

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультета информационных технологий и анализа больших данных**

УТВЕРЖДАЮ

Проректор по учебной
и методической работе

_____ Е.А. Каменева

22.04.2025 г.

Шаталова А.Ю.

Основы веб-разработки

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:
09.03.03 - Прикладная информатика,
ОП «Прикладные информационные системы в экономике и финансах»

*Рекомендовано Ученым советом
Факультета информационных технологий и анализа больших данных
(протокол № 54 от 15 апреля 2025 г.)*

*Одобрено советом Кафедры информационных технологий
(протокол № 2 от 09 апреля 2025 г.)*

Москва 2025

Содержание

1. Наименование дисциплины.....	2
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.....	2
3. Место дисциплины в структуре образовательной программы.....	4
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	4
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.....	12
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	12
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	12
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю.....	13
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	15
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.....	23
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	24
10. Методические указания для обучающихся по освоению дисциплины....	26
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем.....	26
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	27

1. Наименование дисциплины

«Основы веб-разработки».

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-3	Способен наладить и администрировать процесс построения и функционирования отказоустойчивого программного обеспечения информационных систем цифровой экономики, в том числе непрерывную поставку и развертывание программного кода	1. Применяет основные принципы организации командной работы разработчиков над проектом, процедуры, подходы и методы оптимизации и автоматизации процедур тестирования, доставки кода и запуска приложений на серверах	Знать: Основные принципы Agile и Scrum методологий разработки. Методы организации командной работы (например, Kanban, парное программирование). Роли и обязанности в команде разработчиков. Принципы работы с системами контроля версий (например, Git, GitLab, GitHub). Концепции CI/CD (Continuous Integration/Continuous Delivery/Deployment). Методы автоматизации тестирования (модульное, интеграционное, системное). Инструменты автоматизации сборки, тестирования и развертывания (например, Jenkins, Travis CI, CircleCI). Основные подходы к контейнеризации (Docker, Kubernetes). Методы мониторинга и логирования приложений. Основные принципы безопасности при развертывании и эксплуатации приложений. Основы работы с облачными платформами (AWS, Azure, Google Cloud). Уметь: Использовать системы контроля версий для совместной работы над кодом.

			Настраивать процессы автоматической сборки и тестирования (CI). Автоматизировать процесс доставки и развертывания приложений (CD). Использовать инструменты мониторинга для отслеживания состояния приложений. Оптимизировать процессы тестирования и развертывания для ускорения поставки кода.
	2. Разрабатывает стратегию DevOps и контейнеризованные приложения, работает с системами контроля версий, создает решения для мониторинга обратной связи		Знать: Принципы и практики DevOps. - Архитектуру и преимущества контейнеризации (Docker, Kubernetes). - Системы контроля версий (Git). - Инструменты и методы мониторинга приложений и инфраструктуры. - Способы получения и обработки обратной связи от пользователей. - Стратегии развертывания и rollback'а приложений. Уметь: Разрабатывать DevOps стратегию для проекта. - Контейнеризировать приложения с использованием Docker. - Использовать системы оркестрации контейнеров (Kubernetes). - Работать с системами контроля версий (Git, GitLab, GitHub). - Настраивать системы мониторинга (Prometheus, Grafana, ELK Stack) для получения информации о состоянии приложений и инфраструктуры. - Анализировать обратную связь от пользователей и использовать ее для улучшения продукта.
	3. Владеет навыками автоматизации развертывания, масштабирования контейнеризованных приложений и управления ими, а также навыками автоматизации настройки серверов и инфраструктурных		Знать: Инструменты автоматизации развертывания (Ansible, Terraform, Chef, Puppet). - Стратегии масштабирования приложений (горизонтальное, вертикальное). - Принципы построения отказоустойчивых систем. - Технологии виртуализации и облачные платформы (AWS, Azure, Google Cloud). Уметь: Автоматизировать развертывание приложений с использованием инструментов Infrastructure as Code

		сервисов для обеспечения быстрого развертывания и сокращения времени восстановления после сбоев	(IaaS). - Масштабировать контейнеризированные приложения. - Настраивать автоматическое восстановление после сбоев. - Автоматизировать настройку серверов и инфраструктурных сервисов. - Использовать облачные платформы для развертывания и управления приложениями. - Реализовывать стратегии резервного копирования и восстановления данных.
--	--	---	--

3. Место дисциплины в структуре образовательной программы

Дисциплина «Основы веб-разработки» относится к Циклу профиля (элективный) по направлению подготовки 09.03.03 - Прикладная информатика, ОП «Прикладные информационные системы в экономике и финансах».

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

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з.е. и часах)	Семестр 6 (в часах)
Общая трудоёмкость дисциплины	3/108	108
Контактная работа-Аудиторные занятия	50	50
Лекции	16	16
Семинары, практические занятия	34	34
Самостоятельная работа	58	58
Вид текущего контроля	контрольная работа	контрольная работа
Вид промежуточной аттестации	зачет	зачет

Очно-заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з.е. и часах)	Семестр 7 (в часах)
Общая трудоёмкость дисциплины	3/108	108
<i>Контактная работа-Аудиторные занятия</i>	34	34
Лекции	16	16
Семинары, практические занятия	18	18
<i>Самостоятельная работа</i>	74	74
Вид текущего контроля	контрольная работа	контрольная работа
Вид промежуточной аттестации	зачет	зачет

Институт открытого образования

Заочная форма обучения

Вид учебной работы по дисциплине	Всего (в з.е. и часах)	Семестр 8 (в часах)
Общая трудоёмкость дисциплины	3/108	108
<i>Контактная работа-Аудиторные занятия</i>	10	10
Лекции	2	2
Семинары, практические занятия	8	8
<i>Самостоятельная работа</i>	98	98
Вид текущего контроля	контрольная работа	контрольная работа
Вид промежуточной аттестации	зачет	зачет

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 контентом - назначение, использование примеры. Создания контентного сайта на WordPress.

5.2. Учебно-тематический план

Очная форма обучения

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	*Контактная работа- Аудиторная работа			Самостоя тельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Введение в веб разработку	4	3	1	2	1	Выполнение задач, устный опрос
2	Основы HTML	17	9	3	6	8	Выполнение задач, устный опрос
3	CSS	12	8	2	6	4	Выполнение

							задач, устный опрос
4	JavaScript	16	8	2	6	8	Выполнение задач, устный опрос
5	Деплой статичных сайтов	5	3	1	2	2	Выполнение задач, устный опрос
6	Фронтенд фреймворк React.js	14	6	2	4	8	Выполнение задач, устный опрос
7	Серверное программирование	14	6	2	4	8	Выполнение задач, устный опрос
8	ORM	7	3	1	2	4	Выполнение задач, устный опрос
9	API	7	2	1	1	5	Выполнение задач, устный опрос
10	CMS	12	2	1	1	10	Выполнение задач, устный опрос
	В целом по дисциплине	108	50	16	34	58	Согласно учебному плану: контрольная работа
	Итого в %		46	32	68	54	

Очно-заочная форма обучения

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	*Контактная работа- Аудиторная работа			Самостояте льная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Введение в веб разработку	4	2	1	1	2	Выполнение задач, устный опрос

2	Основы HTML	17	5	3	2	12	Выполнение задач, устный опрос
3	CSS	12	4	2	2	8	Выполнение задач, устный опрос
4	JavaScript	16	4	2	2	12	Выполнение задач, устный опрос
5	Деплой статических сайтов	5	2	1	1	3	Выполнение задач, устный опрос
6	Фронтенд фреймворк React.js	14	4	2	2	10	Выполнение задач, устный опрос
7	Серверное программирование	14	6	2	4	8	Выполнение задач, устный опрос
8	ORM	7	3	1	2	4	Выполнение задач, устный опрос
9	API	7	2	1	1	5	Выполнение задач, устный опрос
10	CMS	12	2	1	1	10	Выполнение задач, устный опрос
	В целом по дисциплине	108	34	16	18	74	Согласно учебному плану: контрольная работа
	Итого в %		31	47	53	69	

**Институт открытого образования
Заочная форма обучения**

№ п/п	Наименование тем (разделов) дисциплины	Трудоемкость в часах					Формы текущего контроля успеваемости
		Всего	*Контактная работа- Аудиторная работа			Самостояте льная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		

1	Введение в веб разработку	7	1	-	-	6	Выполнение задач, устный опрос
2	Основы HTML	10	2	1	1	8	Выполнение задач, устный опрос
3	CSS	9	1	-	1	8	Выполнение задач, устный опрос
4	JavaScript	14	2	1	1	14	Выполнение задач, устный опрос
5	Деплой статичных сайтов	7	1	-	1	6	Выполнение задач, устный опрос
6	Фронтенд фреймворк React.js	15	1	-	1	14	Выполнение задач, устный опрос
7	Серверное программирование	16	2	-	1	14	Выполнение задач, устный опрос
8	ORM	9	1	-	1	8	Выполнение задач, устный опрос
9	API	9	1	-	1	8	Выполнение задач, устный опрос
10	CMS	12	-	-	-	12	Выполнение задач, устный опрос
	В целом по дисциплине	108	10	2	8	98	Согласно учебному плану: контрольная работа
	Итого в %		9	20	80	91	

* объем контактной работы в очно-заочной/заочной формах обучения и индивидуальных учебных планах определяется соответствующими учебными планами. Темы, реализуемые в виде контактной работы, определяются преподавателем самостоятельно, исходя из уровня их сложности.

5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9	Формы проведения занятий
Введение в веб разработку	Установка и настройка WebStorm IDE. Установка Web-сервера 8.[1-4].	Решение и разбор задач
Основы HTML	Создание статичных веб-страниц. Табличная и блочная верстка 8.[1-4].	Решение и разбор задач
CSS	Стилизация веб-страниц. Создание альтернативных стилей. Адаптивная верстка. Использование фреймворков. Создание дизайна на основе Bootstrap 8.[1-4]	Решение и разбор задач
JavaScript	Создание динамических веб-страниц. Создание обработчиков событий 8.[1-4].	Решение и разбор задач
Деплой статичных сайтов	Развертывание статического сайта на облачном хостинге 8.[1-4]	Решение и разбор задач
Фронтенд фреймворк React.js	Создание одностраничного веб-приложения. Создание приложения на React с навигацией. Продвинутое возможности React. Хуки 8.[1-4].	Решение и разбор задач
Серверное программирование	Создание простого серверного приложения на Express.js 8.[1-4]	Решение и разбор задач
ORM	Создание трехзвенного веб-приложения с использованием Sequelize 8.[1-4]	Решение и разбор задач
API	Создание ReST API приложения 8.[1-4].	Решение и разбор задач
CMS	Создания контентного сайта на WordPress 8.[1-4].	Решение и разбор задач

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
Введение в веб разработку	Языки и библиотеки веб-разработки, ландшафт профессий в области веб-программирования	Работа с интернет-источниками по теме, решение задач

Основы HTML	Продвинутые возможности языка HTML. Универсальные и специализированные атрибуты.	Работа с интернет-источниками по теме, решение задач
CSS	Использование препроцессоров и создание сложных стилей. Svelte. Material.	Работа с интернет-источниками по теме, решение задач
JavaScript	TypeScript	Работа с интернет-источниками по теме, решение задач
Деплой статичных сайтов	Развертывание сайта на собственном или выделенном хостинге.	Работа с интернет-источниками по теме, решение задач
Фронтенд фреймворк React.js	Создание прогрессивного веб приложения	Работа с интернет-источниками по теме, решение задач
Серверное программирование	Создание простого серверного приложения на FastAPI	Работа с интернет-источниками по теме, решение задач
ORM	Создание трехзвенного веб-приложения с использованием SQLAlchemy	Работа с интернет-источниками по теме, решение задач
API	Создание API приложения на основе GraphQL	Работа с интернет-источниками по теме, решение задач
CMS	Деплой сложного веб-приложения.	Работа с интернет-источниками по теме, решение задач

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

Примерные задания контрольной работы

1. Создать простую страницу с использованием HTML и CSS, добавить стили с помощью Bootstrap.
2. Создать форму на странице с использованием HTML и CSS, добавить валидацию полей с помощью JavaScript.
3. Создать адаптивный дизайн страницы с использованием Bootstrap.
4. Создать простой блог на WordPress, добавить стили с помощью CSS.

5. Создать простой сайт-визитку на React, добавить стили с помощью CSS.
6. Создать CRUD-приложение на Express и MongoDB, использовать ORM для работы с базой данных.
7. Создать API на FastAPI для получения и обработки данных, использовать ORM для работы с базой данных.
8. Создать административную панель на WordPress для управления контентом сайта.
9. Создать простую игру на React, добавить стили с помощью CSS.
10. Создать приложение для обмена сообщениями на Express и Socket.IO, использовать Bootstrap для стилизации интерфейса.
11. Создать интерактивную карту на React, использовать сторонние библиотеки для работы с картами (например, Leaflet).
12. Создать онлайн-магазин на Express и MongoDB, использовать ORM для работы с базой данных, добавить стили с помощью Bootstrap.
13. Создать приложение для записи заметок на React, использовать localStorage для хранения данных, добавить стили с помощью CSS.
14. Создать систему авторизации и аутентификации на Express, использовать JSON Web Tokens для безопасной передачи данных.
15. Создать простой блокнот на FastAPI, использовать базу данных для хранения заметок, добавить стили с помощью CSS.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

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

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы 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

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-3 Способен наладить и администрировать процесс построения и функционирования отказоустойчивого программного обеспечения информационных систем цифровой экономики, в том числе непрерывную поставку и развертывание программного кода	1. Применяет основные принципы организации командной работы разработчиков над проектом, процедуры, подходы и методы оптимизации и автоматизации процедур тестирования, доставки кода и запуска приложений на серверах	Знать: Основные принципы Agile и Scrum методологий разработки. Методы организации командной работы (например, Kanban, парное программирование). Роли и обязанности в команде разработчиков. Принципы работы с системами контроля версий (например, Git, GitLab, GitHub). Концепции CI/CD (Continuous Integration/Continuous Delivery/Deployment). Методы автоматизации тестирования (модульное, интеграционное, системное). Инструменты автоматизации сборки, тестирования и развертывания (например, Jenkins, Travis CI, CircleCI).	I. Agile и Scrum: 1. (Тест с выбором ответа): Какой из перечисленных принципов является ключевым в Agile-манифесте? ○ а) Подробная документация ○ б) Следование плану ○ в) Работаящий продукт ○ г) Контрактные обязательства 2. (Открытый вопрос): Опишите основные этапы Scrum-спринта и их цели. 3. (Ситуационная задача): Ваша команда столкнулась с трудностями в оценке задач. Какие методы оценки (например, Story Points, Planning Poker) вы можете предложить и почему? II. Организация командной работы: 1. (Сопоставление): Сопоставьте методологию (Kanban, парное программирование) с ее основными характеристиками. (Предоставляются списки характеристик и методологий). 2. (Открытый вопрос): В чем преимущества и недостатки парного программирования? В каких ситуациях оно наиболее эффективно? 3. (Ситуационная задача): В команде возник конфликт из-за разных подходов к решению задачи. Как вы, как Scrum-мастер или тимлид, можете разрешить конфликт, используя принципы Agile и методы командной работы? III. Роли и обязанности: 1. (Тест с выбором ответа): Кто несет ответственность за определение приоритетов задач в Scrum?

		Основные подходы к контейнеризации (Docker, Kubernetes). Методы мониторинга и логирования приложений. Основные принципы безопасности при развертывании и эксплуатации приложений. Основы работы с облачными платформами (AWS, Azure, Google Cloud). Уметь: Использовать системы контроля версий для совместной работы над кодом. Настраивать процессы автоматической сборки и тестирования (CI). Автоматизировать процесс доставки и развертывания приложений (CD). Использовать инструменты мониторинга для отслеживания состояния приложений. Оптимизировать процессы тестирования и развертывания для ускорения поставки кода.	☐ a) Команда разработчиков ☐ b) Scrum-мастер ☐ c) Product Owner ☐ d) Менеджер проекта 2. (Открытый вопрос): Опишите роль Scrum-мастера и его основные обязанности. 3. (Ситуационная задача): Product Owner постоянно меняет требования в середине спринта. Как это влияет на команду и что вы, как Scrum-мастер, можете сделать? IV. Системы контроля версий (Git): 1. (Тест с выбором ответа): Какая команда Git используется для объединения изменений из одной ветки в другую? ☐ a) git push ☐ b) git commit ☐ c) git merge ☐ d) git checkout 2. (Практическое задание): Используя Git, создайте новую ветку, внесите изменения в файл, добавьте изменения в индекс, сделайте коммит и отправьте ветку на удаленный репозиторий (имитация реальной задачи). 3. (Открытый вопрос): Объясните разницу между командами git pull и git fetch. V. CI/CD: 1. (Тест с выбором ответа): Что означает Continuous Integration (CI)? ☐ a) Непрерывное тестирование ☐ b) Непрерывное развертывание ☐ c) Непрерывная интеграция кода ☐ d) Непрерывный мониторинг 2. (Открытый вопрос): Опишите этапы типичного CI/CD пайплайна. 3. (Ситуационная задача): Как вы настроите CI/CD пайплайн для автоматического развертывания веб-приложения на тестовый сервер после каждого коммита в главную ветку?
	2. Разрабатывает стратегию DevOps и контейнеризованные приложения, работает с системами контроля версий, создает решения для мониторинга обратной связи	Знать: Принципы и практики DevOps. - Архитектуру и преимущества контейнеризации (Docker, Kubernetes). - Системы контроля версий (Git). - Инструменты и методы мониторинга приложений и инфраструктуры. - Способы получения и обработки обратной связи от пользователей. - Стратегии развертывания и rollback'а приложений.	I. Знание (Теоретические задания): 1. (Тест с выбором ответа): Какой принцип DevOps наиболее важен для улучшения сотрудничества между разработчиками и операционными командами? ☐ a) Автоматизация всего ☐ b) Совместная ответственность и владение ☐ c) Непрерывное развертывание ☐ d) Микросервисная архитектура 2. (Открытый вопрос): Опишите, что такое "Infrastructure as Code" (IaC) и какие преимущества она предоставляет в контексте DevOps. 3. (Сопоставление): Сопоставьте концепции из DevOps с их описаниями (например, CI, CD, IaC, мониторинг, обратная связь). (Предоставляется список концепций и их описаний).

		Уметь: Разрабатывать DevOps стратегию для проекта. - Контейнеризировать приложения с использованием Docker. - Использовать системы оркестрации контейнеров (Kubernetes). - Работать с системами контроля версий (Git, GitLab, GitHub). - Настраивать системы мониторинга (Prometheus, Grafana, ELK Stack) для получения информации о состоянии приложений и инфраструктуры. - Анализировать обратную связь от пользователей и использовать ее для улучшения продукта.	4. (Тест с несколькими вариантами ответа): Какие преимущества предоставляет контейнеризация с использованием Docker? (Выберите все подходящие варианты) ☐ а) Изоляция приложений ☐ б) Воспроизводимость среды ☐ в) Упрощение развертывания ☐ г) Повышение производительности оборудования (не всегда, зависит от случая) ☐ д) Ускорение разработки 5. (Открытый вопрос): Объясните разницу между <code>git merge</code> и <code>git rebase</code>. В каких ситуациях целесообразно использовать каждый из этих подходов? 6. (Перечисление): Перечислите 3-5 ключевых метрик, которые необходимо мониторить для веб-приложения, и объясните, почему они важны. 7. (Открытый вопрос): Опишите различные способы получения обратной связи от пользователей о работе вашего приложения и как эту обратную связь можно использовать для улучшения продукта. 8. (Открытый вопрос): Опишите стратегии roll-back'a приложений, объясните их важность и назовите основные инструменты, применяемые в каждом подходе. II. Умение (Практические задания и ситуационные задачи): 1. (Ситуационная задача): Ваша команда разрабатывает новое веб-приложение. Опишите DevOps-стратегию, которую вы будете использовать, учитывая следующие факторы: быстрый вывод на рынок, высокая доступность, масштабируемость, безопасность и возможность быстрого восстановления после сбоев. Укажите инструменты, которые будете использовать для каждого этапа. 2. (Практическое задание): ☐ Дано: Простое веб-приложение (например, на Python с Flask или Node.js). ☐ Задача: Создайте Dockerfile для этого приложения, настройте Docker Compose для запуска приложения и его зависимостей (например, базы данных) и убедитесь, что приложение запускается корректно в контейнере.
3.	Владеет навыками автоматизации развертывания, масштабирования контейнеризова	Знать: Инструменты автоматизации развертывания (Ansible, Terraform, Chef, Puppet). - Стратегии	I. Знание (Теоретические задания): 1. (Тест с выбором ответа): Какой инструмент из перечисленных относится к инструментам Infrastructure as Code (IaC)? ☐ а) Kubernetes ☐ б) Docker ☐ в) Terraform ☐ г) Grafana

	нных приложений и управления ими, а также навыками автоматизации настройки серверов и инфраструктурных сервисов для обеспечения быстрого развертывания и сокращения времени восстановления после сбоев	масштабирования приложений (горизонтальное, вертикальное). - Принципы построения отказоустойчивых систем. - Технологии виртуализации и облачные платформы (AWS, Azure, Google Cloud). Уметь: Автоматизировать развертывание приложений с использованием инструментов Infrastructure as Code (IaC). - Масштабировать контейнеризированные приложения. - Настраивать автоматическое восстановление после сбоев. - Автоматизировать настройку серверов и инфраструктурных сервисов. - Использовать облачные платформы для развертывания и управления приложениями. - Реализовывать стратегии резервного копирования и восстановления данных.	2. (Открытый вопрос): Опишите, в чем разница между горизонтальным и вертикальным масштабированием приложений. Какие факторы следует учитывать при выборе стратегии масштабирования? 3. (Перечисление): Перечислите как минимум три ключевых принципа построения отказоустойчивых систем. 4. (Сопоставление): Сопоставьте облачные платформы (AWS, Azure, Google Cloud) с предоставляемыми ими сервисами (например, вычислительные ресурсы, хранилище, базы данных). 5. (Открытый вопрос): Объясните, что такое “immutable infrastructure” и как этот подход связан с инструментами IaC. 6. (Открытый вопрос): Объясните, какие этапы включает в себя процесс резервного копирования и восстановления данных (backup and recovery). Какие стратегии восстановления вы знаете? II. Умение (Практические задания и ситуационные задачи): 1. (Ситуационная задача): Вам необходимо автоматизировать развертывание веб-приложения (например, на Python с Django или Node.js с Express) на AWS. Используя Terraform или Ansible, опишите, какие шаги вы предпримите для создания инфраструктуры (EC2 instances, VPC, Security Groups, Load Balancer) и развертывания приложения. Предоставьте примеры кода или конфигурационных файлов. 2. (Ситуационная задача): У вас есть контейнеризированное приложение, работающее в Kubernetes. При росте нагрузки необходимо масштабировать приложение. Объясните, как вы будете использовать Horizontal Pod Autoscaler (HPA) в Kubernetes для автоматического масштабирования приложения на основе метрик CPU и памяти. Приведите пример YAML-файла для HPA.
--	---	---	---

Примеры практико-ориентированных (ситуационных) заданий

1. Даны картинки. По нажатию на любую картинку увеличьте ее в 2 раза.
2. Даны N инпутов с классом .num и кнопка. По нажатию на кнопку получите числа, стоящие в этих инпутах и запишите их сумму в абзац с id=”result”.
3. Дана таблица с числами. По нажатию на кнопку найдите ячейку, в которой хранится максимальное число, и сделайте ее фон красным.
4. Дана таблица с числами. По нажатию на ячейку она активируется и становится красного цвета. Активировать можно много ячеек. Под таблицей кнопка. По нажатию по этой кнопке в абзац ниже выведите

сумму активированных ячеек. Реализуйте кнопку ‘сбросить активированные ячейки’.

5. Реализуйте раскрывающийся список. По умолчанию есть список стран (ul), по нажатию на страну внутри li со страной появляется список городов.
6. В инпут через запятую вводятся страны. По нажатию на кнопку сделайте так, чтобы эти страны записались в ul под инпутом (каждая страна отдельный li).
7. Дан ряд ссылок. Сделайте так, чтобы по нажатию на ссылку она становилась с красным фоном. По нажатию на другую ссылку выделение первой ссылки снимается и выделяется та, на которую мы нажали. В абзац ниже пишите текст активной ссылки.

Примерные вопросы для подготовки к зачету

1. Понятие веб-технологий. Чем веб отличается от других сетевых служб?
2. Основные сетевые протоколы, использующиеся 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 обеспечения 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.

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, 2025. — 204 с. — (Высшее образование). — ЭБС Юрайт. — URL: <https://urait.ru/bcode/567610> (дата обращения: 14.03.2025). — Текст: электронный.
- 2.Гладкий, А. А. Веб-самоделки. Как самому создать сайт быстро и профессионально: практическое пособие / А. А. Гладкий. – Москва; Берлин: Директ-Медиа, 2020. – 266 с. – ЭБС Университетская библиотека ONLINE. – URL: <https://biblioclub.ru/index.php?page=book&id=577164> (дата обращения: 14.03.2025). – Текст: электронный.

Дополнительная литература:

- 3.Савельев, А. О. HTML5. Основы клиентской разработки: учебное пособие / А. О. Савельев, А. А. Алексеев. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 272 с. - ЭБС Университетская библиотека ONLINE. - URL:

<http://biblioclub.ru/index.php?page=book&id=429150> (дата обращения: 14.03.2025). - Текст: электронный.

4. Брокшмидт, К. Введение в разработку приложений для Windows 8 с использованием HTML, CSS и JavaScript / К. Брокшмидт. – 2-е изд., испр. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 460 с. – ЭБС Университетская библиотека ONLINE. - URL: <https://biblioclub.ru/index.php?page=book&id=428973> (дата обращения : 14.03.2025). – Текст: электронный.

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

1. Электронная библиотека Финансового университета (ЭБ)
<http://elib.fa.ru/>
2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>
4. Электронно-библиотечная система Znanium <http://www.znaniy.com>
5. Электронно-библиотечная система издательства «ЮРАЙТ»
<https://urait.ru/>
6. Электронно-библиотечная система издательства Проспект
<http://ebs.prospekt.org/books>
7. Электронно-библиотечная система издательства Лань
<https://e.lanbook.com/>
8. Деловая онлайн-библиотека Alpina Digital <http://lib.alpinadigital.ru/>
9. Электронная библиотека Издательского дома «Гребенников»
<https://grebennikon.ru/>

10. Математические журналы: полнотекстовая коллекция
Математического института им. В.А. Стеклова РАН
<https://www.mathnet.ru/>
11. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
12. Национальная электронная библиотека <http://нэб.рф/>
13. Ресурсы информационно-аналитического агентства по финансовым
рынкам Cbonds.ru <https://cbonds.ru/>
14. СПАРК <https://spark-interfax.ru/>
15. CNKI. Academic Reference <https://ar.oversea.cnki.net/>
16. Электронные продукты издательства Elsevier
<http://www.sciencedirect.com>
17. Emerald: Management eJournal Portfolio <https://www.emerald.com/insight/>
18. Реферативная база данных по математике MathSciNET
<https://mathscinet.ams.org/mathscinet/>
19. Коллекция научных журналов Oxford University Press
<https://academic.oup.com/journals/>
20. Электронные коллекции книг и журналов издательства Springer:
<http://link.springer.com/>
21. Платформа STATISTA <https://www.statista.com/>
22. База данных научных журналов издательства Wiley
<https://onlinelibrary.wiley.com/>
23. PyIru 1.0.9 [Электронный ресурс]: сайт. – Режим доступа:
<https://pypi.python.org/pypi/pylru>
24. Дорожная карта Frontend-разработчика - <https://roadmap.sh/frontend>
25. Дорожная карта Backend-разработчика - <https://roadmap.sh/backend>
26. Канал Traversy Media - <https://www.youtube.com/user/TechGuyWeb>
27. Учебник JavaScript - <https://learn.javascript.ru>
28. Учебник React.js - <https://ru.reactjs.org>

10. Методические указания для обучающихся по освоению дисциплины

При изучении теоретического материала необходимо опираться на рабочую программу дисциплины, материалы лекций и литературу из основного списка. Кроме этого, необходимо активно работать с Интернет-источниками и пособиями других авторов, помогающими усвоить материал отдельных разделов программы.

Необходимо конспектировать лекции, пометая сложные и непонятные моменты с тем, чтобы задать вопросы лектору в конце лекции или же на консультации.

При подготовке к семинарским занятиям необходимо изучить вопросы, вынесенные на самостоятельное изучение, так как семинарские занятия предполагают их обсуждение и дискуссию по теме; кроме того, задания для самостоятельной работы необходимы для того, чтобы успешно выполнить самостоятельные задания на семинарах.

Индивидуальные задания для работы на компьютере, файлы с выполненными заданиями необходимо хранить в личной сетевой папке в компьютерной сети вуза.

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. Пакет офисных программ;
2. Антивирус Kaspersky;

11.2. Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»;

2. Информационно-правовая система «Консультант Плюс»;
 3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>;
 4. 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

Для проведения лекций и практических занятий необходима аудитория, оснащенная проектором и компьютерами с постоянным подключением к сети Интернет.