текст существующих элементов или обрабатывать пользовательские действия, такие как клики и ввод данных.
7. **Какой метод JavaScript используется для добавления элемента в конец массива и как он работает? (ПК-1.2)**
- **Правильный ответ:** Метод push() используется для добавления одного или нескольких элементов в конец массива в JavaScript. Например, если у нас есть массив чисел, метод push добавляет новый элемент в конец этого массива, увеличивая его длину. Это полезно для динамического добавления данных в массив.
8. **Что делает функция JSON.stringify() в JavaScript и в каких случаях она полезна? (ПК-2.2)**
- **Правильный ответ:** Функция JSON.stringify() преобразует объект JavaScript в строку JSON. Это полезно для передачи данных между клиентом и сервером в формате JSON. Например, перед отправкой объекта через AJAX-запрос, его можно преобразовать в строку JSON, чтобы сервер мог его правильно обработать. Это упрощает сериализацию данных и обмен ими между различными системами.
9. **Что такое Node.js и какие возможности оно предоставляет для разработки серверных приложений? (ПК-3.1)**
- **Правильный ответ:** Node.js - это среда выполнения JavaScript вне браузера, основанная на движке V8. Она позволяет создавать серверные приложения с использованием JavaScript. Node.js предоставляет множество модулей для работы с HTTP, файловой системой, базами данных и другими системными ресурсами, что делает его мощным инструментом для разработки серверных приложений и веб-сервисов. Node.js также поддерживает асинхронное программирование, что позволяет эффективно обрабатывать большое количество запросов.
10. **Какой модуль в Node.js используется для создания HTTP-сервера и как это делается? (ПК-3.2)**
- **Правильный ответ:** Модуль http используется для создания HTTP-сервера в Node.js. Этот модуль позволяет создавать сервер, который будет слушать запросы на определенном порту и отправлять ответы клиентам. Сервер может обрабатывать различные типы запросов, такие как GET и POST, и отправлять соответствующие ответы. Это основа для создания веб-приложений и API на Node.js.
11. **Какой метод HTTP используется для получения данных с сервера и в каких ситуациях его применяют? (УК-2.2)**
- **Правильный ответ:** Метод GET используется для получения данных с сервера в HTTP. Он обычно применяется для запросов, которые не изменяют состояние сервера, таких как получение информации о пользователе, загрузка веб-страниц или получение данных из базы данных. GET-запросы могут кэшироваться и индексироваться поисковыми системами, что делает их подходящими для запросов, которые могут часто повторяться.
12. **Что из перечисленного является основным компонентом Vue.js и как он используется? (ПК-4.2)**
- **Правильный ответ:** Основным компонентом Vue.js является компонент (Component). Компоненты в Vue.js инкапсулируют HTML, CSS и JavaScript, обеспечивая модульность и переиспользуемость кода. Компоненты можно создавать и использовать для создания сложных интерфейсов из отдельных,

независимых блоков. Это позволяет эффективно управлять кодом и упрощать разработку сложных приложений.

13. **Какой из методов Vue.js используется для создания реактивных данных и как он работает? (ПК-1.1)**

- **Правильный ответ:** Метод `ref()` используется для создания реактивных данных в Vue.js. Он позволяет отслеживать изменения данных и автоматически обновлять интерфейс при изменении этих данных. Например, использование `ref()` позволяет создавать переменные, которые автоматически обновляют связанные с ними элементы интерфейса при изменении их значений. Это упрощает управление состоянием приложения и обеспечивает реактивность.

14. **Какое из следующих утверждений правильно описывает REST API и его применение? (ПК-4.1)**

- **Правильный ответ:** REST API (Representational State Transfer Application Programming Interface) - это архитектурный стиль для создания веб-сервисов, который использует стандартные HTTP-методы для взаимодействия с ресурсами. REST API позволяет создавать, читать, обновлять и удалять данные, используя методы GET, POST, PUT и DELETE. Это широко используемый подход для построения масштабируемых и гибких веб-сервисов, обеспечивающий простоту и стандартизацию взаимодействия.

15. **Какие меры могут быть приняты для защиты веб-приложений от XSS-атак и почему это важно? (УК-2.1, ПК-2.1)**

- **Правильный ответ:** Валидация и экранирование ввода пользователя являются основными мерами защиты веб-приложений от XSS-атак. Валидация проверяет корректность и безопасность введенных данных, а экранирование предотвращает выполнение вредоносного кода. Эти меры важны для предотвращения атак, при которых злоумышленник может внедрить и выполнить вредоносный скрипт на веб-странице, что может привести к утечке данных, краже учетных записей и другим проблемам безопасности.

16. **Что такое CORS и как он используется для обеспечения безопасности веб-приложений? (УК-2.2)**

- **Правильный ответ:** CORS (Cross-Origin Resource Sharing) - это механизм, который позволяет веб-сайтам обращаться к ресурсам на других доменах, соблюдая определенные правила безопасности. CORS добавляет заголовки HTTP, которые указывают, какие домены и методы запросов разрешены для взаимодействия с сервером. Это помогает предотвратить несанкционированный доступ к ресурсам и защищает веб-приложения от различных атак.

17. **Какой атрибут HTML используется для задания пути к внешнему файлу CSS и как это применяется? (ПК-4.1)**

- **Правильный ответ:** Атрибут `href` используется в теге `<link>` для задания пути к внешнему файлу CSS. Например, `<link rel="stylesheet" href="styles.css">` связывает HTML-документ с внешним CSS-файлом, что позволяет применить стили из этого файла ко всем элементам документа. Это обеспечивает централизованное управление стилями и упрощает поддержку кода.

18. **Что такое middleware в Express.js и как оно используется для обработки запросов? (ПК-3.1, ПК-3.2)**

- **Правильный ответ:** Middleware в Express.js - это функция, которая обрабатывает запросы и ответы в приложении, выполняя различные задачи, такие как аутентификация, логирование и обработка ошибок. Middleware функции могут изменять запросы и ответы, а также передавать управление следующей middleware функции или завершать обработку запроса. Это позволяет гибко и последовательно обрабатывать HTTP-запросы в приложении.

19. **Какой метод JavaScript используется для удаления последнего элемента из массива и как он работает? (ПК-1.2)**

- **Правильный ответ:** Метод pop() используется для удаления последнего элемента из массива в JavaScript. Например, если у нас есть массив чисел, метод pop удалит последний элемент из этого массива, уменьшая его длину. Это полезно для динамического удаления данных из массива.

20. **Что такое Vuex в контексте Vue.js и как оно используется для управления состоянием? (ПК-4.2)**

- **Правильный ответ:** Vuex - это хранилище состояния для управления данными в приложении на Vue.js. Оно обеспечивает централизованное управление состоянием и упрощает работу с данными в крупных приложениях. Vuex позволяет определять состояния, мутации, действия и геттеры для управления и изменения данных, обеспечивая реактивность и синхронизацию между различными компонентами приложения.

21. **Какой метод Vue.js используется для отслеживания изменений данных и как он работает? (ПК-1.1, ПК-2.1)**

- **Правильный ответ:** Метод watch() используется в Vue.js для отслеживания изменений данных и выполнения определенных действий при их изменении. Например, можно отслеживать изменение переменной и запускать определенную функцию каждый раз, когда эта переменная изменяется. Это полезно для выполнения дополнительных действий или вычислений на основе изменений данных.

22. **Какое из следующих утверждений правильно описывает принцип работы AJAX и его применение? (УК-2.2, ПК-1.1)**

- **Правильный ответ:** AJAX (Asynchronous JavaScript and XML) - это метод асинхронного взаимодействия клиента с сервером, позволяющий обновлять содержимое веб-страницы без перезагрузки. AJAX-запросы могут отправлять данные на сервер и получать ответы в фоновом режиме, что обеспечивает более плавный и быстрый пользовательский опыт. Например, AJAX используется для автозаполнения полей формы, загрузки новых данных без перезагрузки страницы и других задач.

23. **Какой метод используется для удаления свойства объекта в JavaScript и как он применяется? (ПК-1.2, ПК-2.2)**

- **Правильный ответ:** Оператор delete используется для удаления свойства объекта в JavaScript. Например, если у нас есть объект, вызов этого оператора удалит указанное свойство из объекта. Это полезно для динамического управления свойствами объектов.

24. **Какие из перечисленных мер могут улучшить производительность веб-приложений и как они работают? (УК-2.1, ПК-3.2)**

- **Правильный ответ:** Минимизация количества запросов, использование кеширования и оптимизация изображений - все эти меры могут улучшить производительность веб-приложений. Минимизация количества запросов уменьшает задержки и время загрузки страниц. Кеширование позволяет сохранять часто используемые данные, уменьшая нагрузку на сервер. Оптимизация изображений снижает их размер, улучшая время загрузки страниц. Эти меры помогают создать более быстрые и эффективные веб-приложения.

25. **Что такое ECMAScript и как он связан с JavaScript? (УК-2.1)**

- **Правильный ответ:** ECMAScript - это спецификация стандарта для языка JavaScript, определяющая его синтаксис и поведение. JavaScript является реализацией стандарта ECMAScript. Спецификация ECMAScript обновляется и развивается, добавляя новые возможности и улучшения в язык, что отражается в различных версиях JavaScript.

26. **Какой оператор в JavaScript используется для создания промиса и как он работает? (ПК-1.1)**
- **Правильный ответ:** Оператор `new Promise` используется для создания нового промиса в JavaScript. Промисы представляют собой объекты, которые позволяют работать с асинхронными операциями, предоставляя методы для обработки успешного выполнения или ошибки. Промисы упрощают управление асинхронными задачами и обеспечивают более чистый и удобочитаемый код.
27. **Что такое Vue Router и как он используется для создания одностраничных приложений? (ПК-4.2)**
- **Правильный ответ:** Vue Router - это маршрутизатор для создания одностраничных приложений на Vue.js. Он позволяет управлять навигацией между различными компонентами, обеспечивая изменение URL без перезагрузки страницы. Vue Router упрощает создание динамических и интерактивных веб-приложений, предоставляя функции для определения маршрутов, передачи параметров и обработки навигационных событий.
28. **Какое свойство CSS отвечает за прозрачность элемента и как оно используется? (УК-2.2)**
- **Правильный ответ:** Свойство `opacity` в CSS отвечает за прозрачность элемента. Оно задает уровень непрозрачности элемента, где значение от 0 (полностью прозрачный) до 1 (полностью непрозрачный). Например, значение `opacity: 0.5` делает элемент наполовину прозрачным, что позволяет видеть элементы, расположенные под ним.
29. **Что такое WebSocket и как он используется для двустороннего обмена данными? (ПК-3.1)**
- **Правильный ответ:** WebSocket - это протокол для двустороннего обмена данными в режиме реального времени между клиентом и сервером. Он устанавливает постоянное соединение, позволяя отправлять и получать данные одновременно. WebSocket часто используется для создания приложений, требующих мгновенного обновления данных, таких как чаты, онлайн-игры и системы уведомлений.
30. **Какой метод JavaScript используется для создания асинхронных функций и как он работает? (ПК-2.2)**
- **Правильный ответ:** Оператор `async` используется для создания асинхронных функций в JavaScript. Асинхронные функции позволяют работать с промисами и упрощают управление асинхронным кодом. Они используют ключевое слово `await` для приостановки выполнения функции до завершения промиса, что делает код более последовательным и удобочитаемым.
31. **Что такое unit-тестирование и как оно применяется в разработке? (ПК-2.1, ПК-2.2)**
- **Правильный ответ:** Unit-тестирование - это метод тестирования, при котором отдельные модули или компоненты кода проверяются на правильность работы. Unit-тесты изолируют отдельные части кода и проверяют их поведение в различных сценариях. Это помогает выявлять ошибки на ранних этапах разработки, упрощает отладку и повышает качество кода.
32. **Какой инструмент используется для автоматического тестирования Vue.js компонентов и как он работает? (ПК-4.1)**
- **Правильный ответ:** Jest - это инструмент для автоматического тестирования Vue.js компонентов. Jest позволяет писать и выполнять тесты, проверяя правильность работы компонентов и их взаимодействие с другими частями приложения. Jest поддерживает модульное тестирование, что позволяет изолировать компоненты и проверять их в различных сценариях.
33. **Что такое CI/CD и как оно используется в разработке программного обеспечения? (ПК-3.2)**

- **Правильный ответ:** CI/CD (Continuous Integration/Continuous Deployment) - это методология, включающая непрерывную интеграцию и непрерывное развертывание, обеспечивающая автоматизацию разработки и доставки программного обеспечения. CI/CD помогает уменьшить время развертывания новых версий, автоматизируя тестирование и деплой, что позволяет быстро и безопасно внедрять изменения.
34. **Какой тег HTML используется для внедрения JavaScript кода в страницу и как это работает? (УК-2.2)**
- **Правильный ответ:** Тег `<script>` используется для внедрения JavaScript кода в HTML-страницу. Этот тег позволяет включать внешние JavaScript-файлы или писать встроенные скрипты непосредственно в HTML-документе. Скрипты могут выполняться при загрузке страницы или при наступлении определенных событий, что позволяет динамически изменять содержимое и поведение страницы.
35. **Что из перечисленного является примером асинхронного взаимодействия в JavaScript и как оно работает? (ПК-2.2)**
- **Правильный ответ:** Метод `setTimeout()` является примером асинхронного взаимодействия в JavaScript. Этот метод выполняет указанный код через заданный промежуток времени. Асинхронные операции позволяют выполнять другие задачи, не блокируя основной поток выполнения, что улучшает отзывчивость приложения.
36. **Какой метод JavaScript используется для удаления первого элемента из массива и как он работает? (ПК-1.2)**
- **Правильный ответ:** Метод `shift()` используется для удаления первого элемента из массива в JavaScript. Этот метод удаляет первый элемент массива и возвращает его значение, сдвигая все остальные элементы на одну позицию влево. Это полезно для работы с очередями и другими структурами данных, где важен порядок элементов.
37. **Какое свойство CSS используется для изменения цвета текста и как оно применяется? (УК-2.2)**
- **Правильный ответ:** Свойство `color` в CSS используется для изменения цвета текста. Оно задает цвет текста элементов и может принимать значения в виде цветовых слов, шестнадцатеричных кодов или RGB/RGBA значений. Например, значение `color: red` делает текст красным.
38. **Что такое JSON и как он используется для обмена данными? (ПК-4.1)**
- **Правильный ответ:** JSON (JavaScript Object Notation) - это формат обмена данными, основанный на JavaScript, используемый для передачи структурированных данных. JSON представляет собой текстовый формат, который легко читается и пишется как человеком, так и машиной. JSON широко используется для обмена данными между клиентом и сервером, обеспечивая удобный способ сериализации и десериализации данных.
39. **Какой инструмент используется для проверки производительности веб-страниц и как он работает? (ПК-3.2)**
- **Правильный ответ:** Lighthouse - это инструмент для проверки производительности веб-страниц. Lighthouse анализирует различные аспекты производительности, доступности и SEO, предоставляя отчеты с рекомендациями по улучшению. Этот инструмент помогает оптимизировать веб-страницы, делая их более быстрыми и эффективными.
40. **Что такое E2E-тестирование и как оно применяется в разработке? (ПК-2.1, ПК-2.2)**
- **Правильный ответ:** E2E (End-to-End) тестирование - это метод тестирования, который проверяет работу всей системы от начала до конца, включая взаимодействие всех компонентов. E2E-тесты имитируют реальные сценарии использования, проверяя, как система работает в целом. Это помогает выявлять проблемы на уровне интеграции и обеспечения качества конечного продукта.

41. **Какой метод JavaScript используется для создания нового массива из всех элементов, которые прошли проверку, реализованную в предоставленной функции, и как он работает? (ПК-1.2)**
- **Правильный ответ:** Метод `filter()` используется для создания нового массива из всех элементов, которые прошли проверку, реализованную в предоставленной функции. Этот метод применяет функцию к каждому элементу массива и возвращает новый массив, содержащий только те элементы, для которых функция вернула `true`. Это полезно для фильтрации данных на основе определенных критериев.
42. **Что такое пакетный менеджер `npm` и как он используется для управления зависимостями проекта? (ПК-4.1)**
- **Правильный ответ:** `npm` (Node Package Manager) - это пакетный менеджер для JavaScript, который позволяет управлять зависимостями проекта. `npm` предоставляет доступ к библиотекам и модулям, которые можно установить и использовать в проекте. `npm` упрощает установку, обновление и удаление зависимостей, обеспечивая удобство управления и поддержку экосистемы JavaScript.
43. **Какое из следующих утверждений правильно описывает `async/await` и как они используются в JavaScript? (ПК-2.2)**
- **Правильный ответ:** Конструкции `async/await` упрощают работу с промисами и асинхронным кодом в JavaScript, делая его более читаемым и управляемым. `async` позволяет объявлять асинхронные функции, которые возвращают промисы, а `await` приостанавливает выполнение функции до завершения промиса. Это помогает писать асинхронный код в линейном стиле, улучшая его читаемость и поддержку.

Критерии оценки ответов на экзамене

- **"Отлично" (5 баллов)**
 - **Критерии:**
 - Полное и точное объяснение вопроса.
 - Ответ включает все ключевые аспекты и детали.
 - Примеры, если требуются, приведены и правильно объяснены.
 - Ответ демонстрирует глубокое понимание темы.
- **"Хорошо" (4 балла)**
 - **Критерии:**
 - Корректное объяснение вопроса.
 - Ответ охватывает основные аспекты, но может отсутствовать незначительная деталь или пример.
 - Демонстрируется хорошее, но не полное понимание темы.
- **"Удовлетворительно" (3 балла)**
 - **Критерии:**
 - Общее представление о вопросе.
 - Ответ включает основные аспекты, но содержит неточности или пропуски.
 - Примеры, если требуются, могут отсутствовать или быть неверно объяснены.
 - Демонстрируется базовое понимание темы.
- **"Неудовлетворительно" (2 балла)**
 - **Критерии:**
 - Некорректное или неполное объяснение вопроса.
 - Отсутствие ключевых аспектов и деталей.
 - Примеры, если требуются, отсутствуют или приведены неверные.
 - Ответ демонстрирует недостаточное понимание темы.

