

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шавалович

Должность: Ректор

Дата подписания: 05.06.2026 10:00:17

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a5825f9fa4304cc

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени академика М.Д. Миллионщикова

«УТВЕРЖДАЮ»

Первый проректор

И.Г. Гайрабеков



РАБОЧАЯ ПРОГРАММА

дисциплины

«Разработка Web-приложений с применением фреймворков»

Направление подготовки/специальность

09.03.03 Прикладная информатика

Направленность/специализация

«Разработка мобильных и Web-приложений»

Квалификация

Бакалавр

Год начала подготовки 2025

Грозный - 2025

1. Цели и задачи освоения дисциплины

Целью освоения дисциплины является подготовка к профессиональной деятельности в области разработки мобильных и веб-приложений на языке JavaScript, по направлению подготовки 09.03.03 Прикладная информатика (профиль подготовки: Разработка мобильных и Web-приложений) посредством обеспечения этапов формирования компетенций, предусмотренных ФГОС, в части представленных ниже знаний, умений и навыков.

Задачами дисциплины является изучение различных фреймворков для разработки веб-приложений, таких как React.js.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части блока 1.

В свою очередь, данный курс, помимо самостоятельного значения, является предшествующей дисциплиной для курса «Информационная безопасность мобильных и Web-приложений».

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Таблица 1

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине (ЗУВ)
Общепрофессиональные		
ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.1. Выбирает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.2. Применяет современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Знает принципы работы с отечественным программным обеспечением и информационными технологиями. Умеет применять современные информационные технологии и программные средства для решения задач профессиональной деятельности. Владеет навыками работы с современными информационными технологиями и программными средствами на уровне, необходимом для эффективного выполнения профессиональных задач.
Профессиональные		
ПК – 5. Способен разрабатывать и адаптировать прикладное программное обеспечение	ПК – 5.1 Проектирует и разрабатывает прототипы ИС ПК – 5.2 Применяет технологии разработки и управления базами данных ИС ПК – 5.3 Обеспечивает соответствие разработанного кода и	Знает Принципы работы с DOM (Document Object Model) и его основные методы. Принципы работы с асинхронным программированием и использование промисов и async/await. Основные концепции разработки

	процесса кодирования на языках программирования принятым в организации или проекте стандартам и технологиям	веб-приложений, такие как роутинг, обработка событий и взаимодействие с сервером. Использование библиотеки React для разработки веб-приложений и основные понятия, такие как компоненты, JSX, Virtual DOM и управление состоянием. Умеет Создавать интерактивные веб-приложения с использованием языка JavaScript и библиотеки React. Работать с DOM для динамического изменения содержимого веб-страницы. Осуществлять асинхронные запросы к серверу с помощью XMLHttpRequest или Fetch API. Разрабатывать SPA (Single Page Applications) с использованием React и обеспечивать навигацию в приложении. Осуществлять стилизацию компонентов с помощью CSS и CSS-in-JS подходов. Работать с формами и обрабатывать события в React-приложениях. Тестировать компоненты и приложения на React с использованием различных инструментов и методов. Владеет Умением эффективно использовать библиотеку React для создания масштабируемых и легко поддерживаемых веб-приложений. Навыками работы с архитектурными паттернами и лучшими практиками разработки приложений на React. Умением оптимизировать производительность веб-приложений и использовать SSR для улучшения пользовательского опыта.
--	--	---

		Навыками тестирования и отладки веб-приложений для обеспечения их качества и надежности.
--	--	--

4. Объем дисциплины и виды учебной работы

Таблица 2

Вид учебной работы		Всего часов/ зач.ед.		Всего часов/ зач.ед.	
		ОФО 7 семестр	ОФО 8 семестр	ЗФО 8 семестр	ЗФО 9 семестр
Контактная работа (всего)		68/1.9	48/1.3	16/0.4	16/0.4
В том числе:					
Лекции		34/0.95	24/0.66	8/0.2	8/0.2
Практические занятия					
Семинары					
Лабораторные работы		34/0.95	24/0.66	8/0.2	8/0.2
Самостоятельная работа (всего)		76/2.1	96/2.7	128/3.6	128/3.6
В том числе:					
Курсовая работа (проект)					
Расчетно-графические работы					
Индивидуальное задание					72/2
Рефераты		40/1.1	39/1.1	72/2	
Доклады					
Презентации					
<i>И (или) другие виды самостоятельной работы:</i>					
Подготовка к лабораторным работам		36/1	36/1	36/1	36/1
Подготовка к практическим занятиям					
Подготовка к зачету				20/0.6	
Подготовка к экзамену			21/0.6		20/0.5
Вид отчетности		Зачет	Экзамен	Зачет	Экзамен
Общая трудоемкость дисциплины	ОФО - 288	144	144	144	144
	ЗФО - 288	4	4	4	4

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Таблица 3

№ п/п	Наименование раздела дисциплины по семестрам	Часы лекц. занятий ОФО	Часы лаборат. занятий ОФО	Часы лекц. занятий ЗФО	Часы лаборат. занятий ЗФО
1.	Тема 1. Введение в разработку веб-приложений с использованием React.	2	2	2	2
2.	Тема 2. Основы работы с React.: компоненты, JSX, Virtual DOM	3	3	2	2
3.	Тема 3. Управление состоянием в React.: useState и useEffect	3	3	2	2
4.	Тема 4. Роутинг и навигация в веб-приложениях на React.	3	3	2	2
5.	Тема 5. Стилизация компонентов в React.: CSS Modules, Styled Components	3	3	2	2
6.	Тема 6. Формы и обработка событий в React.	3	3	2	2
7.	Тема 7. Взаимодействие с сервером: AJAX запросы и работа с API	3	3	2	2
8.	Тема 8. Аутентификация и авторизация в приложениях на React.	3	3	-	-
9.	Тема 9. Оптимизация производительности веб-приложений на React.	3	3	-	-
10.	Тема 10. Тестирование компонентов и приложений на React.	4	4	-	-
11.	Тема 11. Разработка Single Page Applications (SPA) на React.	4	4	-	-
12.	Тема 12. Работа с контекстом и хуками в React.	4	4	-	-
13.	Тема 13. Разработка Single Page Applications (SPA) на React.	4	4	-	-
14.	Тема 14. Работа с контекстом и хуками в React.	4	4	2	2
15.	Тема 15. Архитектура приложений на React.: лучшие практики и паттерны.	4	4	-	-
16.	Тема 16. SSR (Server-Side Rendering) и его применение в приложениях на React.	4	4	-	-
17.	Тема 17. Введение в Next.js и его основные возможности	4	4	-	-
	ИТОГО	58	58	16	16

5.2. Лекционные занятия

Таблица 4

Раздел	Наименование раздела дисциплины	Содержание раздела
6 –й семестр		
1	Тема 1. Введение в разработку веб-приложений с использованием React.	Определение React. и его роль в веб-разработке Основные преимущества использования React. Обзор основных концепций и инструментов React.
2	Тема 2. Основы работы с React.: компоненты, JSX, Virtual DOM	Компонентный подход в React. Использование JSX для описания интерфейса Работа с виртуальным DOM в React.
3	Тема 3. Управление состоянием в React.: useState и useEffect	Понятие состояния в React. Использование useState для управления состоянием компонентов Применение useEffect для работы с побочными эффектами
4	Тема 4. Роутинг и навигация в веб-приложениях на React.	Организация навигации в веб-приложениях Использование библиотеки React. Router для роутинга Создание динамических маршрутов и вложенных маршрутов
5	Тема 5. Стилизация компонентов в React.: CSS Modules, Styled Components	Применение CSS Modules для локальной стилизации компонентов Использование Styled Components для создания стилей внутри JavaScript Сравнение различных подходов к стилизации компонентов в React.
6	Тема 6. Формы и обработка событий в React.	Создание и валидация форм в React. Обработка событий в React. компонентах Использование управляемых и неуправляемых компонентов для работы с формами
7	Тема 7. Взаимодействие с сервером: AJAX запросы и работа с API	Основы работы с AJAX запросами в React. Использование библиотеки Axios для отправки HTTP запросов Работа с RESTful API для получения и отправки данных
8	Тема 8. Аутентификация и авторизация в приложениях на React.	Различные методы аутентификации в веб-приложениях. Реализация механизмов аутентификации и авторизации в React. приложениях Использование JSON Web Tokens (JWT) для безопасной аутентификации

7 –й семестр		
9	Тема 9. Оптимизация производительности веб-приложений на React.	Основные методы оптимизации производительности в React.-приложениях. Асинхронная загрузка компонентов и данных для ускорения рендеринга. Минимизация количества ререндеров компонентов. Использование мемоизации и мемоизированных селекторов.
10	Тема 10. Тестирование компонентов и приложений на React.	Обзор принципов модульного тестирования в React.. Настройка тестового окружения с использованием Jest и Enzyme. Написание тестов для функциональных и классовых компонентов. Тестирование взаимодействия компонентов с внешними библиотеками и API.
11	Тема 11. Разработка Single Page Applications (SPA) на React.	Определение концепции Single Page Applications (SPA). Создание маршрутов с помощью React. Router для SPA. Работа с динамической загрузкой компонентов для SPA. Обработка состояний приложения и истории переходов.
12	Тема 12. Работа с контекстом и хуками в React.	Использование контекста для передачи данных в дочерние компоненты. Обзор хуков React.: useContext, useReducer, useCallback и useMemo. Применение контекста и хуков для управления состоянием и логикой приложения.
13	Тема 13. Разработка Single Page Applications (SPA) на React.	Основные принципы и концепции Single Page Applications (SPA). Создание маршрутизации и навигации для SPA на базе React.. Работа с асинхронной загрузкой данных для улучшения производительности SPA. Использование хуков React. для управления состоянием и сайд-эффектами в SPA.
14	Тема 14. Работа с контекстом и хуками в React.	Продолжение изучения контекста и хуков React.. Применение контекста и хуков для организации глобального состояния приложения. Разработка пользовательских хуков для повторного использования логики.

15	Тема 15. Архитектура приложений на React.: лучшие практики и паттерны.	Обзор основных архитектурных паттернов при разработке React.-приложений. Применение компонентной архитектуры для улучшения масштабируемости и поддерживаемости кода. Реализация контейнер-компонентов и презентационных компонентов. Использование паттерна контейнеров с высокоуровневой логикой и презентационных компонентов для отображения данных.
16	Тема 16. SSR (Server-Side Rendering) и его применение в приложениях на React.	Понятие Server-Side Rendering (SSR) и его роль в разработке React.-приложений. Реализация SSR с использованием библиотеки Next.js. Преимущества и недостатки SSR по сравнению с клиентским рендерингом. Применение SSR для улучшения SEO и производительности приложения.
17	Тема 17. Введение в Next.js и его основные возможности	Обзор фреймворка Next.js и его назначение. Создание SSR приложений с помощью Next.js. Использование статической генерации страниц для оптимизации производительности. Разработка API роутов и маршрутизации в Next.js.

5.3. Практические (семинарские) занятия

Практические занятия учебным планом не предусмотрены.

5.4 Лабораторные занятия

Таблица 5

№ п/ п	Наименование работы	Содержание лабораторной работы
6 – й семестр		
1	Лабораторная работа 1. Основы работы с React.	<i>Цель лабораторной работы: Ознакомиться с базовыми принципами работы с библиотекой React, научиться создавать простые компоненты и использовать JSX синтаксис.</i>
2	Лабораторная работа 2. Управление состоянием в React.	<i>Цель лабораторной работы: Изучить методы управления состоянием компонентов в React с помощью хуков useState и useEffect, научиться обновлять состояние и реагировать на изменения.</i>
3	Лабораторная работа 3. Роутинг и навигация в React.	<i>Цель лабораторной работы: Познакомиться с библиотекой React Router и научиться создавать маршруты для навигации между страницами в веб-приложении на React.</i>
4	Лабораторная работа 4. Стилизация компонентов в React.	<i>Цель лабораторной работы: Изучить различные способы стилизации компонентов в React, включая CSS Modules и Styled Components, научиться создавать стилизованные интерфейсы.</i>
5	Лабораторная работа 5. Формы и обработка событий в React.	<i>Цель лабораторной работы: Практически изучить создание и валидацию форм в React, научиться обрабатывать события в компонентах для реализации интерактивности.</i>
6	Лабораторная работа 6. Взаимодействие с сервером в React.	<i>Цель лабораторной работы: Освоить основы взаимодействия с сервером в React-приложениях с использованием Fetch API для отправки запросов на сервер и получения данных.</i>
7 – й семестр		
7	Лабораторная работа 7. Оптимизация производительности в React.	<i>Цель лабораторной работы: Изучить методы оптимизации производительности React-приложений, включая мемоизацию, useMemo и useCallback, и применить их на практике.</i>
8	Лабораторная работа 8. Тестирование компонентов и приложений на React.	<i>Цель лабораторной работы: Научиться писать модульные и интеграционные тесты для React-компонентов и приложений с использованием библиотеки Jest.</i>
9	Лабораторная работа 9. Разработка SPA на React.	<i>Цель лабораторной работы: Практически освоить разработку Single Page Applications (SPA) на React, организовать структуру приложения и управлять его состоянием.</i>
10	Лабораторная работа 10. Работа с контекстом и хуками в React.	<i>Цель лабораторной работы: Изучить контекст и пользовательские хуки в React и применить их для передачи данных между компонентами и управления состоянием.</i>
11	Лабораторная работа 11. Архитектура приложений на React.	<i>Цель лабораторной работы: Разработать приложение с использованием лучших практик и паттернов, организовать его архитектуру и структуру компонентов.</i>

12	Лабораторная работа 12. SSR и применение в React.	<i>Цель лабораторной работы: Изучить Server-Side Rendering (SSR) и его применение в React-приложениях для улучшения производительности и SEO.</i>
----	---	---

6. Самостоятельная работа студентов по дисциплине

6.1. Подготовка рефератов

В рамках самостоятельной работы студент выполняет ряд работ по предложенным темам. Студент самостоятельно собирает необходимую для выполнения работ информацию, в ряде случаев дополняя ее своими обоснованными оценками и допущениями.

Темы рефератов (6 семестр)

1. Введение в React: основные концепции и принципы работы.
2. JSX: особенности синтаксиса и преимущества использования в разработке на React.
3. Управление состоянием в React: хуки `useState` и `useEffect`.
4. Маршрутизация в React-приложениях с использованием `React Router`.
5. Стилизация компонентов в React: сравнение различных подходов.
6. Формы и их обработка в React-приложениях.
7. Асинхронное программирование в React: обработка асинхронных операций.
8. Аутентификация и авторизация в приложениях на React.
9. Оптимизация производительности веб-приложений на React.
10. Тестирование React-компонентов и приложений: методы и инструменты.

6.2. Подготовка рефератов

Темы рефератов (7 семестр)

1. Разработка Single Page Applications (SPA) на React: принципы и практика.
2. Контекст и хуки в React: использование для управления состоянием приложения.
3. Архитектура приложений на React: лучшие практики и паттерны проектирования.
4. Server-Side Rendering (SSR) и его роль в разработке React-приложений.
5. Введение в `Next.js` и его основные возможности.
6. Менеджмент состояния в React: сравнение различных подходов (`Redux`, `MobX` и др.).
7. Оптимизация маршрутизации и навигации в React-приложениях.
8. Использование контекста для передачи данных в дочерние компоненты в React.
9. Разработка компонентов высшего порядка (HOC) в React.
10. Работа с анимациями в React-приложениях: библиотеки и подходы.

7. Оценочные средства

7.1. Вопросы к первой рубежной аттестации (6 семестр)

1. Какова основная цель использования библиотеки React в разработке веб-приложений?
2. Какие основные преимущества предоставляет JSX в сравнении с обычным JavaScript?
3. Какие методы React используются для управления состоянием компонентов? Приведите примеры их применения.
4. Как осуществляется роутинг и навигация в веб-приложениях на React? Какие инструменты для этого используются?
5. Как происходит стилизация компонентов в React-приложениях? Расскажите о преимуществах подходов CSS Modules и Styled Components.
6. Как обрабатываются формы и события в React-приложениях? Как организовать валидацию форм?
7. Как осуществляется взаимодействие с сервером в React-приложениях? Укажите на основные методы и инструменты.
8. Какие механизмы используются для реализации аутентификации и авторизации в приложениях на React?
9. Какие средства и методы оптимизации производительности могут быть применены в React-приложениях?
10. Какие инструменты и подходы используются для тестирования компонентов и приложений на React?

*Образец аттестационного билета
(за 1 рубежную аттестацию)*

Грозненский государственный нефтяной технический университет

им. акад. М.Д. Миллионщикова

ИЦЭ и ТП

Группа «ПИ» Семестр «_6_»

Дисциплина «Разработка Веб-приложений с применением фреймворков»

Билет № 1

1. Как происходит стилизация компонентов в React-приложениях?
2. Расскажите о преимуществах подходов CSS Modules и Styled Components.
3. Как обрабатываются формы и события в React-приложениях?
4. Как организовать валидацию форм?

Подпись преподавателя _____ Подпись заведующего кафедрой _____

Вопросы ко второй рубежной аттестации (6 семестр)

1. Каковы особенности разработки Single Page Applications (SPA) на React? В чем их преимущества и недостатки?
2. Какие методы и инструменты используются для работы с контекстом и хуками в React-приложениях?
3. Какие паттерны и лучшие практики применяются при проектировании архитектуры приложений на React?
4. Как SSR (Server-Side Rendering) влияет на производительность веб-приложений на React? В каких случаях его стоит использовать?
5. Что такое Next.js и какие возможности он предоставляет для разработки приложений на React?
6. Какие средства и методы используются для тестирования компонентов и приложений на React?
7. Какие аспекты безопасности следует учитывать при разработке и аутентификации в приложениях на React?
8. Какие сценарии оптимизации производительности чаще всего применяются в веб-приложениях на React?
9. Какие особенности работы с API следует учитывать при взаимодействии с сервером в React-приложениях?
10. Какие инструменты и методы используются для управления состоянием приложения в React и обеспечения его целостности?

*Образец аттестационного билета
(за 2 рубежную аттестацию)*

Грозненский государственный нефтяной технический университет

им. акад. М.Д. Миллионщикова

ИЦЭ и ТП

Группа «ПИ» Семестр «_6_»

Дисциплина «Разработка Веб-приложений с применением фреймворков»

Билет № 1

1. Какие средства и методы используются для тестирования компонентов и приложений на React?
2. Какие аспекты безопасности следует учитывать при разработке и аутентификации в приложениях на React?
3. Какие сценарии оптимизации производительности чаще всего применяются в веб-приложениях на React?
4. Какие особенности работы с API следует учитывать при взаимодействии с сервером в React-приложениях?

Подпись преподавателя _____ **Подпись заведующего кафедрой** _____

7.2. Вопросы на зачет (6 семестр)

1. Какова основная цель использования библиотеки React в разработке веб-приложений?
2. Какие основные преимущества предоставляет JSX в сравнении с обычным JavaScript?
3. Какие методы React используются для управления состоянием компонентов? Приведите примеры их применения.
4. Как осуществляется роутинг и навигация в веб-приложениях на React? Какие инструменты для этого используются?
5. Как происходит стилизация компонентов в React-приложениях? Расскажите о преимуществах подходов CSS Modules и Styled Components.
6. Как обрабатываются формы и события в React-приложениях? Как организовать валидацию форм?
7. Как осуществляется взаимодействие с сервером в React-приложениях? Укажите на основные методы и инструменты.
8. Какие механизмы используются для реализации аутентификации и авторизации в приложениях на React?
9. Какие средства и методы оптимизации производительности могут быть применены в React-приложениях?
10. Какие инструменты и подходы используются для тестирования компонентов и приложений на React?
11. Каковы особенности разработки Single Page Applications (SPA) на React? В чем их преимущества и недостатки?
12. Какие методы и инструменты используются для работы с контекстом и хуками в React-приложениях?
13. Какие паттерны и лучшие практики применяются при проектировании архитектуры приложений на React?
14. Как SSR (Server-Side Rendering) влияет на производительность веб-приложений на React? В каких случаях его стоит использовать?
15. Что такое Next.js и какие возможности он предоставляет для разработки приложений на React?
16. Какие средства и методы используются для тестирования компонентов и приложений на React?
17. Какие аспекты безопасности следует учитывать при разработке и аутентификации в приложениях на React?
18. Какие сценарии оптимизации производительности чаще всего применяются в веб-приложениях на React?
19. Какие особенности работы с API следует учитывать при взаимодействии с сервером в React-приложениях?
20. Какие инструменты и методы используются для управления состоянием приложения в React и обеспечения его целостности?

Образец билета на зачет

**ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. акад. М.Д. Миллионщикова**

Билет № 1

**Дисциплина «Разработка Веб-приложений с применением фреймворков»
Институт ЦЭ и ТП специальность ПИ 6 семестр**

1. Организационные структуры управления.
2. Участники проектной деятельности.
3. Руководство и лидерство.
4. Факторы выбора руководителя проектной деятельности.

УТВЕРЖДЕНО

зав. кафедрой на заседании кафедры

протокол № ____ от _____

Л.Р. Магомаева

Вопросы к первой рубежной аттестации (7 семестр)

1. Какие основные принципы оптимизации производительности следует учитывать при разработке веб-приложений на React?
2. Какие инструменты и методы можно использовать для оценки производительности веб-приложений на React?
3. Какие типичные проблемы производительности могут возникать в веб-приложениях на React и как их можно решить?
4. Какие сценарии тестирования компонентов и приложений на React вы считаете наиболее важными для обеспечения качества кода?
5. Какие основные методы и инструменты используются для тестирования компонентов и приложений на React?
6. Какие преимущества и недостатки связаны с разработкой Single Page Applications (SPA) на React?
7. Какие особенности разработки SPA на React следует учитывать для обеспечения быстрой загрузки и отзывчивости приложения?
8. Как работает контекст в React и какие задачи можно решать с его помощью? Приведите примеры использования контекста.
9. Какие хуки доступны в React и для каких задач они предназначены? Приведите примеры использования хуков.
10. Какие паттерны и лучшие практики применяются при проектировании архитектуры приложений на React?

Образец аттестационного билета (за 1 рубежную аттестацию)

Грозненский государственный нефтяной технический университет

им. акад. М.Д. Миллионщикова

ИЦЭ и ТП

Группа «ПИ» Семестр «_7_»

Дисциплина «Разработка Веб-приложений с применением фреймворков»

Билет № 1

1. Какие основные методы и инструменты используются для тестирования компонентов и приложений на React?
2. Какие преимущества и недостатки связаны с разработкой Single Page Applications (SPA) на React?
3. Какие особенности разработки SPA на React следует учитывать для обеспечения быстрой загрузки и отзывчивости приложения?
4. Как работает контекст в React и какие задачи можно решать с его помощью? Приведите примеры использования контекста.

Подпись преподавателя _____ Подпись заведующего кафедрой _____

Вопросы ко второй рубежной аттестации (7 семестр)

1. Как SSR (Server-Side Rendering) влияет на производительность и SEO веб-приложений на React? В каких случаях его стоит применять?
2. Какие основные преимущества предоставляет SSR в сравнении с клиентским рендерингом?
3. Какие основные принципы и подходы следует учитывать при разработке архитектуры приложений на React с использованием SSR?
4. Какие ключевые особенности и возможности предоставляет фреймворк Next.js для разработки веб-приложений на React?
5. Какие основные задачи можно решать с помощью Next.js и какие преимущества он предоставляет разработчикам?

6. Какие сценарии тестирования компонентов и приложений на React вы считаете наиболее важными для обеспечения качества кода?
7. Какие средства и методы используются для тестирования компонентов и приложений на React?
8. Какие архитектурные паттерны и лучшие практики помогают обеспечить масштабируемость и поддерживаемость приложений на React?
9. Какие сценарии оптимизации производительности чаще всего применяются в веб-приложениях на React?
10. Какие аспекты безопасности следует учитывать при разработке приложений на React с использованием SSR и Next.js?

Образец аттестационного билета (за 2 рубежную аттестацию)

Грозненский государственный нефтяной технический университет

им. акад. М.Д. Миллионщикова

ИЦЭ и ТП

Группа «ПИ» Семестр «_7_»

Дисциплина «Разработка Веб-приложений с применением фреймворков»

Билет № 1

1. Какие основные преимущества предоставляет SSR в сравнении с клиентским рендерингом?
2. Какие основные принципы и подходы следует учитывать при разработке архитектуры приложений на React с использованием SSR?
3. Какие ключевые особенности и возможности предоставляет фреймворк Next.js для разработки веб-приложений на React?
4. Какие основные задачи можно решать с помощью Next.js и какие преимущества он предоставляет разработчикам?

Подпись преподавателя_____ **Подпись заведующего кафедрой**_____

Вопросы на экзамен (7-й семестр)

1. Какие основные принципы оптимизации производительности следует учитывать при разработке веб-приложений на React?
 2. Какие инструменты и методы можно использовать для оценки производительности веб-приложений на React?
 3. Какие типичные проблемы производительности могут возникать в веб-приложениях на React и как их можно решить?
 4. Какие сценарии тестирования компонентов и приложений на React вы считаете наиболее важными для обеспечения качества кода?
 5. Какие основные методы и инструменты используются для тестирования компонентов и приложений на React?
 6. Какие преимущества и недостатки связаны с разработкой Single Page Applications (SPA) на React?
 7. Какие особенности разработки SPA на React следует учитывать для обеспечения быстрой загрузки и отзывчивости приложения?
 8. Как работает контекст в React и какие задачи можно решать с его помощью? Приведите примеры использования контекста.
 9. Какие хуки доступны в React и для каких задач они предназначены? Приведите примеры использования хуков.
 10. Какие паттерны и лучшие практики применяются при проектировании архитектуры приложений на React?
 11. Как SSR (Server-Side Rendering) влияет на производительность и SEO веб-приложений на React? В каких случаях его стоит применять?
 12. Какие основные преимущества предоставляет SSR в сравнении с клиентским рендерингом?
 13. Какие основные принципы и подходы следует учитывать при разработке архитектуры приложений на React с использованием SSR?
 14. Какие ключевые особенности и возможности предоставляет фреймворк Next.js для разработки веб-приложений на React?
 15. Какие основные задачи можно решать с помощью Next.js и какие преимущества он предоставляет разработчикам?
 16. Какие сценарии тестирования компонентов и приложений на React вы считаете наиболее важными для обеспечения качества кода?
 17. Какие средства и методы используются для тестирования компонентов и приложений на React?
 18. Какие архитектурные паттерны и лучшие практики помогают обеспечить масштабируемость и поддерживаемость приложений на React?
 19. Какие сценарии оптимизации производительности чаще всего применяются в веб-приложениях на React?
 20. Какие аспекты безопасности следует учитывать при разработке приложений на React с использованием SSR и Next.js?
-

**ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. акад. М.Д. Миллионщикова**

Билет № 1

Дисциплина «Разработка Веб-приложений с применением фреймворков»

Институт ЦЭ и ТП специальность ПИ 7 семестр

1. Какие сценарии тестирования компонентов и приложений на React вы считаете наиболее важными для обеспечения качества кода?
2. Какие средства и методы используются для тестирования компонентов и приложений на React?
3. Какие архитектурные паттерны и лучшие практики помогают обеспечить масштабируемость и поддерживаемость приложений на React?
4. Какие сценарии оптимизации производительности чаще всего применяются в веб-приложениях на React?

УТВЕРЖДЕНО

зав. кафедрой

на заседании кафедры

протокол № ____ от _____

Л.Р. Магомаева

7.3. Текущий контроль

В качестве оценочных средств текущего контроля используются средства контроля выполнения и защиты лабораторных работ по дисциплине. Защита лабораторной работы – ответ на контрольные вопросы после выполнения лабораторной работы.

6-й семестр

Лабораторная работа 1.
Основы работы с React.

Цель лабораторной работы: Ознакомиться с базовыми принципами работы с библиотекой React, научиться создавать простые компоненты и использовать JSX синтаксис.

Лабораторная работа 2.
Управление состоянием в React.

Цель лабораторной работы: Изучить методы управления состоянием компонентов в React с помощью хуков `useState` и `useEffect`, научиться обновлять состояние и реагировать на изменения.

Лабораторная работа 3.
Роутинг и навигация в React.

Цель лабораторной работы: Познакомиться с библиотекой React Router и научиться создавать маршруты для навигации между страницами в веб-приложении на React.

Лабораторная работа 4.
Стилизация компонентов в React.

Цель лабораторной работы: Изучить различные способы стилизации компонентов в React, включая CSS Modules и Styled Components, научиться создавать стилизованные интерфейсы.

Лабораторная работа 5.
Формы и обработка событий в React.

Цель лабораторной работы: Практически изучить создание и валидацию форм в React, научиться обрабатывать события в компонентах для реализации интерактивности.

Лабораторная работа 6.
Взаимодействие с сервером в React.

Цель лабораторной работы: Освоить основы взаимодействия с сервером в React-приложениях с использованием Fetch API для отправки запросов на сервер и получения данных.

7-й семестр

Лабораторная работа 7.
Оптимизация производительности в React.

Цель лабораторной работы: Изучить методы оптимизации производительности React-приложений, включая мемоизацию, `useMemo` и `useCallback`, и применить их на практике.

Лабораторная работа 8.
Тестирование компонентов и приложений на React.

Цель лабораторной работы: Научиться писать модульные и интеграционные тесты для React-компонентов и приложений с использованием библиотеки Jest.

Лабораторная работа 9.
Разработка SPA на React.

Цель лабораторной работы: Практически освоить разработку Single Page Applications (SPA) на React, организовать структуру приложения и управлять его состоянием.

Лабораторная работа 10.
Работа с контекстом и хуками в React.

Цель лабораторной работы: Изучить контекст и пользовательские хуки в React и применить их для передачи данных между компонентами и управления состоянием.

Лабораторная работа 11.
Архитектура приложений на React.

Цель лабораторной работы: Разработать приложение с использованием лучших практик и паттернов, организовать его архитектуру и структуру компонентов.

Лабораторная работа 12. *Цель лабораторной работы: Изучить Server-Side SSR и применение в React. Rendering (SSR) и его применение в React-приложениях для улучшения производительности и SEO.*

Образец выполненной лабораторной работы

Лабораторная работа 1. Основы работы с React

Цель работы: Ознакомиться с базовыми принципами работы с библиотекой React, научиться создавать простые компоненты и использовать JSX синтаксис.

Описание работы:

- Установка и настройка проекта
- Создание нового проекта:
- Убедитесь, что у вас установлены Node.js и npm.
- В командной строке выполните команду для создания нового проекта с помощью Create React App:

```
«npx create-react-app my-first-react-app»
```

Перейдите в созданную директорию проекта:

```
«cd my-first-react-app»
```

Запустите проект:

```
«npm start»
```

В браузере откроется страница с приветственным сообщением от Create React App.

Создание простого компонента

- **Создание компонента:**
 - Откройте проект в вашем любимом редакторе кода (например, VS Code).
 - В директории src создайте файл HelloWorld.js.
 - Добавьте в HelloWorld.js следующий код для создания простого компонента

```
import React from 'react';

function HelloWorld() {
  return (
    <div>
      <h1>Hello, World!</h1>
    </div>
  );
}

export default HelloWorld;
```

Использование компонента:

- Откройте файл `App.js`.
- Импортируйте компонент `HelloWorld`:

```
import React from 'react';
import HelloWorld from './HelloWorld';

function App() {
  return (
    <div className="App">
      <HelloWorld />
    </div>
  );
}

export default App;
```

Использование JSX синтаксиса

- **Изучение JSX:**
 - В React компоненты пишутся с использованием JSX — синтаксиса, который позволяет писать HTML-подобный код внутри JavaScript.
 - В компоненте `HelloWorld` уже используется JSX для отображения заголовка `<h1>`.
- **Добавление элементов с использованием JSX:**
 - Обновите компонент `HelloWorld`, чтобы он отображал дополнительные элементы:

```
import React from 'react';

function HelloWorld() {
  return (
    <div>
      <h1>Hello, World!</h1>
      <p>Welcome to your first React component.</p>
      <ul>
        <li>Learn JSX syntax</li>
        <li>Create React components</li>
        <li>Build amazing UIs</li>
      </ul>
    </div>
  );
}
```

Стилизация компонентов

- **Добавление CSS:**

- Создайте файл `HelloWorld.css` в директории `src`.
- Добавьте стили для компонента `HelloWorld`:

```
.hello-world {  
  text-align: center;  
  color: #333;  
}
```

Применение стилей к компоненту:

- Обновите файл `HelloWorld.js` для использования CSS:

```
import React from 'react';  
import './HelloWorld.css';  
  
function HelloWorld() {  
  return (  
    <div className="hello-world">  
      <h1>Hello, World!</h1>  
      <p>Welcome to your first React component.</p>  
      <ul>  
        <li>Learn JSX syntax</li>  
        <li>Create React components</li>  
        <li>Build amazing UIs</li>  
      </ul>  
    </div>  
  );  
}  
  
export default HelloWorld;
```

Композиция компонентов

- **Создание нового компонента:**

- В директории `src` создайте файл `ListItem.js`.
- Добавьте в `ListItem.js` следующий код:

```
import React from 'react';  
  
function ListItem(props) {  
  return <li>{props.item}</li>;  
}  
  
export default ListItem;
```

Использование компонента `ListItem` в компоненте `HelloWorld`:

- Обновите файл `HelloWorld.js`:

```
import React from 'react';
import './HelloWorld.css';
import ListItem from './ListItem';

function HelloWorld() {
  const items = ['Learn JSX syntax', 'Create React components', 'Build amazing UIs'];

  return (
    <div className="hello-world">
      <h1>Hello, World!</h1>
      <p>Welcome to your first React component.</p>
      <ul>
        {items.map((item, index) => (
          <ListItem key={index} item={item} />
        ))}
      </ul>
    </div>
  );
}

export default HelloWorld;
```

Работа с состоянием компонентов

- Создание компонента с состоянием:
 - В директории `src` создайте файл `Counter.js`.
 - Добавьте в `Counter.js` следующий код:

```
import React, { useState } from 'react';

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Current count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
      <button onClick={() => setCount(count - 1)}>Decrement</button>
    </div>
  );
}

export default Counter;
```

Использование компонента Counter в компоненте HelloWorld:

- Обновите файл `HelloWorld.js`:

```
import React from 'react';
import './HelloWorld.css';
import ListItem from './ListItem';
import Counter from './Counter';

function HelloWorld() {
  const items = ['Learn JSX syntax', 'Create React components', 'Build amazing UIs'];

  return (
    <div className="hello-world">
      <h1>Hello, World!</h1>
      <p>Welcome to your first React component.</p>
      <ul>
        {items.map((item, index) => (
          <ListItem key={index} item={item} />
        ))}
      </ul>
      <Counter />
    </div>
  );
}

export default HelloWorld;
```

Работа с формами и обработкой событий

- Создание компонента формы:
 - В директории `src` создайте файл `Form.js`.
 - Добавьте в `Form.js` следующий код:

```
import React, { useState } from 'react';

function Form() {
  const [inputValue, setInputValue] = useState('');

  const handleSubmit = (event) => {
    event.preventDefault();
    alert(`Submitted value: ${inputValue}`);
  };

  return (
    <form onSubmit={handleSubmit}>
      <label>
        Input:
        <input
          type="text"
          value={inputValue}
          onChange={e => setInputValue(e.target.value)}
        />
      </label>
      <button type="submit">Submit</button>
    </form>
  );
}
```

```
        </form>
      );
}

export default Form;
```

ИСПОЛЬЗОВАНИЕ КОМПОНЕНТА `Form` В КОМПОНЕНТЕ `HelloWorld`:

- Обновите файл `HelloWorld.js`:

```
import React from 'react';
import './HelloWorld.css';
import ListItem from './ListItem';
import Counter from './Counter';
import Form from './Form';

function HelloWorld() {
  const items = ['Learn JSX syntax', 'Create React components', 'Build amazing UIs'];

  return (
    <div className="hello-world">
      <h1>Hello, World!</h1>
      <p>Welcome to your first React component.</p>
      <ul>
        {items.map((item, index) => (
          <ListItem key={index} item={item} />
        ))}
      </ul>
      <Counter />
      <Form />
    </div>
  );
}

export default HelloWorld;
```

Использование эффектов

- Создание компонента с эффектом:
 - В директории `src` создайте файл `DataFetcher.js`.
 - Добавьте в `DataFetcher.js` следующий код:

```

import React, { useState, useEffect } from 'react';

function DataFetcher() {
  const [data, setData] = useState(null);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    fetch('https://jsonplaceholder.typicode.com/posts/1')
      .then((response) => response.json())
      .then((data) => {
        setData(data);
        setLoading(false);
      });
  }, []);

  if (loading) {
    return <p>Loading...</p>;
  }

  return (
    <div>
      <h3>{data.title}</h3>
      <p>{data.body}</p>
    </div>
  );
}

export default DataFetcher;

```

ИСПОЛЬЗОВАНИЕ КОМПОНЕНТА DataFetcher В КОМПОНЕНТЕ HelloWorld:

- Обновите файл HelloWorld.js:

```

import React from 'react';
import './HelloWorld.css';
import ListItem from './ListItem';
import Counter from './Counter';
import Form from './Form';
import DataFetcher from './DataFetcher';
function HelloWorld() {

  const items = ['Learn JSX syntax', 'Create React components', 'Build amazing UIs'];
  return (
    <div className="hello-world">
      <h1>Hello, World!</h1>
      <p>Welcome to your first React component.</p>
      <ul>
        {items.map((item, index) => (
          <ListItem key={index} item={item} />
        ))}
      </ul>
      <Counter />
      <Form />
    </div>
  );
}

```

```
        <DataFetcher />
      </div>
    );
  }
export default HelloWorld;
```

Реализация маршрутизации

- **Установка react-router-dom:**
 - В командной строке выполните команду:

```
npm install react-router-dom
```

Настройка маршрутизации:

- Обновите файл App.js:

```
import React from 'react';
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';
import HelloWorld from './HelloWorld';
import Counter from './Counter';
import Form from './Form';
import DataFetcher from './DataFetcher';

function App() {
  return (
    <Router>
      <div className="App">
        <nav>
          <ul>
            <li>
              <Link to="/">Home</Link>
            </li>
            <li>
              <Link to="/counter">Counter</Link>
            </li>
            <li>
              <Link to="/form">Form</Link>
            </li>
            <li>
              <Link to="/data-fetcher">Data Fetcher</Link>
            </li>
          </ul>
        </nav>
        <Switch>
          <Route path="/" exact component={HelloWorld} />
          <Route path="/counter" component={Counter} />
          <Route path="/form" component={Form} />
          <Route path="/data-fetcher" component={DataFetcher} />
        </Switch>
      </div>
    </Router>
  );
}

export default App;
```

Создание отдельных компонентов для маршрутов:

- Обновите файл `HelloWorld.js` для использования в качестве домашней страницы:

```
import React from 'react';
import './HelloWorld.css';
import ListItem from './ListItem';

function HelloWorld() {
  const items = ['Learn JSX syntax', 'Create React components', 'Build amazing UIs'];

  return (
    <div className="hello-world">
      <h1>Hello, World!</h1>
      <p>Welcome to your first React component.</p>
      <ul>
        {items.map((item, index) => (
          <ListItem key={index} item={item} />
        ))}
      </ul>
    </div>
  );
}

export default HelloWorld;
```

Используйте ранее созданные компоненты для маршрутов `Counter`, `Form`, и `DataFetcher`.

Заключение

- **Рефакторинг кода:**
 - Проверьте, что все компоненты организованы и структурированы.
 - Убедитесь, что нет дублирующегося кода и все компоненты переиспользуются.
- **Тестирование и отладка:**
 - Проверьте работу всех компонентов и маршрутов.
 - Убедитесь, что приложение работает без ошибок и предупреждений в консоли.
- **Завершение работы:**
- Остановите проект, выполнив команду:

Ctrl + C

Сохраните все изменения и закройте проект.

7.4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания.

Таблица 7

Планируемые результаты освоения компетенции	Критерии оценивания результатов обучения				Наименование оценочного средства
	менее 41 баллов (неудовлетворительно)	41-60 баллов (удовлетворительно)	61-80 баллов (хорошо)	81-100 баллов (отлично)	
ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.					
Знать: принципы работы с отечественным программным обеспечением и информационными технологиями.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	задания для лабораторной работы, билеты рубежных аттестаций, темы докладов
Уметь: применять современные информационные технологии и программные средства для решения задач профессиональной деятельности.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: навыками работы с современными информационными технологиями и программными средствами на уровне, необходимом для эффективного выполнения профессиональных задач.	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

ПК – 5. Способен разрабатывать и адаптировать прикладное программное обеспечение

Знать: библиотеку React для разработки веб-приложений и основные понятия, такие как компоненты, JSX, Virtual DOM и управление состоянием.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>
Уметь: создавать интерактивные веб-приложения с использованием языка JavaScript и библиотеки React.	Частичные умения	Неполные умения	Умения полные, допускаются небольшие ошибки	Сформированные умения	
Владеть: умением эффективно использовать библиотеку React для создания масштабируемых и легко поддерживаемых веб-приложений	Частичное владение навыками	Несистематическое применение навыков	В систематическом применении навыков допускаются пробелы	Успешное и систематическое применение навыков	

8. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся созданы фонды оценочных средств, адаптированные для инвалидов и лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе. Форма проведения текущей аттестации для студентов-инвалидов устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При тестировании для слабовидящих студентов используются фонды оценочных средств с укрупненным шрифтом.

На экзамен приглашается сопровождающий, который обеспечивает техническое сопровождение студенту. При необходимости студенту-инвалиду предоставляется дополнительное время для подготовки ответа на экзамене (или зачете). Обучающиеся с ограниченными возможностями здоровья и обучающиеся инвалиды обеспечиваются печатными и электронными образовательными ресурсами (программы, учебные пособия для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

1) для инвалидов и лиц с ограниченными возможностями здоровья **по зрению:**

- **для слепых:** задания для выполнения на семинарах и практических занятиях оформляются рельефно-точечным шрифтом Брайля или в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом; письменные задания выполняются на бумаге рельефно-точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых либо надиктовываются ассистенту; обучающимся для выполнения задания при необходимости предоставляется комплект письменных принадлежностей и бумага для письма рельефно-точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;

- **для слабовидящих:** обеспечивается индивидуальное равномерное освещение не менее 300 люкс; обучающимся для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; задания для выполнения заданий оформляются увеличенным шрифтом;

2) для инвалидов и лиц с ограниченными возможностями здоровья **по слуху:**

- для **глухих и слабослышащих:** обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования; предоставляются услуги сурдопереводчика;

- для **слепоглухих** допускается присутствие ассистента, оказывающего услуги тифлосурдопереводчика (помимо требований, выполняемых соответственно для слепых и глухих);

3) для лиц с тяжелыми нарушениями речи, глухих, слабослышащих лекции и семинары, проводимые в устной форме, проводятся в письменной форме;

4) для инвалидов и лиц с ограниченными возможностями здоровья, **имеющих нарушения опорно-двигательного аппарата:**

- для лиц с нарушениями опорно-двигательного аппарата, нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей: письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту; выполнение заданий (тестов, контрольных работ), проводимые в письменной форме, проводятся в устной форме путем опроса, беседы с обучающимся.

9. Учебно-методическое и информационное обеспечение дисциплины

9.1. Литература

1. Хортон, А. Разработка веб-приложений в ReactJS / Хортон А. , Вайс Р. - Москва : ДМК Пресс, 2016. - 254 с. - ISBN 978-5-94074-819-9. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785940748199.html> (дата обращения: 04.04.2024). - Режим доступа : по подписке.
2. Кириченко, А. В. JavaScript для FrontEnd-разработчиков. Написание. Тестирование. Развертывание / А. В. Кириченко. — Санкт-Петербург : Наука и Техника, 2020. — 320 с. — ISBN 978-5-94387-789-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/175389> (дата обращения: 04.04.2024). — Режим доступа: для авториз. пользователей.

9.2. Методические указания для обучающихся по освоению дисциплины (приложение)

10. Материально-техническое обеспечение дисциплины

10.1. Материально-техническая база

Лекционная аудитория, оснащенная компьютером, видеопроекционным оборудованием, в том числе для презентаций, средствами звуковоспроизведения, экраном. Мультимедийные средства и другая техника для презентаций учебного материала, офисный пакет программ MS Windows (MS Excel, MSWord) для оформления расчетов экономической эффективности информационных систем, OpenOfficeGoogleChrome, редактор кода (Visual Studio Code)

10.2. Помещения для самостоятельной работы

Помещение для самостоятельной работы (Главный учебный корпус ФГБОУ ВО «Грозненский государственный нефтяной технический университет» 364902, Чеченская республика, г. Грозный, проспект им. Х.А. Исаева, 100. Аудитория оснащена необходимой компьютерной техникой, в наличии есть необходимое ПО: WinPro 10 RUS Upgrd OLP NL Acdmc; OfficeStd RUS OLP NL Acdmc (право на использование согласно Контракту № 267-ЭА/19 от 15.09.2019 г.).

Система ГАРАНТ (проприетарная лицензия) Visual Studio-(Freemium) 1С Предприятие договор от 02.12.2020 регистрационные номера продуктов (9334859; 9334952) Sublime Text-(открытый доступ) Notepad++ (открытый доступ).

Методические указания по освоению дисциплины «Разработка Веб-приложений с применением фреймворков»

1. Методические указания для обучающихся по планированию и организации времени, необходимого для освоения дисциплины.

Изучение рекомендуется начать с ознакомления с рабочей программой дисциплины, ее структурой и содержанием разделов (модулей), фондом оценочных средств, ознакомиться с учебно-методическим и информационным обеспечением дисциплины.

Дисциплина «Разработка Веб-приложений с применением фреймворков» состоит из 17 связанных между собой тем, обеспечивающих последовательное изучение материала.

Обучение по дисциплине «Разработка Веб-приложений с применением фреймворков» осуществляется в следующих формах:

1. Аудиторные занятия (лекции, практические занятия).
2. Самостоятельная работа студента (подготовка к лекциям, лабораторным занятиям, рефератам__и иным формам письменных работ, индивидуальная консультация с преподавателем).
3. Интерактивные формы проведения занятий (лекция-дискуссия, групповое решение кейса и др. формы).

Учебный материал структурирован и изучение дисциплины производится в тематической последовательности. Каждому практическому занятию и самостоятельному изучению материала предшествует лекция по данной теме. Обучающиеся самостоятельно проводят предварительную подготовку к занятию, принимают активное и творческое участие в обсуждении теоретических вопросов, разборе проблемных ситуаций и поисков путей их решения. Многие проблемы, изучаемые в курсе, носят дискуссионный характер, что предполагает интерактивный характер проведения занятий на конкретных примерах.

Описание последовательности действий обучающегося:

При изучении курса следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий:

1. После окончания учебных занятий для закрепления материала просмотреть и обдумать текст лекции, прослушанной сегодня, разобрать рассмотренные примеры (10 – 15 минут).
2. При подготовке к лекции следующего дня повторить текст предыдущей лекции, подумать о том, какая может быть следующая тема (10 - 15 минут).
3. В течение недели выбрать время для работы с литературой в библиотеке (по 1 часу).
4. При подготовке к практическому занятию повторить основные понятия по теме, изучить примеры. Решая конкретную ситуацию, - предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить лабораторные задания.

2. Методические указания по работе обучающихся во время проведения лекций.

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах. Лекции обычно излагаются в традиционном или в проблемном стиле. Для студентов в большинстве случаев в проблемном стиле. Проблемный стиль позволяет стимулировать активную познавательную деятельность обучающихся и их интерес к дисциплине, формировать творческое мышление, прибегать к противопоставлениям и сравнениям, делать обобщения, активизировать внимание обучающихся путем постановки проблемных вопросов, поощрять дискуссию.

Во время лекционных занятий рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления, или процессов, выводы и практические рекомендации.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает преподаватель, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, необходимо использовать не только основную литературу, но и ту литературу, которую дополнительно рекомендовал преподаватель. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

Тематика лекций дается в рабочей программе дисциплины.

3. Методические указания обучающимся по подготовке к лабораторным/семинарским занятиям.

На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике семинарских занятий.

Студенту рекомендуется следующая схема подготовки к семинарскому занятию:

1. Ознакомление с планом практического занятия, который отражает содержание предложенной темы;

2. Проработать конспект лекций;

3. Прочитать основную и дополнительную литературу.

В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов отношение к конкретной проблеме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса;

4. Ответить на вопросы плана практического занятия;

5. Выполнить домашнее задание;

6. Проработать тестовые задания и задачи;

7. При затруднениях сформулировать вопросы к преподавателю.

Результат такой работы должен проявиться в способности студента свободно ответить

на теоретические вопросы практикума, выступать и участвовать в коллективном обсуждении вопросов изучаемой темы, правильно выполнять практические задания и иные задания, которые даются в фонде оценочных средств дисциплины.

4. Методические указания обучающимся по организации самостоятельной работы.

Цель организации самостоятельной работы по дисциплине «Разработка Веб-приложений с применением фреймворков» — это углубление и расширение знаний в области гуманитарных наук; формирование навыка и интереса к самостоятельной познавательной деятельности.

Самостоятельная работа обучающихся является важнейшим видом освоения содержания дисциплины, подготовки к практическим занятиям и к контрольной работе. Сюда же относятся и самостоятельное углубленное изучение тем дисциплины. Самостоятельная работа представляет собой постоянно действующую систему, основу образовательного процесса и носит исследовательский характер, что послужит в будущем основанием для написания выпускной квалификационной работы, практического применения полученных знаний.

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению, с учетом потребностей и возможностей личности.

Правильная организация самостоятельных учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет студентам развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивать высокий уровень успеваемости в период обучения, получить навыки повышения профессионального уровня.

Подготовка к лабораторному занятию включает, кроме проработки конспекта и презентации лекции, поиск литературы (по рекомендованным спискам и самостоятельно), решение задач из перечня лабораторных работ (текущий контроль). Лабораторное занятие – выполнение поставленных перед студентом задач с использованием ПК и специального программного обеспечения.

При подготовке к контрольной работе обучающийся должен повторять пройденный материал в строгом соответствии с учебной программой, используя конспект лекций и литературу, рекомендованную преподавателем. При необходимости можно обратиться за консультацией и методической помощью к преподавателю.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий - на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания - на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Виды СРС и критерии оценок

(по балльно-рейтинговой системе ГГНТУ, СРС оценивается в 15 баллов)

1. Реферат/Доклад/Презентация
2. Индивидуальное задание

Темы для самостоятельной работы прописаны в рабочей программе дисциплины. Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

Составитель:

Ст. преподаватель каф. «ИСЭ»



/Абдулаев М.К./

СОГЛАСОВАНО:

Зав. выпускающей каф. «ИСЭ»



/Магомаева Л.Р./

Директор ДУМР



/Магомаева М.А./