

Содержание

1. Наименование дисциплины	3
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.....	3
3. Место дисциплины в структуре образовательной программы	5
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий.....	5
5.1. Содержание дисциплины	5
5.2. Учебно-тематический план	8
5.3. Содержание семинаров, практических занятий.....	8
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	10
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	10
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	11
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	18
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	32
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	32
10. Методические указания для обучающихся по освоению дисциплины.....	33
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	39
11.1 Комплект лицензионного программного обеспечения	40
11.2 Современные профессиональные базы данных, и информационные справочные системы	40
11.3 Сертифицированные программные и аппаратные средства защиты информации	40
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	40

1. Наименование дисциплины

Дисциплина «Веб-разработка».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания) соотнесенные с индикаторами достижения компетенции
ПКП-5	Способность проектировать и реализовывать интеллектуальные информационные системы	1. Демонстрирует знания основных методов машинного обучения и интеллектуального анализа данных, применяет готовые инструменты для создания интеллектуальных алгоритмов.	Знать: основные методы машинного обучения и интеллектуального анализа данных Уметь: применяет готовые инструменты для создания интеллектуальных алгоритмов
		2. Понимает особенности интеллектуальных информационных систем в части операций разработки, развертывания и сопровождения.	Знать: особенности интеллектуальных информационных систем Уметь: применяет знания в части операций разработки, развертывания и сопровождения.
		3. Адаптирует практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем.	Знать: как адаптировать практики создания программных продуктов Уметь: адаптировать практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем
		4. Организует сбор и подготовку данных для систем машинного обучения, в том числе потоковых, онлайн обучения.	Знать: как организовывать сбор и подготовку данных для систем машинного обучения Уметь: организовывать сбор и подготовку данных для систем машинного обучения, в том числе потоковых, онлайн обучения.

ПКП-6	Способность вести разработку программных систем в команде, вести эффективную коммуникацию	1. Придерживается рекомендованного в команде стиля написания кода, проводит рефакторинг написанного кода, в том числе, другими членами команды, проводит код-ревью.	Знать: как придерживаться рекомендованного в команде стиля написания кода Уметь: проводить рефакторинг написанного кода, в том числе, другими членами команды, проводит код-ревью
		2. Документирует код, архитектуру, дизайн, требования к программной системе с использованием принятой в команде системы технических стандартов.	Знать: о документации кода, архитектуру, дизайн, требования к существующим программным системам Уметь: документировать код, архитектуру, дизайн, требования к программной системе с использованием принятой в команде системы технических стандартов
		3. Использует инструментальные средства командной разработки программного обеспечения, контроля версий, отслеживания ошибок, планирования процессов разработки в зависимости от принятой в команде методологии разработки.	Знать: инструментальные средства командной разработки программного обеспечения, контроля версий, отслеживания ошибок, планирования процессов разработки в зависимости от принятой в команде методологии разработки Уметь: использовать инструментальные средства командной разработки программного обеспечения
		4. Выстраивает эффективную двустороннюю коммуникацию с нетехническими специалистами по предмету разработки предметной области в целом.	Знать: как выстраивать эффективную двустороннюю коммуникацию с нетехническими специалистами Уметь: выстраивать эффективную двустороннюю коммуникацию с нетехническими специалистами
		5. Коммуницирует задачи разработки между членами команды, оценивает трудоемкость и сложность выполнения задач разработки, ставит задачи и контролирует их выполнение, в том числе в письменной формализованной форме.	Знать: как ставить задачи разработки между членами команды Уметь: оценивать трудоемкость и сложность выполнения задач разработки, ставить задачи и контролирует их выполнение, в том числе в письменной формализованной форме.

3. Место дисциплины в структуре образовательной программы

Дисциплина «Веб-разработка» является дисциплиной профиля части, формируемой участниками образовательных отношений, учебного плана образовательной программы «Технологии разработки программного обеспечения» по направлению подготовки 09.03.04 Программная инженерия, профиль «Технологии разработки программного обеспечения».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 5 (в часах)	Семестр 6 (в часах)
Общая трудоемкость дисциплины	8/288	126	162
Контактная работа- Аудиторные занятия	118	50	68
<i>Лекции</i>	32	16	16
<i>Семинары, практические занятия</i>	86	34	52
Самостоятельная работа	170	76	94
Вид текущего контроля	Проектная работа, проектная работа	Проектная работа	Проектная работа
Вид промежуточной аттестации	Зачет, экзамен	Зачет	Экзамен

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в веб-разработку

Основные понятие веб-технологий. Основы компьютерных сетей. Сетевые модели и протоколы. Клиент-серверное взаимодействие. Протокол HTTP и служба World Wide Web. История развития веб-приложений. Основы функционирования протокола HTTP. Запросы и ответы. Структура запроса. Методы запросов. Поля заголовков запросов и ответов. Коды статусов ответов. Программное обеспечение, необходимое для работы веб-

приложений. Веб-сервер. Браузер. Обзор существующих решений. Клиентская и серверная часть веб-приложений. Языки и инструментальные средства веб-разработки. Программное обеспечение, необходимое для веб-разработки. Текстовые редакторы, IDE, облачные среды разработки. Установка и работа в WebStorm IDE.

Тема 2. Основы HTML

HTML как язык разметки, его назначение, история развития. Основы синтаксиса - теги, атрибуты, содержимое. Структура веб-страницы. Основные теги - теги заголовков и тела страницы. Семантические теги HTML5.

Тема 3. CSS

Основы каскадных таблиц стилей. Принцип разделения структуры и представления. Способы подключения стилей к странице. Основы синтаксиса CSS. Виды селекторов. Основные правила CSS. CSS Flexbox, CSS Grid. CSS препроцессоры. Sass, less. CSS-фреймворки. Bootstrap 4.

Тема 4. JavaScript

Основы клиентского программирования. Способы встраивания сценариев в веб-страницу. Основы и синтаксис языка программирования JavaScript. Событийная модель веб-страницы. Обработчики событий. Событийно-ориентированное программирование. Объектная модель документа DOM. Основные приемы работы с DOM в чистом JavaScript. AJAX и FetchAPI. TypeScript.

Тема 5. Деплой статичных сайтов

Хостинг. Развертывание статичных сайтов. Развертывание сайта на выделенном хостинге. Копирование файлов по FTP. Бесплатные статичные хостинги. GitHub Pages и аналоги. Мониторинг и отладка сайтов. Google Development Tools и аналоги.

Тема 6. Фронтенд фреймворк React.js

Понятие и назначение фронтенд фреймворков. Фреймворк React.js - общая характеристика, синтаксис, основные приемы работы. JSX. Функциональные и классовые компоненты, рендеринг, пропсы, состояния.

Понятие одностраничного приложения. Навигация в React. Хуки. Использование хуков состояния. Redux. Управление состоянием приложения. Чистые функции, редьюсеры, события.

Тема 7. Серверное программирование

Дистрибутив node.js - назначение, установка, основные приемы работы. Особенности серверного исполнения JavaScript. Менеджер пакетов NPM. Серверные фреймворки - назначение, примеры, обзор возможностей. Обработка запросов разных методов. Шаблонизация. Работа с базой данных. Возврат и чтение данных в формате JSON, XML. Создание простого API. Фреймворк Express.js - основные приемы работы, создание простого сервера. Серверный фреймворк FastAPI - основные характеристики, приемы работы, создание простого сервера.

Тема 8. ORM

Объектно-реляционное отображение - назначение, основные характеристики. Обзор различных СУБД. Реляционные и нереляционные базы данных. Определение модели данных. Контрактные классы. Подключение к БД. Выполнение операций CRUD. Миграции данных. ORM Sequelize для node.js. Основные приемы работы. ORM SQLAlchemy для Python. Основные приемы работы.

Тема 9. API

Прикладной интерфейс программирования в веб-разработке. Назначение, приемы. История развития. CGI-программирование. Архитектура RestAPI. Основные требования. GraphQL как альтернатива ReST. Реализация на разных фреймворках.

Тема 10. CMS

Системы управления содержимым - назначение, использование примеры. Создания контентного сайта на WordPress.

5.2. Учебно-тематический план

№ п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	Контактная работа - Аудиторная работа			Самостоятельна я работа	
			Общая, в т.ч.:	Лекции и	Семинары, практи		
1	Тема 1. Введение в веб-разработку	17	7	1	6	10	Опрос, решение задач
2	Тема 2. Основы HTML	34	14	4	10	20	Опрос, решение задач
3	Тема 3. CSS	34	14	4	10	20	Опрос, решение задач
4	Тема 4. JavaScript	75	15	5	10	60	Опрос, решение задач, защита проектной работы
5	Тема 5. Деплой статических сайтов	25	15	5	10	10	Опрос, решение задач
6	Тема 6. Фронтенд фреймворк React.js	25	15	5	10	10	Опрос, решение задач
7	Тема 7. Серверное программирование	25	15	5	10	10	Опрос, решение задач
8	Тема 8. ORM	21	11	1	10	10	Опрос, решение задач
9	Тема 9. API	16	6	1	5	10	Опрос, решение задач
10	Тема 10. CMS	16	6	1	5	10	Опрос, решение задач, защита проектной работы
В целом по дисциплине		288	118	32	86	170	Согласно учебному плану: две проектные работы
Итого в %		100	41	27	73	59	

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятия
Тема 1. Введение в веб-разработку	Установка и настройка WebStorm IDE. Установка Web-сервера	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 2. Основы HTML	Создание статических веб-страниц. Табличная и блочная верстка	Интерактивная форма: опрос, решение задач с

		последующим коллективным обсуждением их результатов
Тема 3. CSS	Стилизация веб-страниц. Создание альтернативных стилей. Адаптивная верстка. Использование фреймворков. Создание дизайна на основе Bootstrap	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 4. JavaScript	Создание динамических веб-страниц. Создание обработчиков событий	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 5. Деплой статичных сайтов	Развертывание статического сайта на облачном хостинге	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов, защита проектной работы
Тема 6. Фронтенд фреймворк React.js	Создание одностраничного веб-приложения. Создание приложения на React с навигацией. Продвинутое возможности React. Хуки	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 7. Серверное программирование	Создание простого серверного приложения на Express.js	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 8. ORM	Создание трехзвенного веб-приложения с использованием Sequelize	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 9. API	Создание ReST API приложения	Интерактивная форма: опрос, решение задач с последующим коллективным обсуждением их результатов
Тема 10. CMS	Создания контентного сайта на WordPress	Интерактивная форма: опрос, решение задач с последующим

		коллективным обсуждением их результатов
--	--	---

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Тема 1. Введение в веб-разработку	Языки и библиотеки веб-разработки, ландшафт профессий в области веб-программирования	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 2. Основы HTML	Продвинутые возможности языка HTML. Универсальные и специализированные атрибуты.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 3. CSS	Использование препроцессоров и создание сложных стилей. Svelte. Material.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 4. JavaScript	TypeScript	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, подготовка проектной работы
Тема 5. Деплой статичных сайтов	Развертывание сайта на собственном или выделенном хостинге.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 6. Фронтенд фреймворк React.js	Создание прогрессивного веб приложения	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение

Тема 7. Серверное программирование	Создание простого серверного приложения на FastAPI	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 8. ORM	Создание трехзвенного веб-приложения с использованием SQLAlchemy	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 9. API	Создание API приложения на основе GraphQL	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение
Тема 10. CMS	Деплой сложного веб-приложения.	Изучение материалов лекций и литературы. Работа с учебной литературой, поиск и анализ информации, содержащейся в Интернет-ресурсах. Разбор вопросов, отводимых на самостоятельное освоение, подготовка проектной работы

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерный перечень вопросов для подготовки к опросу

1. Основные понятие веб-технологий.
2. Установка и настройка WebStorm IDE. Установка Web-сервера
3. Создание статичных веб-страниц. Табличная и блочная верстка
4. Стилизация веб-страниц. Создание альтернативных стилей. Адаптивная верстка. Использование фреймворков. Создание дизайна на основе Bootstrap
5. HTML как язык разметки, его назначение, история развития.
6. Основы каскадных таблиц стилей.
7. Принцип разделения структуры и представления.
8. Способы подключения стилей к странице. Основы синтаксиса CSS.
9. Создание динамических веб-страниц. Создание обработчиков событий

10. Развертывание статического сайта на облачном хостинге
11. Создание одностраничного веб-приложения. Создание приложения на React с навигацией. Продвинутое возможности React.
12. Способы встраивания сценариев в веб-страницу.
13. Хостинг. Развертывание статических сайтов.
14. Создание простого серверного приложения на Express.js
15. Создание трехзвенного веб-приложения с использованием Sequelize
16. Создание REST API приложения
17. Прикладной интерфейс программирования в веб-разработке.
18. Системы управления содержимым.
19. Создания контентного сайта на WordPress

Примеры задач

Задача 1. Таймер обратного отсчёта. Реализуйте простой таймер обратного отсчёта: есть поле ввода для указания количества секунд и кнопка «Запустить». При нажатии на кнопку начинается отсчёт — значение в специальном `<div>` уменьшается каждую секунду. Когда таймер достигает нуля, выводится сообщение «Время вышло!». Добавьте кнопку «Остановить» для прерывания отсчёта.

Задача 2. Валидация формы регистрации. Создайте форму с полями «Имя», «Email» и «Пароль». При нажатии кнопки «Зарегистрироваться»: проверьте, что имя содержит не менее 3 символов; email соответствует стандартному формату (содержит @ и .); пароль не короче 6 символов.

При ошибках подсвечивайте соответствующие поля красным и выводите подсказки под ними. При успешной валидации — выводите сообщение «Регистрация успешна!».

Задача 3. Переключение темы оформления. Добавьте на страницу кнопку «Переключить тему». При нажатии: фон страницы меняется на чёрный, текст — на белый (тёмная тема); при повторном нажатии — возврат к светлой теме (белый фон, чёрный текст).

Сохраните выбор пользователя в localStorage, чтобы тема сохранялась при перезагрузке страницы.

Задача 4. Фильтрация списка товаров. Дан список товаров в виде `` с элементами ``, где каждый элемент содержит название и категорию (например, `<li data-category="электроника">Смартфон`). Добавьте выпадающий список `<select>` с категориями. При выборе категории в списке отображаются только товары из неё, остальные скрываются.

Задача 5. Динамическое добавление и удаление элементов. Есть поле ввода и кнопка «Добавить». При вводе текста и нажатии кнопки новый элемент `` добавляется в список `` под полем ввода. Каждый элемент списка должен иметь кнопку «Удалить», которая убирает соответствующий `` из DOM.

Задача 6. Калькулятор скидки. Создайте интерфейс с двумя полями ввода: «Цена» и «Процент скидки». При изменении любого из полей автоматически (без кнопки) пересчитывайте итоговую цену и выводите её в отдельный `<div>`. Формула: $\text{итоговая цена} = \text{цена} * (1 - \text{процент скидки} / 100)$.

Задача 7. Поиск по таблице. Дана таблица с данными (например, список пользователей с колонками «Имя», «Возраст», «Город»). Добавьте поле ввода «Поиск». При вводе текста в это поле фильтруйте строки таблицы — показывайте только те, где в любой колонке есть совпадение с введённым текстом. Регистр не должен учитываться.

Примерные задания проектной работы (семестр 5)

Задание 1. Микроблог: создать веб-приложение для коротких сообщений (до 280 символов). Реализовать: регистрацию и авторизацию пользователей; публикацию, редактирование и удаление постов; подписку на других пользователей; ленту новостей с постами подписчиков; поиск пользователей и постов.

Задание 2. Сервис для изучения иностранных слов: разработать

веб-приложение для запоминания лексики. Функционал: добавление новых слов с переводом и примерами; карточки для повторения с системой интервальных повторений; тесты на знание слов (выбор правильного перевода, ввод слова по памяти); статистика прогресса пользователя; возможность создавать тематические наборы слов.

Задание 3. Генератор резюме: сделать онлайн-конструктор резюме. Реализовать: заполнение формы с данными (ФИО, опыт работы, образование и т. д.); выбор шаблона оформления (минимум 3 варианта); предпросмотр резюме в браузере; экспорт в PDF; сохранение черновиков в локальном хранилище.

Задание 4. Трекер привычек: создать приложение для отслеживания ежедневных привычек. Функционал: добавление привычек с указанием цели (например, «пить 2 л воды в день»); отметка выполнения привычки за день; визуализация прогресса (графики, календарь с отметками); уведомления о необходимости выполнить привычку; статистика за неделю/месяц.

Задание 5. Сервис коротких ссылок: реализовать веб-сервис для сокращения URL-адресов. Функционал: вставка длинной ссылки и получение короткой; хранение истории созданных ссылок для авторизованных пользователей; настройка пользовательских псевдонимов для ссылок; статистика переходов по каждой ссылке (количество, дата, страна).

Задание 6. Виртуальная доска для совместной работы: разработать интерактивную доску для командной работы. Реализовать: рисование фигур и линий; добавление текста и изображений; создание стикеров и заметок; совместное редактирование в реальном времени; экспорт доски в PNG/PDF.

Задание 7. Планировщик бюджета: создать веб-приложение для учёта личных финансов. Функционал: добавление доходов и расходов с категориями (еда, транспорт и т. д.); установка бюджета на месяц по категориям; визуализация трат (круговые диаграммы, графики); импорт CSV-файлов из банковских выписок; отчёты за период.

Задание 8. Каталог фильмов и сериалов: разработать сайт-агрегатор

киноконента. Функционал: просмотр каталога фильмов с фильтрацией по жанру, году, рейтингу; страница фильма с описанием, трейлером, актёрским составом; добавление фильмов в «Избранное»; поиск по названию; интеграция с API для получения данных.

Задание 9. Онлайн-редактор кода: создать веб-редактор для написания и запуска кода в браузере. Реализовать: подсветку синтаксиса для нескольких языков; выполнение кода в песочнице; сохранение проектов в локальном хранилище; возможность делиться кодом по ссылке.

Задание 10. Система бронирования билетов (кино, концерты): разработать сервис для покупки билетов. Функционал: просмотр расписания сеансов/мероприятий; выбор мест на интерактивной схеме зала; бронирование и оплата билетов (имитация); личный кабинет с историей заказов; отправка билетов на email.

Задание 11. Интерактивный тест/квиз: создать веб-приложение для прохождения тестов. Функционал: создание тестов с вопросами и вариантами ответов (администратор); прохождение тестов пользователями с таймером; подсчёт баллов и выдача результата; рейтинг пользователей по баллам; адаптивные вопросы (следующий вопрос зависит от ответа на предыдущий).

Задание 12. Погода и прогноз: разработать погодный сервис с использованием API. Функционал: отображение текущей погоды по городу (температура, влажность, ветер и т. д.); прогноз на 5 дней с почасовым разбиением; автоматическое определение местоположения пользователя; поиск погоды по любому городу мира; виджет погоды для встраивания на другие сайты.

Примерные задания проектной работы (семестр 6)

Задание 1. Сервис для поиска попутчиков в поездки: создать веб-приложение, где пользователи могут искать попутчиков для междугородних поездок. Реализовать: регистрацию и профиль пользователя;

создание объявлений о поездке (маршрут, дата, время, количество свободных мест); поиск поездок по направлению и дате; отправку запросов на участие в поездке; чат между водителем и пассажирами.

Задание 2. Виртуальная галерея искусств: разработать онлайн-галерею для демонстрации произведений искусства. Функционал: каталог работ с фильтрацией по художнику, стилю, году создания; просмотр работ в высоком разрешении с возможностью увеличения; добавление работ в «Избранное»; комментарии и оценки работ; виртуальные «экскурсии» по залам галереи (360° панорамы).

Задание 3. Трекер спортивных тренировок: создать приложение для учёта спортивных достижений. Реализовать: добавление тренировок с указанием типа активности, длительности, интенсивности; отслеживание прогресса (графики веса, объёмов, результатов); создание индивидуальных планов тренировок; календарь с расписанием занятий; импорт данных из фитнес-браслетов (имитация API).

Задание 4. Система управления библиотекой: разработать веб-систему для автоматизации работы библиотеки. Функционал: учёт книг (добавление, редактирование, удаление); поиск книг по автору, названию, жанру; регистрация читателей и выдача книг; уведомления о сроках возврата; статистика по популярности книг и читателей.

Задание 5. Генератор резюме и сопроводительных писем: сделать онлайн-конструктор документов для трудоустройства. Реализовать: заполнение формы с личными данными, опытом работы, навыками; выбор шаблона оформления (минимум 4 варианта); предпросмотр документов в браузере; экспорт в PDF и DOCX; сохранение черновиков в облаке.

Задание 6. Платформа для обмена книгами: создать сервис, где пользователи могут обмениваться книгами. Функционал: добавление книг в каталог с описанием и фото; поиск книг по названию, автору, жанру; отправка запросов на обмен; рейтинг пользователей и отзывы о сделках; уведомления о новых предложениях.

Задание 7. Интерактивный квест-тренажёр: разработать веб-приложение для обучения через игровые сценарии. Реализовать: создание квестов с ветвлением сюжета; добавление заданий (выбор ответа, ввод текста, перетаскивание элементов); систему очков и достижений; статистику прохождения для администратора; адаптивные подсказки в зависимости от ошибок пользователя.

Задание 8. Сервис мониторинга цен: создать систему для отслеживания цен на товары в интернет-магазинах. Функционал: добавление URL товаров для отслеживания; автоматическое парсинг цен (имитация); графики изменения цен во времени; уведомления при снижении цены ниже заданного порога; сравнение цен на один товар в разных магазинах.

Задание 9. Виртуальный гардероб: разработать приложение для планирования образов. Реализовать: добавление вещей в гардероб (фото, категория, цвет, сезон); составление комплектов из имеющихся вещей; фильтрация по погоде и событиям (работа, вечеринка и т. д.); календарь с планированием образов на дни; рекомендации на основе погоды.

Задание 10. Система учёта домашних расходов: создать веб-приложение для семейного бюджета. Функционал: добавление доходов и расходов с категориями; совместный доступ для нескольких пользователей (семья); визуализация трат (круговые диаграммы, графики); установка лимитов по категориям; отчёты за месяц/год с экспортом в CSV.

Задание 11. Платформа для репетиторов: разработать сервис для поиска преподавателей. Реализовать: профили репетиторов с рейтингом и отзывами; поиск по предмету, цене, формату занятий (онлайн/очно); бронирование уроков с календарем занятости; встроенный видеочат для занятий; систему оплаты и отзывов после урока.

Задание 12. Генератор интерактивных викторин: создать инструмент для создания онлайн-викторин. Функционал: конструктор вопросов (текст, изображения, аудио); настройка таймера и системы баллов; запуск викторины в режиме реального времени; отображение рейтинга игроков во время игры;

аналитика результатов для создателя викторины.

Задание 13. Электронная библиотека: создать онлайн-библиотеку с книгами в формате PDF/EPUB. Реализовать: каталог книг с фильтрацией по жанру, автору, году; просмотр книг в браузере; добавление книг в «Читальный список»; поиск по названиям и авторам; личный кабинет для отслеживания прогресса чтения.

Задание 14. Сервис для поиска попутчиков: разработать платформу для совместных поездок. Функционал: регистрация водителей и пассажиров; публикация объявлений о поездке (маршрут, дата, стоимость); поиск поездок по направлению и дате; бронирование мест пассажирами; чат между водителем и пассажирами.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях кафедры.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов достижения и планируемых результатов обучения по дисциплине содержится в разделе 2 «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Примерные вопросы для подготовки к зачету (семестр 5)

1. 1. Что такое HTML и какие основные элементы он содержит?

2. Какие атрибуты могут быть присвоены HTML-элементам?
3. Какие существуют типы списков в HTML?
4. Какие существуют типы таблиц в HTML?
5. Какие существуют типы форм в HTML?
6. Что такое CSS и какие стили можно задавать с помощью него?
7. Какие селекторы используются в CSS?
8. Что такое каскадность и специфичность в CSS?
9. Какие существуют типы единиц измерения в CSS?
10. Как задать фоновый цвет и изображение для элемента в CSS?
11. Как задать границы и рамки для элемента в CSS?
12. Как задать отступы и выравнивание для элемента в CSS?
13. Что такое JavaScript и какие функции он может выполнять на странице?
14. Как объявить переменную в JavaScript?
15. Какие существуют операторы в JavaScript и как они работают?
16. Какие существуют условные операторы в JavaScript?
17. Какие существуют циклы в JavaScript и как они работают?
18. Как объявить функцию в JavaScript и как она может быть вызвана?
19. Как передать параметры в функцию в JavaScript?
20. Как использовать объекты в JavaScript и какие свойства они могут иметь?
21. Как создать массив в JavaScript и как его использовать?
22. Какие существуют методы массивов в JavaScript?
23. Как работает обработка событий в JavaScript?
24. Как создать элемент на странице с помощью JavaScript?
25. Как изменять стили элементов на странице с помощью JavaScript?
26. Как отправить данные на сервер с помощью JavaScript?
27. Что такое AJAX и как он используется в JavaScript?
28. Как выполнить асинхронный запрос с помощью AJAX?
29. Что такое JSON и как он используется в JavaScript?
30. Как парсить JSON-данные в JavaScript?

Примерные вопросы для подготовки к экзамену (семестр 6)

1. Понятие веб-технологий. Чем веб отличается от других сетевых служб?
2. Основные сетевые протоколы, используемые в вебе.
3. История развития веб-технологий.
4. Клиентская часть веб-приложений. Назначение, технологии, схема работы.
5. Серверная часть веб-приложений. Назначение, технологии, схема работы.
6. Необходимое программное обеспечение для работы веб-приложений.
7. Необходимое программное обеспечение для веб-разработки.
8. URL и URI. Понятие, различия, структура, примеры, использование.
9. Схема соединения по протоколу HTTP.
10. HTTP-запросы. Структура, примеры, использование, методы.
11. HTTP-ответы. Структура, примеры, поля ответа, использование.
12. Методы HTTP. Использование в современных API.
13. Общая характеристика языка HTML. Назначение, структура. Понятие тега. Виды тегов.
14. HTML5. Особенности, примеры тегов, назначение. Понятие семантических элементов.
15. HTML. Теги заголовка веб-страницы.
16. HTML. Теги форматирования текста.
17. HTML. Встроенные и блочные элементы.
18. HTML. Теги форматирования списков.
19. HTML. Таблицы.
20. HTML. Гиперссылки и якоря.
21. HTML. Универсальные атрибуты тегов.
22. HTML. Изображения, рисунки и мультимедиа.
23. HTML. Формы и поля ввода.

24. HTML. DOM. Дерево элементов. Основные понятия.
25. CSS. Понятие таблицы стилей, назначение, общая характеристика языка.
26. Способы подключения стилей к веб-странице. Как выбрать правильный?
27. CSS. Селекторы. Виды селекторов.
28. CSS. Задание цвета элементов. Способы задания цветов.
29. CSS. Задание параметров шрифтов.
30. CSS. Задание параметров границ элементов.
31. CSS. Задание размеров блочных элементов.
32. CSS. Задание внутренних и внешних отступов блочных элементов.
33. CSS. Единицы измерения размеров. Преимущества и недостатки.
34. CSS. Выравнивание и позиционирование блочных элементов.
35. CSS. Способы позиционирования блочных элементов.
36. CSS. Псевдоклассы и псевдоэлементы.
37. CSS. Каскадность, наследование, приоритеты стилей.
38. CSS. Адаптивная верстка.
39. CSS. Flexbox. Основные понятия, структура, примеры свойств.
40. CSS. Grid. Основные понятия, структура, примеры свойств.
41. CSS. Препроцессоры: основные понятия, назначение, примеры кода на SASS или LESS.
42. Понятие CSS-фреймворков. Примеры, сравнение, назначение.
43. Bootstrap. назначение, общая характеристика, примеры работы.
44. Общая характеристика языка программирования JavaScript.
45. Клиентский JavaScript. Схема работы, назначение, примеры, способы связывания.
46. JavaScript. Инструкции, комментарии, объявление переменных.
47. JavaScript. Простые типы данных.
48. JavaScript. Базовые математические и логические операторы.
49. JavaScript. Операции со строками.

50. JavaScript. Массивы. Создание, использование, основные операции.
51. JavaScript. Условные операторы.
52. JavaScript. Операторы циклов.
53. JavaScript. Функции.
54. JavaScript. Обработка исключений.
55. JavaScript. Литерация объектов. JSON.
56. JavaScript. Классы в ES6.
57. JavaScript. Функции высших порядков.
58. JavaScript. Наследование.
59. JavaScript. Деструктуризация массивов и объектов.
60. JavaScript. Стрелочные функции.
61. JavaScript. Промисы.
62. DOM. Браузерное окружение в JavaScript.
63. DOM. Навигация. Способы нахождения элементов. Поиск по дереву.
64. DOM. Свойства узлов. Изменение свойств элементов.
65. DOM. Браузерные события. Обработчики событий.
66. DOM. Методы обхода дерева элементов.
67. DOM. Программное создание нового элемента и добавление его в дерево.
68. Понятие фронтенд фреймворков. Назначение, примеры, использование.
69. React.js. Общая характеристика. Установка, запуск.
70. React.js. Понятие компонента. Способы задания.
71. React.js. Состояние компонента. Назначение, использование.
72. React.js. JSX.
73. React.js. Структура проекта. Основные файлы.
74. React.js. Передача параметров как свойств.
75. React.js. Задание стилей.
76. React.js. Маршрутизация.
77. React.js. Хуки.
78. React.js. Redux.

79. Node.js. Общая характеристика, назначение, использование, установка.
80. Node.js. Структура проекта на node. package.json.
81. Node.js. Установка и импорт модулей. Зависимости.
82. Node.js. Семантическое версионирование.
83. Технологии разработки серверных веб-приложений.
84. Серверные фреймворки. Общая характеристика, назначение, примеры, сравнение.
85. Express.js. Основные понятия. Установка, запуск, простейшее приложение.
86. Express.js. Статический сервер.
87. Express.js. Возврат JSON.
88. Express.js. Middleware.
89. REST API. Основные понятия, назначение, использование.
90. Express.js. Создание RESTfull API.
91. Express.js. Использование роутера.
92. Express.js. Шаблонизатор handlebars.
93. Express.js. Работа с базой данных.
94. MongoDB. Общая характеристика, сравнение.
95. Express.js. CRUD операции с MongoDB.
96. ORM. Понятие, назначение, использование, примеры.
97. Библиотека Sequelize. Общая характеристика, назначение, установка, подключение.
98. Библиотека Sequelize. Определение модели данных.
99. Библиотека Sequelize. Выполнение запросов к базе данных.
100. Библиотека Sequelize. Геттеры, сеттеры и виртуальные атрибуты.

Примеры задач для подготовки к экзамену (семестр 6)

Задача 1. Даны N данных-чисел с классом `.number` и кнопка «Вычислить». По нажатию на кнопку: получите числа из всех инпутов; найдите среди них максимальное и минимальное значение;выведите

результаты в два отдельных абзаца с `id="max"` и `id="min"`; если в каком-то поле введено не число, игнорируйте его и подсветите красным.

Задача 2. Дана таблица с оценками учеников (столбцы: «Ученик», «Математика», «Физика», «Химия»). Реализуйте: при наведении на ячейку с оценкой она подсвечивается жёлтым; по клику на ячейку оценка увеличивается на 1 (максимум 5); под таблицей кнопка «Средний балл» — при нажатии в абзаце ниже выводится средний балл по каждому предмету; кнопка «Сбросить все оценки» возвращает все значения к исходным.

Задача 3. Есть поле ввода и кнопка «Добавить задачу». По нажатию: новый элемент `` с текстом из поля добавляется в список `<ul id="tasks">`; каждая задача имеет кнопку «Готово» справа и кнопку «Удалить» (крестик); при нажатии «Готово» текст задачи зачёркивается, кнопка меняется на «Вернуть»; при нажатии «Вернуть» зачёркивание снимается; под списком выводите количество выполненных задач.

Задача 4. Дан список товаров в виде карточек (`<div class="product">`), у каждой есть атрибуты `data-category` и `data-price`. Реализуйте: выпадающий список для фильтрации по категории; переключатель сортировки: «По цене (возр.)» / «По цене (убыв.)»; поле поиска, фильтрующее товары по названию (без учёта регистра); все изменения применяются мгновенно без перезагрузки страницы.

Задача 5. Создайте таймер с: полем ввода для установки времени в секундах; кнопками «Старт», «Пауза», «Сброс»; отображением оставшегося времени в формате MM:SS в `<div id="timer">`; звуковым сигналом при достижении нуля; блокировкой поля ввода во время работы таймера.

Задача 6. Реализуйте опрос с 3–5 вариантами ответа: каждый вариант — кнопка с названием и счётчиком голосов; при клике на кнопку счётчик увеличивается на 1; под кнопками постройте столбчатую диаграмму (используйте `<div>` с разной высотой), показывающую процент голосов за каждый вариант; запрет на повторное голосование (сохраняйте выбор в `localStorage`).

Задача 7. Дан список `<ul id="sortable">` с элементами ``. Реализуйте: возможность перетаскивать элементы списка для изменения их порядка; визуальную обратную связь: полупрозрачная копия элемента при перетаскивании, подсветка места вставки; после завершения перетаскивания обновите порядок элементов; кнопку «Сохранить порядок», которая записывает текущую последовательность.

Задача 8. Создайте поле ввода «Город». При вводе текста: показывается выпадающий список подходящих городов из массива `cities = ["Москва", "Санкт-Петербург", ...]`; совпадения подсвечиваются жирным; при выборе города из списка он вставляется в поле ввода, а список скрывается; если список не скрыт, его можно закрыть нажатием Esc или кликом вне поля.

Задача 9. Постройте линейный график на основе массива данных [5, 10, 15, 20, 25]: используйте `<canvas>` для отрисовки осей и линии; при наведении на точку графика показывайте всплывающую подсказку с координатами (x, y); добавьте кнопку «Добавить случайную точку», которая добавляет новое значение в массив и перерисовывает график; график масштабируется под размер контейнера.

Задача 10. Разработайте простой интерфейс чата: поле ввода сообщения и кнопка «Отправить»; область сообщений, где отображаются отправленные тексты (каждое сообщение — `<div class="message">`); сообщения пользователя выравниваются вправо и имеют синий фон, «ответы» (имитация) — влево и серый фон; имитация ответов: через 1–2 секунды после отправки добавляется случайный ответ; кнопка «Очистить историю» удаляет все сообщения и очищает `localStorage`.

Пример экзаменационного билета

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Кафедра: **Корпоративные инфокоммуникационные системы**
Дисциплина: **Веб-разработка**
Филиал: **Владикавказский**; Форма обучения: **Очная**
Семестр: 6 Направление: **09.03.04 Программная инженерия**
Профиль: **Технологии разработки программного обеспечения**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. **Теория.** Библиотека Sequelize. Выполнение запросов к базе данных. **(15 баллов)**
2. **Теория.** JavaScript. Литерация объектов. JSON **(15 баллов)**
3. **Задача.** Дан список `<ul id="sortable">` с элементами ``. Реализуйте: возможность перетаскивать элементы списка для изменения их порядка; визуальную обратную связь: полупрозрачная копия элемента при перетаскивании, подсветка места вставки; после завершения перетаскивания обновите порядок элементов в DOM; кнопку «Сохранить порядок», которая записывает текущую последовательность в localStorage.. **(30 баллов)**

Подготовил: _____ М.А Ковалева

На основе перечня теоретических вопросов и заданий, утвержденного на заседании кафедры «Корпоративные инфокоммуникационные системы» протокол № ____ от _____.2026 г.

Утверждаю:
Заведующий кафедрой _____ М.А Ковалева
Дата _____.2026г.

**Примеры оценочных средств для проверки индикаторов
достижения компетенций, формируемых дисциплиной**

Код и наименование компетенции	Наименование индикатора достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-5 Способность проектировать и реализовывать интеллектуальные информационные системы	1. Демонстрирует знания основных методов машинного обучения и интеллектуального анализа данных, применяет готовые инструменты для создания интеллектуальных алгоритмов.	Знать: основные методы машинного обучения и интеллектуального анализа данных Уметь: применяет готовые инструменты для создания интеллектуальных алгоритмов	Вопросы: 1. Понятие веб-технологий. Чем веб отличается от других сетевых служб? 2. Понятие фронтенд фреймворков. Назначение, примеры, использование. Задача Реализуйте опрос с 3–5 вариантами ответа: каждый вариант — кнопка с названием и счётчиком голосов; при клике на кнопку счётчик увеличивается на 1; под кнопками постройте столбчатую диаграмму (используйте <div> с разной высотой), показывающую процент голосов за каждый вариант; запрет на повторное голосование (сохраняйте выбор в localStorage).
	2. Понимает особенности интеллектуальных информационных систем в части операций разработки, развертывания и сопровождения.	Знать: особенности интеллектуальных информационных систем Уметь: применяет знания в части операций разработки, развертывания и сопровождения.	Вопросы: 1. REST API. Основные понятия, назначение, использование. 2. Технологии разработки серверных веб-приложений. Задача Даны N данных-чисел с классом .number и кнопка «Вычислить». По нажатию на кнопку:

			получите числа из всех инпутов; найдите среди них максимальное и минимальное значение; выведите результаты в два отдельных абзаца с id="max" и id="min"; если в каком то поле введено не число, игнорируйте его и подсветите красным.
	3.Адаптирует практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем.	<p>Знать: как адаптировать практики создания программных продуктов</p> <p>Уметь: адаптировать практики создания программных продуктов, в том числе командные, для интеллектуальных информационных систем</p>	<p>Вопросы:</p> <ol style="list-style-type: none"> 1. Браузерное окружение в JavaScript. 2. Клиентский JavaScript. Схема работы, назначение, примеры, способы связывания <p>Задача</p> <p>Дана таблица с оценками учеников (столбцы: «Ученик», «Математика», «Физика», «Химия»). Реализуйте: при наведении на ячейку с оценкой она подсвечивается жёлтым; по клику на ячейку оценка увеличивается на 1 (максимум 5); под таблицей кнопка «Средний балл» — при нажатии в абзаце ниже выводится средний балл по каждому предмету; кнопка «Сбросить все оценки» возвращает все значения к исходным</p>
	4. Организует сбор и подготовку данных для систем машинного обучения, в том числе потоковых, онлайн обучения.	<p>Знать: как организовывать сбор и подготовку данных для систем машинного обучения</p> <p>Уметь: организовывать сбор и подготовку данных для систем машинного</p>	<p>Вопросы:</p> <ol style="list-style-type: none"> 1. Библиотека Sequelize. Выполнение запросов к базе данных. 2. Общая характеристика языка HTML. Назначение, структура. Понятие тега. Виды тегов. 2. DOM. Навигация. Способы нахождения

		обучения, в том числе потоковых, онлайн обучения.	<p>элементов. Поиск по дереву.</p> <p>Задача</p> <p>Есть поле ввода, и кнопка «Добавить задачу». По нажатию: новый элемент <code></code> с текстом из поля добавляется в список <code><ul id="tasks"></code>; каждая задача имеет кнопку «Готово» справа и кнопку «Удалить» (крестик); при нажатии «Готово» текст задачи зачёркивается, кнопка меняется на «Вернуть»; при нажатии «Вернуть» зачёркивание снимается; под списком выводите количество выполненных задач.</p>
ПКП-6 Способность вести разработку программных систем в команде, вести эффективную коммуникацию	1. Придерживается рекомендованного в команде стиля написания кода, проводит рефакторинг написанного кода, в том числе, другими членами команды, проводит кодревью.	<p>Знать: как придерживаться рекомендованного в команде стиля написания кода</p> <p>Уметь: проводить рефакторинг написанного кода, в том числе, другими членами команды, проводить код-ревью</p>	<p>Вопросы:</p> <p>1. CSS. Понятие таблицы стилей, назначение, общая характеристика языка.</p> <p>2. Способы подключения стилей к веб-странице. Как выбрать правильный?</p> <p>Задача</p> <p>Дан список товаров в виде карточек (<code><div class="product"></code>), у каждой есть атрибуты <code>data-category</code> и <code>data-price</code>. Реализуйте: выпадающий список для фильтрации по категории; переключатель сортировки: «По цене (возр.)» / «По цене (убыв.)»; поле поиска, фильтрующее товары по названию (без учёта регистра); все изменения применяются мгновенно</p>

			без перезагрузки страницы.
2. Документирует код, архитектуру, дизайн, требования к программной системе с использованием принятой в команде системы технических стандартов.	Знать: о документации кода, архитектуру, дизайн, требования к существующим программным системам Уметь: документировать код, архитектуру, дизайн, требования к программной системе с использованием принятой в команде системы технических стандартов	Вопросы: 1. Методы HTTP. Использование в современных API. 2. REST API. Основные понятия, назначение, использование Задача Создайте таймер с: полем ввода для установки времени в секундах; кнопками «Старт», «Пауза», «Сброс»; отображением оставшегося времени в формате MM:SS в <div id="timer">; звуковым сигналом при достижении нуля; блокировкой поля ввода во время работы таймера.	
3. Использует инструментальные средства командной разработки программного обеспечения, контроля версий, отслеживания ошибок, планирования процессов разработки в зависимости от принятой в команде методологии разработки.	Знать: инструментальные средства командной разработки программного обеспечения, контроля версий, отслеживания ошибок, планирования процессов разработки в зависимости от принятой в команде методологии разработки Уметь: использовать инструментальные средства командной разработки	Вопросы: 1. REST API. Основные понятия, назначение, использование. 2. Node.js. Семантическое версионирование. Задача Дан список <ul id="sortable"> с элементами . Реализуйте: возможность перетаскивать элементы списка для изменения их порядка; визуальную обратную связь: полупрозрачная копия элемента при перетаскивании, подсветка места вставки; после завершения перетаскивания	

		программного обеспечения	обновите порядок элементов; кнопку «Сохранить порядок», которая записывает текущую последовательность.
4. Выстраивает эффективную двустороннюю коммуникацию с нетехническими специалистами по предмету разработки предметной области в целом.	Знать: как выстраивать эффективную двустороннюю коммуникацию с нетехническими специалистами Уметь: выстраивать эффективную двустороннюю коммуникацию с нетехническими специалистами	Вопросы: 1. React.js. Структура проекта. Основные файлы. 2. React.js. Понятие компонента. Способы задания Задача Создайте поле ввода «Город». При вводе текста: показывается выпадающий список подходящих городов из массива cities = ["Москва", "Санкт-Петербург", ...]; совпадения подсвечиваются жирным; при выборе города из списка он вставляется в поле ввода, а список скрывается; если список не скрыт, его можно закрыть нажатием Esc или кликом вне поля	
5. Коммуницирует задачи разработки между членами команды, оценивает трудоемкость и сложность выполнения задач разработки, ставит задачи и контролирует их выполнение, в том числе в письменной формализованной форме.	Знать: как ставить задачи разработки между членами команды Уметь: оценивать трудоемкость и сложность выполнения задач разработки, ставить задачи и контролирует их выполнение, в том числе в письменной формализованной форме.	Вопросы: 1. Структура проекта на node. package.json. 2. Node.js. Установка и импорт модулей. Зависимости. 2. Базовые принципы объектно-ориентированного Задача Разработайте простой интерфейс чата: поле ввода сообщения и кнопка «Отправить»; область сообщений, где отображаются отправленные тексты	

			(каждое сообщение — <code><div class="message"></code>); сообщения пользователя выравниваются вправо и имеют синий фон, «ответы» (имитация) — влево и серый фон; имитация ответов: через 1–2 секунды после отправки добавляется случайный ответ; кнопка «Очистить историю» удаляет все сообщения и очищает localStorage.
--	--	--	---

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература

1. Полуэктова, Н. Р. Разработка веб-приложений: учебник для вузов / Н. Р. Полуэктова. — Москва: Юрайт, 2026. — 204 с. — ЭБС Юрайт. — URL: <https://urait.ru/bcode/588458>. — Текст: электронный.

2. Абдуллаева, О. С. Информационные технологии. Практикум: учебное пособие / О. С. Абдуллаева. — Москва: Русайнс, 2024. — 119 с. — ЭБС BOOK.ru. — URL: <https://book.ru/book/952937>. — Текст: электронный.

Дополнительная литература

3. Брокшмидт, К. Введение в разработку приложений для Windows 8 с использованием HTML, CSS и JavaScript / К. Брокшмидт. — 2-е изд., испр. — Москва: НОУ «ИНТУИТ», 2016. — 460 с. — ЭБС Университетская библиотека ONLINE. - URL: <http://biblioclub.ru/index.php?page=book&id=428973>. - Текст: электронный.

4. Гуськова, О. И. Объектно ориентированное программирование в Java: учебное пособие / О. И. Гуськова. — Москва: МПГУ, 2018. — 240 с. — ЭБС ZNANIUM. - URL: <https://znanium.com/catalog/product/1020593>. — Текст: электронный.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
2. Электронно-библиотечная система Znanium <http://www.znanium.com>
3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.biblio-online.ru>
5. Электронная библиотека издательского дома «Гребенников» <https://grebennikon.ru>
6. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com>

10. Методические указания для обучающихся по освоению дисциплины

Методика освоения дисциплины предусматривает подготовку обучающихся к лекциям, семинарам и практическим занятиям, выполнение студентами самостоятельной внеаудиторной работы.

Рекомендации по подготовке к лекционным занятиям.

Для наиболее полного освоения дисциплины студентам необходимо:

- перед каждой лекцией просматривать рабочую программу дисциплины, ее основные вопросы и рекомендуемую литературу. Это позволит сэкономить время на записывание основных вопросов темы;
- перед очередной лекцией просматривать материалы предыдущих, чтобы освоение материала не оставляло пробелов.

Рекомендации по подготовке к семинарам, практическим занятиям.

Студентам следует:

- проработать теоретический материал к занятию по рекомендованным литературным источникам и лекциям;

- использовать при подготовке к занятию нормативно-правовые документы, научные публикации, информационный материал, рекомендуемый преподавателем;
- перед занятиями задать вопросы по невыясненным в ходе самостоятельной подготовки темам или отдельным положениям темы;
- в ходе занятия давать четкие и исчерпывающие ответы на вопросы;
- на занятии демонстрировать понимание обсуждаемых тем и вопросов.

Студентам, пропустившим занятия по различным причинам, необходимо перед очередным занятием отработать пропущенный материал, подготовив его самостоятельно.

Методические рекомендации по выполнению различных форм самостоятельной работы

Студентам при организации самостоятельной работы следует руководствоваться Приказом Финансового университета № 1040/о от 11.05.2021г. «Об утверждении методических рекомендаций по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете».

Самостоятельная работа содержит в себе различные виды и формы работ. Самостоятельная работа студентов включает в себя выполнение различного рода заданий, которые ориентированы на более глубокое усвоение материала изучаемой дисциплины. По теме учебной дисциплины студентам предлагается перечень заданий для самостоятельной работы.

В ходе изучения дисциплины предусмотрены следующие формы самостоятельной работы:

- подготовка к опросу;
- разбор вопросов, отводимых на самостоятельное освоение,
- решение задач;
- выполнение проектных работ;

- подготовка к зачету и экзамену.

К выполнению заданий для самостоятельной работы предъявляются следующие требования: задания должны выполняться самостоятельно и представляться в установленный срок, а также должны соответствовать установленным требованиям по оформлению.

Студентам следует:

- руководствоваться графиком самостоятельной работы, определенным РПД;
- выполнять все плановые задания, выдаваемые преподавателем для самостоятельного выполнения, разбирать на занятиях и консультациях неясные вопросы;
- прорабатывать соответствующие теоретические и практические разделы дисциплины, фиксируя неясные фрагменты для их обсуждения на консультации.

Методические рекомендации для обучающихся по выполнению проектной работы

Проектная работа является обязательной формой внеаудиторной самостоятельной работы студентов по дисциплине.

Целью проектной работы является развитие у студентов способности прогнозировать, проектировать, моделировать, формирование учебно-исследовательских навыков, закрепление умений самостоятельно работать с различными источниками информации; проверка сформированности компетенций.

Проектная работа может выполняться как индивидуально, так и в составе группы. Количество групп и их численный состав определяет преподаватель, ведущий семинарские занятия.

Заказчиками выполнения проекта могут являться представители работодателей. В этом случае проектная работа выполняется исходя из потребностей заказчика.

Выполнение проекта предполагает:

- диагностику ситуации (проблематизация, целеполагание, конкретизация цели, форматирование проекта);

- проектирование (уточнение цели, функций, задач и плана работы; теоретическое моделирование методов и средств решения задач; детальная проработка этапов решения конкретных задач; пошаговое выполнение запланированных проектных действий; систематизация и обобщение полученных результатов, конструирование предполагаемого результата, пошаговое выполнение проектных действий);

- рефлексия (выяснение соответствия полученного результата замыслу; определение качества полученного продукта; перспективы его развития и использования);

- фиксация результатов в виде исполненного проекта.

Проектная работа состоит из нескольких частей. Состав проектной работы и очередность размещения отдельных частей:

- титульный лист;
- основная часть;
- список использованных источников;
- приложения (при наличии).

Титульный лист является первой страницей проектной работы и заполняется по определенным правилам.

Основная часть выполняется согласно заданию преподавателя или исходя из потребностей заказчика.

В список использованных источников включаются названия законодательных актов, нормативных документов, книг, статей, учебных пособий и т. п., которые, так или иначе, использовались студентом при выполнении работы.

Иллюстративный материал (схемы, диаграммы, рисунки, таблицы и др.) встраивается в текст работы или выносится в Приложения. В Приложения выносятся вспомогательные материалы, которые не содержат основную информацию, либо материалы, которые сложно разместить по тексту работы

(большие схемы, таблицы, графические материалы, расчетные справочные данные, образцы первичных документов и т.п.). Непременным условием включения данных материалов в приложение является ссылка на них в тексте работы.

Требования к оформлению проектной работы.

Проектная работа выполняется на компьютере на одной стороне белой бумаги формата А4 (210х297 мм). Размер шрифта -13 или 14, междустрочный интервал – одинарный или полуторный.

Размеры полей: левое - 30 мм, правое - 15 мм, верхнее - 20 мм, нижнее- 20 мм. Отступ первой строки абзаца - 1,25. Нумерация страниц – внизу в центре.

При написании допускаются только общепринятые сокращения (например, тыс. руб.).

Общий объем проектной работы составляет не более 10 страниц, не включая таблицы, графики и т.п. (при наличии), а также приложения (при наличии).

В тексте обязательны ссылки на литературные источники, лучше всего постраничные.

Законченная проектная работа, содержащая все требуемые элементы оформления, вставленная в папку (или файл) и скрепленная по левому краю, сдается на кафедру или непосредственно руководителю проектной работы – преподавателю; ведущему семинарские (практические) занятия по дисциплине. Он осуществляет проверку проектной работы, а также оказывает помощь при подготовке к ее защите.

Проектная работа защищается в назначенные сроки. Защита проектной работы проводится до начала сессии (в крайнем случае, до начала экзамена по соответствующему предмету). При защите студент кратко излагает основные положения работы, последовательность ее выполнения, свои предложения.

При защите проектной работы студент должен свободно ориентироваться в изложенном материале работы; ответить на все замечания

преподавателя; уметь отвечать на вопросы преподавателя по проектной работе.

Оценка проектных работ студентов проводится в процессе текущего контроля успеваемости.

Критерии оценки проектной работы

Оценка «отлично» (5-6 баллов) выставляется студенту, если проектная работа отличается творческим (креативным) подходом, собственным оригинальным отношением автора к идее проекта; содержит полную диагностику ситуации, а также теоретическое моделирование методов и детальную проработку этапов решения конкретных задач; в работе сделаны необходимые выводы, намечены перспективы использования проекта, спланированы действия по его продвижению; работа отличается грамотным оформлением в точном соответствии с установленными правилами, с соблюдением логической последовательности изложения материала; студент в работе выдвигает новые идеи и трактовки, демонстрирует способность анализировать материал; на дополнительные вопросы при защите проектной работы даны полные ответы.

Оценка «хорошо» (3-4 балла) выставляется студенту, если проектная работа содержит достаточно полную диагностику ситуации, а также теоретическое моделирование методов и этапов решения конкретных задач; в работе сделаны выводы, намечены перспективы использования проекта; работа оформлена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно или по требованию преподавателя; в работе соблюдена логическая последовательность изложения материала; студент в работе демонстрирует творческие способности и хорошую способность анализировать материал. На дополнительные вопросы при защите проектной работы даны не совсем полные ответы.

Оценка «удовлетворительно» (1-2 балла) выставляется студенту, если проектная работа содержит отдельные элементы моделирования методов и этапов решения конкретных задач; в работе сделаны выводы, намечены

перспективы использования проекта; работа выполнена и оформлена правильно, но в ней допущены 1-2 погрешности или одна грубая ошибка; в работе соблюдена логическая последовательность изложения материала; студент в работе демонстрирует удовлетворительную способность анализировать материал; допущены ошибки при ответе на дополнительные вопросы при защите проектной работы.

Оценки «неудовлетворительно» (0 баллов) заслуживает студент, если в работе отсутствуют элементы моделирования; студент в работе не проявил способность анализировать, прогнозировать и проектировать; в работе отсутствует логическая последовательность изложения материала, допущены грубые ошибки, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

При оценивании проектной работы на «неудовлетворительно» работа должна быть переделана (исправлена) в соответствии с полученными замечаниями, сдана на проверку заново и защищена не позднее срока окончания ее приёма и защиты.

Оценка результатов текущего контроля успеваемости и промежуточной аттестации обучающихся осуществляется в соответствии с Балльно-рейтинговой системой Финансового университета (Приказ Финансового университета № 2187/о от 01.10.2024 г. «Об утверждении Положения о проведении текущего контроля успеваемости и промежуточной аттестации студентов, обучающихся по образовательным программам высшего образования в Финансовом университете»).

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

11.1 Комплект лицензионного программного обеспечения

1) Антивирусная защита Kaspersky Security для виртуальных и облачных сред;

2) Windows, Microsoft Office или Astra Linux, Libre Office.

11.2 Современные профессиональные базы данных, и информационные справочные системы

1. Информационно-правовая система «Гарант»: <https://www.garant.ru>
2. Большая Российская энциклопедия: <https://bigenc.ru/>
3. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>.

11.3 Сертифицированные программные и аппаратные средства защиты информации

Не используются

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Учебная аудитория для проведения учебных занятий, предусмотренных программой магистратуры, оснащенная оборудованием и техническими средствами обучения

Аудитория № 45

Специализированная мебель:

Стол (учительский) – 1 шт.

Стол компьютерный – 1 шт.

Стол (студенческий) двухместный – 13 шт.

Стулья – 27 шт.

Доска меловая – 1 шт.

Технические средства обучения:

Компьютер в сборе – 1 шт.

Доска интерактивная – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения

Аудитория № 47

Специализированная мебель:

Стол компьютерный – 20 шт.

Стол (двухместный) – 7 шт.

Стул – 34 шт.

Шкаф – 1 шт.

Технические средства обучения:

Компьютер в сборе – 20 шт.

Мультимедиа-проектор – 1 шт.

Экран настенный – 1 шт.

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду Финансового университета

Помещение для самостоятельной работы обучающихся:

Кабинет № 55. Читальный зал:

Специализированная мебель:

Стол – 20 шт.

Стул – 40 шт.

Шкаф для книг – 4 шт.

Стеллаж книжный – 13 шт.

Стеллаж выставочный – 4 шт.

Технические средства обучения:

Компьютер в сборе – 6 шт.

Телевизор – 1 шт.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Финансового университета