

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Максимов Алексей Борисович

Должность: директор департамента по образовательной политике

Дата подписания: 12.10.2026 16:48:36

Уникальный идентификатор документа

8db180d1a3f02ac9e60521a5672742735c18b1d6

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета

«Информационных технологий»



/ Д.Г. Демидов /

« 26 » февраль 2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Веб-программирование и дизайн**

Направление подготовки

09.03.01 Информатика и вычислительная техника

Профиль подготовки (образовательная программа)

«Информационные системы умных пространств»

Год начала обучения:

2026

Квалификация (степень) выпускника

Бакалавр

Форма обучения

Очная

Москва – 2026

Рабочая программа составлена на основе ФГОС ВО и учебного плана Московского политехнического университета по направлению (специальности) 09.03.01 Информатики и вычислительная техника, по профилю подготовки Информационные системы умных пространств.

Разработчик(и):

к. ф.-м. н., доцент кафедры



/ Т.Т. Идиатуллин /

Согласовано:

Заведующий кафедрой
«СМАРТ-технологии»,
к.т.н., доцент



/ Е.В. Петрунина /

Содержание

1. Цели, задачи и планируемые результаты обучения по дисциплине
2. Место дисциплины в структуре образовательной программы
3. Структура и содержание дисциплины
 - 3.1 Виды учебной работы и трудоемкость
 - 3.2 Тематический план изучения дисциплины (по формам обучения)
 - 3.3 Содержание дисциплины
 - 3.4 Тематика семинарских/практических и лабораторных занятий
 - 3.5 Тематика вопросов для самостоятельного изучения
4. Учебно-методическое и информационное обеспечение
 - 4.1 Нормативные документы и ГОСТы
 - 4.2. Основная литература
 - 4.3 Дополнительная литература
 - 4.4 Электронные образовательные ресурсы
 - 4.5 Лицензионное и свободно распространяемое программное обеспечение
 - 4.6 Современные профессиональные базы данных и информационные справочные системы
- 5 Материально-техническое обеспечение
- 6 Методические рекомендации
 - 6.1 Методические рекомендации для преподавателя по организации обучения
 - 6.2 Методические указания для обучающихся по освоению дисциплины
 - 6.3 Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
- 7 Фонд оценочных средств
 - 7.1 Методы контроля и оценивания результатов обучения
 - 7.2 Шкала и критерии оценивания результатов обучения
- Приложение 1
- Приложение 2

1 Цели, задачи и планируемые результаты обучения по дисциплине

К основным целям освоения дисциплины «Веб-программирование и дизайн» следует отнести:

- формирование у обучающихся системного представления об архитектуре веб-приложений и принципах клиент-серверного взаимодействия в сети Интернет;
- ознакомление обучающихся с современными технологиями верстки, программирования на стороне клиента и сервера, а также с принципами веб-дизайна;
- формирование навыков проектирования, разработки, тестирования и развертывания веб-приложений, в том числе веб-панелей управления и мониторинга систем умных пространств.

К основным задачам освоения дисциплины «Веб-программирование и дизайн» следует отнести:

- изучение архитектуры веб-приложений, протокола HTTP и модели клиент-серверного взаимодействия;
- изучение языка гипертекстовой разметки HTML и каскадных таблиц стилей CSS, семантической верстки и адаптивного дизайна;
- изучение языка программирования JavaScript, модели документа (DOM) и обработки событий на стороне клиента;
- изучение подходов к компонентной и одностраничной разработке пользовательских интерфейсов веб-приложений;
- изучение серверной разработки веб-приложений: веб-фреймворки, маршрутизация запросов, шаблонизация, работа с базами данных;
- изучение принципов построения программных интерфейсов (API) веб-приложений, взаимодействия по технологии REST и асинхронного обмена данными;
- изучение основ безопасности веб-приложений и типовых уязвимостей;
- приобретение навыков тестирования, оптимизации и развертывания веб-приложений, в том числе для отображения данных и управления устройствами умных пространств.

Обучение по дисциплине «Веб-программирование и дизайн» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение	ИПК-1.1. Знает способы разработки требований и проектирования программного обеспечения в области интернета вещей и интеллектуализации пространств ИПК-1.2. Умеет проектировать программное обеспечение с применением современных инструментальных средств в области интернета вещей и интеллектуализации пространств ИПК-1.3. Имеет навыки разработки требований и проектирования программного обеспечения с применением современных инструментальных средств в области интернета вещей и интеллектуализации пространств

2 Место дисциплины в структуре образовательной программы

Дисциплина «Веб-программирование и дизайн» относится к числу профессиональных учебных дисциплин базового цикла ((Б1.2) части, формируемой участниками образовательных отношений основной образовательной программы бакалавриата по направлению подготовки 09.03.01 «Информатика и вычислительная техника», профиль «Информационные системы умных пространств».

Изучение дисциплины опирается на знания, умения и навыки, приобретенные в предшествующих дисциплинах: «Программирование», «Базы данных», «UI/UX-дизайн», «Протоколы IoT», «Архитектура систем умных пространств», «Тестирование программного обеспечения».

Дисциплина обеспечивает изучение дисциплины «Информационные системы умных пространств», а также выполнение выпускной квалификационной работы в части, связанной с разработкой веб-компонентов системы.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы, т.е. 144 академических часа, из них аудиторных — 54 часа, самостоятельная работа обучающихся — 90 часов. Форма промежуточной аттестации — экзамен в 7 семестре.

3.1 Виды учебной работы и трудоемкость

(по очной форме обучения)

№ п/п	Вид учебной работы	Количество часов	Семестр 7
1	Аудиторные занятия	54	54
	В том числе:		
1.1	Лекции	18	18
1.2	Семинарские/практические занятия	—	—
1.3	Лабораторные занятия	36	36
2	Самостоятельная работа	90	90
3	Промежуточная аттестация	экзамен	экзамен
	Итого:	144	144

3.2 Тематический план изучения дисциплины

(по очной форме обучения)

№ п/п	Разделы/темы дисциплины	Всего, час.	Лекции	Лабораторные занятия	Самостоятельная работа
1	Тема 1. Архитектура веб-приложений и протокол HTTP	16	2	4	10
2	Тема 2. Язык разметки HTML и каскадные таблицы стилей CSS	18	2	4	12
3	Тема 3. Адаптивный и отзывчивый веб-дизайн	16	2	4	10
4	Тема 4. Язык программирования JavaScript, модель документа и события	18	2	4	12
5	Тема 5. Компонентная и одностраничная разработка пользовательского интерфейса	18	2	4	12
6	Тема 6. Серверная разработка веб-приложений	16	2	4	10
7	Тема 7. Программные интерфейсы (API), асинхронное	18	2	4	12

№ п/п	Разделы/темы дисциплины	Всего, час.	Лекции	Лабораторные занятия	Самостоятельная работа
	взаимодействие и основы безопасности				
8	Тема 8. Тестирование, оптимизация и развертывание веб- приложений умных пространств	24	4	8	12
	Итого	144	18	36	90

3.3 Содержание дисциплины

3.3.1 Содержание лекционных занятий (18 часов)

Тема 1. Архитектура веб-приложений и протокол HTTP (2 ч)

Модель клиент-серверного взаимодействия во всемирной паутине, роль браузера как клиента и веб-сервера. Протокол HTTP: структура запроса и ответа, методы запроса, коды состояния, заголовки. Понятие URI и URL, структура унифицированного указателя ресурса. Протокол HTTPS и его роль в защите передаваемых данных. Многоуровневая архитектура веб-приложения: клиентская часть (frontend), серверная часть (backend), база данных. Модели рендеринга страниц: серверный рендеринг, клиентский рендеринг, гибридные подходы. Обзор жизненного цикла разработки веб-приложения и инструментов веб-разработчика, встроенных в браузер.

Тема 2. Язык разметки HTML и каскадные таблицы стилей CSS (2 ч)

Структура документа HTML: элементы, атрибуты, вложенность. Семантическая разметка документа: заголовки, разделы, навигация, статьи. Формы ввода данных и элементы управления, их атрибуты и валидация на стороне клиента. Основы каскадных таблиц стилей: селекторы, специфичность, каскадирование и наследование. Блочная модель CSS: содержимое, отступы, границы, внешние отступы. Позиционирование элементов и организация макета страницы. Работа со шрифтами, цветом и фоновым оформлением. Использование препроцессоров стилей для повышения удобства сопровождения кода.

Тема 3. Адаптивный и отзывчивый веб-дизайн (2 ч)

Понятие адаптивного и отзывчивого дизайна, различия подходов. Медиазапросы и построение макета под различные размеры экрана. Технологии организации макета Flexbox и CSS Grid: принципы работы и области применения. Мобильно-ориентированный подход к разработке

(mobile-first). Оптимизация изображений и мультимедийного содержимого для различных устройств. Библиотеки и фреймворки CSS для ускорения верстки, компонентные системы стилей. Применение принципов, изученных в дисциплине UI/UX-дизайн, к разработке веб-интерфейсов: типографика, цветовые схемы, доступность.

Тема 4. Язык программирования JavaScript, модель документа и события (2 ч)

Основные конструкции языка JavaScript: переменные, типы данных, функции, объекты, массивы. Модель объекта документа (DOM): дерево элементов страницы, методы поиска и изменения элементов. Обработка событий пользовательского интерфейса: назначение обработчиков, всплытие и перехват событий. Асинхронное программирование в JavaScript: функции обратного вызова, промисы, синтаксис `async/await`. Работа с таймерами и организация периодических действий на странице. Модульная организация кода JavaScript, системы сборки клиентского кода. Отладка кода JavaScript средствами браузера.

Тема 5. Компонентная и одностраничная разработка пользовательского интерфейса (2 ч)

Понятие одностраничного приложения (single-page application), отличие от многостраничных сайтов. Компонентный подход к разработке пользовательского интерфейса: инкапсуляция разметки, стилей и логики в переиспользуемый компонент. Управление состоянием компонентов и приложения в целом. Организация навигации в одностраничном приложении без перезагрузки страницы (клиентская маршрутизация). Жизненный цикл компонента и оптимизация повторного рендеринга. Взаимодействие компонентов и передача данных между ними. Обзор современных инструментов сборки и разработки клиентских приложений.

Тема 6. Серверная разработка веб-приложений (2 ч)

Архитектура серверной части веб-приложения: обработка запросов, маршрутизация, контроллеры. Веб-фреймворки для серверной разработки: назначение, основные компоненты, принципы организации кода. Шаблонизация страниц на стороне сервера, разделение представления и логики. Организация взаимодействия серверного приложения с базой данных: объектно-реляционное отображение, выполнение запросов. Аутентификация и управление сессиями пользователей на сервере. Обработка загружаемых файлов и статического содержимого. Логирование и обработка ошибок

серверного приложения. Развертывание серверного приложения и обеспечение его работы под нагрузкой нескольких пользователей.

Тема 7. Программные интерфейсы (API), асинхронное взаимодействие и основы безопасности (2 ч)

Проектирование программного интерфейса (API) веб-приложения в архитектурном стиле REST: ресурсы, методы, коды ответов. Форматы обмена данными между клиентом и сервером, сериализация и десериализация JSON. Организация асинхронных запросов с клиентской части средствами Fetch API и аналогичных технологий. Взаимодействие в реальном времени: технология WebSocket и ее применение для отображения телеметрии умного пространства в браузере без перезагрузки страницы. Основы безопасности веб-приложений: межсайтовый скриптинг (XSS), подделка межсайтовых запросов (CSRF), SQL-инъекции, меры защиты от типовых уязвимостей. Защита паролей и передаваемых данных, применение протокола HTTPS. Ограничение межсайтовых запросов (CORS) и его настройка.

Тема 8. Тестирование, оптимизация и развертывание веб-приложений умных пространств (4 ч)

Особенности проектирования веб-панели мониторинга и управления системой умного пространства: отображение состояния устройств в реальном времени, визуализация телеметрии, управление сценариями автоматизации. Обеспечение доступности и удобства использования веб-интерфейса на различных устройствах. Тестирование веб-приложений: функциональное тестирование, кроссбраузерное тестирование, автоматизация тестирования пользовательского интерфейса. Оптимизация производительности веб-приложения: минимизация и объединение ресурсов, кэширование, отложенная загрузка содержимого. Подготовка веб-приложения к развертыванию: сборка производственной версии, настройка веб-сервера, применение контейнеризации. Мониторинг работоспособности развернутого веб-приложения. Тенденции развития веб-технологий и их применение в системах умных пространств.

3.3.2 Содержание лабораторных занятий (36 часов)

Лабораторная работа № 1. Разработка структуры веб-страницы средствами HTML и CSS (4 ч)

Разработка семантической структуры веб-страницы с использованием элементов HTML. Разработка формы ввода данных с валидацией на стороне клиента. Оформление страницы средствами CSS: блочная модель,

позиционирование, типографика. Проверка корректности отображения страницы в нескольких браузерах.

Лабораторная работа № 2. Разработка адаптивного макета страницы (4 ч)

Разработка макета страницы с использованием технологии Flexbox или CSS Grid. Настройка медиазапросов для адаптации макета под мобильные устройства, планшеты и настольные компьютеры. Оптимизация изображений для различных размеров экрана. Проверка отзывчивости макета на эмуляторах различных устройств.

Лабораторная работа № 3. Разработка интерактивности страницы средствами JavaScript (4 ч)

Реализация обработки событий пользовательского интерфейса (нажатия, ввод данных) средствами JavaScript. Реализация динамического изменения содержимого страницы через модель документа (DOM). Реализация асинхронного запроса данных с использованием промисов и синтаксиса `async/await`. Отладка разработанного кода средствами браузера.

Лабораторная работа № 4. Разработка компонентов пользовательского интерфейса (4 ч)

Разработка одного или нескольких переиспользуемых компонентов пользовательского интерфейса (например, карточка устройства, элемент списка). Организация передачи данных между компонентами и управления общим состоянием. Реализация клиентской маршрутизации между несколькими экранами приложения без перезагрузки страницы. Тестирование работоспособности разработанных компонентов.

Лабораторная работа № 5. Разработка серверной части веб-приложения (4 ч)

Разработка серверного приложения с несколькими маршрутами обработки запросов средствами веб-фреймворка. Реализация шаблонизации страниц на стороне сервера. Организация взаимодействия серверного приложения с базой данных: выполнение операций чтения и записи. Реализация простой формы аутентификации пользователя.

Лабораторная работа № 6. Разработка программного интерфейса (API) и асинхронное взаимодействие (4 ч)

Проектирование и реализация REST API для управления данными о состоянии устройств умного пространства. Реализация обращения к разработанному API с клиентской стороны средствами Fetch API. Настройка передачи данных в реальном времени по протоколу WebSocket для

отображения изменяющейся телеметрии. Тестирование разработанного API с помощью специализированного инструмента.

Лабораторная работа № 7. Обеспечение безопасности веб-приложения (4 ч)

Анализ разработанного в предыдущих занятиях приложения на наличие типовых уязвимостей (межсайтовый скриптинг, отсутствие экранирования данных). Реализация мер защиты от выявленных уязвимостей. Настройка защищенного соединения по протоколу HTTPS на учебном стенде. Настройка политики ограничения межсайтовых запросов (CORS) для разработанного API.

Лабораторная работа № 8. Разработка веб-панели мониторинга умного пространства и ее развертывание (8 ч)

Разработка веб-панели мониторинга и управления лабораторным макетом умного пространства, объединяющей ранее разработанные компоненты: отображение телеметрии в реальном времени, управление устройствами через API, адаптивный интерфейс. Проведение функционального и кроссбраузерного тестирования разработанного приложения. Оптимизация производительности приложения и подготовка производственной сборки. Развертывание приложения на учебном веб-сервере и защита выполненной работы.

3.3.3 Содержание самостоятельной работы обучающихся (90 часов)

Самостоятельная работа обучающихся включает следующие виды деятельности:

- проработка конспектов лекций и изучение разделов дисциплины, вынесенных на самостоятельное изучение, по рекомендованной литературе и электронным образовательным ресурсам;
- изучение официальной документации языков разметки, стилей, программирования и используемых веб-фреймворков, в том числе на английском языке;
- подготовка к выполнению и защите лабораторных работ, оформление отчетов;
- выполнение сквозного индивидуального задания по разработке веб-приложения (панели мониторинга или управления) для заданного сценария умного пространства;
- анализ и разбор существующих веб-интерфейсов панелей управления умными пространствами;

- подготовка обзора (реферата) по современным веб-технологиям и фреймворкам;
- подготовка к тестированию и к экзамену.

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Минобрнауки России от 19.09.2017 № 929 (ред. от 08.02.2021) «Об утверждении федерального государственного образовательного стандарта высшего образования — бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника» (Зарегистрировано в Минюсте России 10 октября 2017 г. № 48489).
3. Академический учебный план. Направление подготовки: 09.03.01 Информатика и вычислительная техника. Профиль: Информационные системы умных пространств. Форма обучения: очная.
4. Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся федерального государственного автономного образовательного учреждения высшего образования «Московский политехнический университет» (утверждено приказом Московского Политеха от 01.12.2022 № 1375ОД).
5. ГОСТ Р 52872-2019. Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы. Требования доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности.
6. ГОСТ Р 56939-2016. Защита информации. Разработка безопасного программного обеспечения. Общие требования.

4.2 Основная литература

1. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. — Москва : Эксмо, 2020. — 480 с.
2. Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган. — Санкт-Петербург : Символ-Плюс, 2020. — 1184 с.
3. Дакетт, Д. JavaScript и jQuery. Интерактивная веб-разработка / Д. Дакетт. — Москва : Эксмо, 2020. — 640 с.

4. Хортон, И. PHP и MySQL. Разработка веб-приложений / И. Хортон. — Санкт-Петербург : БХВ-Петербург, 2020. — 1152 с.
5. Круг, С. Не заставляйте меня думать. Веб-юзабилити — здравый смысл / С. Круг. — Санкт-Петербург : Питер, 2020. — 256 с.

4.3 Дополнительная литература

1. Ньюмен, С. Создание микросервисов / С. Ньюмен. — 2-е изд. — Санкт-Петербург : Питер, 2023. — 624 с.
2. Хаверберке, М. Выразительный JavaScript / М. Хаверберке. — Санкт-Петербург : Питер, 2019. — 336 с.
3. Соломенчук, В. Г. Основы веб-дизайна : учебное пособие / В. Г. Соломенчук. — Санкт-Петербург : БХВ-Петербург, 2018. — 336 с.
4. Бенкен, Е. С. PHP, MySQL, XML: программирование для Интернета / Е. С. Бенкен. — Санкт-Петербург : БХВ-Петербург, 2018. — 400 с.
5. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. — Санкт-Петербург : Питер, 2018. — 352 с.

4.4 Электронные образовательные ресурсы

1. Курс в ЛМС Московского политехнического университета. — <https://lms.mospolytech.ru/>
2. Документация веб-технологий MDN Web Docs. — <https://developer.mozilla.org/>
3. Документация фреймворка React. — <https://react.dev/>
4. Документация серверного фреймворка Node.js Express. — <https://expressjs.com/>
5. Ресурс исследований юзабилити Nielsen Norman Group. — <https://www.nngroup.com/>
6. Проект Open Web Application Security Project (OWASP Top 10). — <https://owasp.org/www-project-top-ten/>
7. Электронно-библиотечная система «Лань». — <https://e.lanbook.com/>
8. Электронно-библиотечная система «Юрайт». — <https://urait.ru/>

4.5 Лицензионное и свободно распространяемое программное обеспечение

1. Операционная система Windows 10 (или выше) — Microsoft Open License.

2. Офисные приложения Microsoft Office 2016 (или выше) — Microsoft Open License.
3. Среда разработки Visual Studio Code — свободно распространяемое ПО.
4. Среда выполнения Node.js и менеджер пакетов npm — свободно распространяемое ПО.
5. Библиотека для разработки пользовательских интерфейсов React (или аналог) — свободно распространяемое ПО (лицензия MIT).
6. Веб-фреймворк Express.js (или аналог для другого языка программирования) — свободно распространяемое ПО.
7. Система управления базами данных PostgreSQL или MySQL — свободно распространяемое ПО.
8. Инструмент тестирования API Postman (бесплатный тарифный план) или аналог — свободно распространяемое ПО.
9. Веб-сервер Nginx — свободно распространяемое ПО (лицензия BSD).
10. Система контроля версий Git — свободно распространяемое ПО (лицензия GPL).

5 Материально-техническое обеспечение

Для проведения всех видов занятий необходимо презентационное оборудование (мультимедийный проектор, компьютер, экран) — 1 комплект.

Для проведения лабораторных занятий необходимо наличие компьютерного класса, оборудованного современной вычислительной техникой из расчета одно рабочее место на одного обучающегося с доступом в сеть Интернет и с установленным программным обеспечением для веб-разработки, а также следующего оборудования:

- лабораторный макет системы умного пространства (датчики, исполнительные устройства) с программным интерфейсом для интеграции с веб-приложением;
- серверное оборудование или виртуальная инфраструктура для развертывания разрабатываемых веб-приложений и учебного веб-сервера;
- мобильные устройства и планшеты для кроссплатформенного и кроссбраузерного тестирования разрабатываемых веб-интерфейсов.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы — примеры кода, разборы удачных и неудачных решений верстки и дизайна, задания для лабораторных работ, а также заранее проверить работоспособность используемого программного обеспечения.

Лабораторный практикум целесообразно организовать как сквозной проект: обучающиеся последовательно разрабатывают веб-панель мониторинга и управления умным пространством, наращивая ее функциональность от статической верстки до полноценного приложения с серверной частью, API и защитой от уязвимостей.

При проведении занятий рекомендуется разбирать примеры реальных веб-интерфейсов, обращая внимание как на технические решения, так и на аспекты дизайна и удобства использования, изученные в курсе UI/UX-дизайна.

При проверке работ и отчетов следует учитывать не только работоспособность приложения, но и качество верстки и оформления, структурированность и читаемость кода, корректность обработки ошибок и нештатных ситуаций, а также внимание к вопросам безопасности разработанного решения.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи с учебным планом. Основой теоретической подготовки обучающихся являются лекции.

В процессе самостоятельной работы обучающиеся закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты лекций, изучают официальную документацию используемых технологий и фреймворков, готовятся к защите лабораторных работ и к экзамену, а также самостоятельно изучают отдельные темы учебной программы.

Особое внимание при подготовке следует уделять практическому написанию кода: рекомендуется регулярно повторять реализацию изученных на занятиях приемов на новых, самостоятельно придуманных примерах, а

также анализировать исходный код открытых веб-проектов для изучения принятых в индустрии практик.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы обучающихся, оценочные средства текущего контроля успеваемости и промежуточной аттестации:

- подготовка к решению прикладных задач;
- выполнение и защита лабораторных работ, оформление отчетов;
- выполнение сквозного индивидуального задания по разработке веб-приложения и его защита;
- подготовка обзора (реферата) по современным веб-технологиям;
- тест;
- экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ПК-1. Способен разрабатывать требования и проектировать программное обеспечение

Показатель	2	3	4	5
знать: архитектуру веб-приложений и протокол HTTP, основы HTML, CSS и JavaScript, принципы серверной разработки, построения API и обеспечения безопасности веб-приложений	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие указанных знаний	Обучающийся демонстрирует неполное соответствие указанных знаний, освещена основная, наиболее важная часть материала, допущена одна значительная ошибка или неточность	Обучающийся демонстрирует частичное соответствие указанных знаний, допускаются 2—3 несущественные ошибки и неточности	Обучающийся демонстрирует полное соответствие указанных знаний, свободно оперирует материалом дисциплины

Показатель	2	3	4	5
уметь: разрабатывать адаптивную верстку и клиентскую логику, проектировать компоненты пользовательского интерфейса, реализовывать серверную часть и API, обеспечивать безопасность веб-приложения	Обучающийся не умеет или в недостаточной степени умеет выполнять указанные действия	Обучающийся демонстрирует неполное соответствие указанных умений, испытывает затруднения при переносе умений на новые ситуации	Обучающийся демонстрирует частичное соответствие указанных умений, допускает незначительные ошибки и неточности	Обучающийся демонстрирует полное соответствие указанных умений, уверенно применяет их в нестандартных ситуациях
владеть: навыками разработки, тестирования, оптимизации и развертывания веб-приложений, в том числе панелей мониторинга и управления системами умных пространств	Обучающийся не владеет или в недостаточной степени владеет указанными навыками	Обучающийся владеет указанными навыками, но допускаются значительные ошибки, проявляется недостаточность навыков по ряду показателей	Обучающийся частично владеет указанными навыками, допускаются незначительные ошибки и затруднения при аналитических операциях	Обучающийся в полном объеме владеет указанными навыками, свободно оперирует приобретенными знаниями

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю), методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Обучающийся демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, свободно оперирует приобретенными знаниями, умениями и навыками, применяет их в ситуациях повышенной сложности. Все лабораторные работы и индивидуальное задание выполнены и защищены. Присутствовал более чем на $\frac{3}{4}$ занятий по дисциплине.
хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Обучающийся демонстрирует неполное, но правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо при этом были допущены 2—3 незначительные ошибки. Присутствовал более чем на $\frac{3}{4}$ занятий по дисциплине.
удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Обучающийся демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность. Присутствовал более чем на $\frac{1}{2}$ занятий по дисциплине.
неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Обучающийся демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1 Список вопросов для экзамена

1. Модель клиент-серверного взаимодействия во всемирной паутине.
2. Протокол HTTP: методы запроса, коды состояния, заголовки.
3. Протокол HTTPS и его роль в защите передаваемых данных.
4. Многоуровневая архитектура веб-приложения.
5. Модели рендеринга страниц: серверный, клиентский, гибридный.

6. Структура документа HTML и семантическая разметка.
7. Формы ввода данных и валидация на стороне клиента.
8. Основы каскадных таблиц стилей: селекторы, каскадирование, наследование.
9. Блочная модель CSS и позиционирование элементов.
10. Понятие адаптивного и отзывчивого веб-дизайна.
11. Медиазапросы и построение макета под различные размеры экрана.
12. Технологии Flexbox и CSS Grid.
13. Мобильно-ориентированный подход к разработке (mobile-first).
14. Основные конструкции языка JavaScript.
15. Модель объекта документа (DOM) и методы работы с ней.
16. Обработка событий пользовательского интерфейса в JavaScript.
17. Асинхронное программирование в JavaScript: промисы, async/await.
18. Понятие одностраничного приложения (SPA).
19. Компонентный подход к разработке пользовательского интерфейса.
20. Управление состоянием компонентов и приложения.
21. Клиентская маршрутизация в одностраничном приложении.
22. Архитектура серверной части веб-приложения.
23. Назначение и структура веб-фреймворков для серверной разработки.
24. Шаблонизация страниц на стороне сервера.
25. Организация взаимодействия серверного приложения с базой данных.
26. Аутентификация и управление сессиями пользователей на сервере.
27. Проектирование программного интерфейса (API) в архитектурном стиле REST.
28. Форматы обмена данными между клиентом и сервером.
29. Организация асинхронных запросов с клиентской части (Fetch API).
30. Технология WebSocket и ее применение для взаимодействия в реальном времени.
31. Межсайтовый скриптинг (XSS) и меры защиты.
32. Подделка межсайтовых запросов (CSRF) и меры защиты.
33. SQL-инъекции и меры защиты от них.
34. Ограничение межсайтовых запросов (CORS) и его настройка.

35. Особенности проектирования веб-панели мониторинга умного пространства.
36. Тестирование веб-приложений: функциональное и кроссбраузерное тестирование.
37. Оптимизация производительности веб-приложения.
38. Подготовка веб-приложения к развертыванию.
39. Мониторинг работоспособности развернутого веб-приложения.
40. Тенденции развития веб-технологий и их применение в умных пространствах.

7.3.2 Примерные темы индивидуальных заданий и обзоров

- Разработка веб-панели мониторинга параметров микроклимата умного здания.
- Разработка веб-интерфейса управления освещением и климатическими устройствами умного дома.
- Разработка одностраничного приложения для визуализации телеметрии сети датчиков в реальном времени.
- Проектирование и реализация REST API для системы бронирования рабочих пространств офиса.
- Разработка веб-приложения учета энергопотребления умного здания с визуализацией статистики.
- Обеспечение безопасности веб-панели управления системой контроля доступа.
- Сравнительный анализ современных фреймворков для разработки пользовательских интерфейсов.
- Разработка адаптивного веб-интерфейса диспетчерского центра умного пространства.
- Организация взаимодействия в реальном времени между веб-приложением и платформой Интернета вещей по протоколу WebSocket.
- Оптимизация производительности веб-приложения панели управления умным пространством.