
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол №2

**Рабочая программа дисциплины (модуля)
«Промышленная фронтенд-разработка»**

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль) подготовки: Бизнес и аналитика

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	6
3. Тематический план	8
4. Содержание дисциплины (модуля)	9
5. Учебно-методическое обеспечение	10
6. Материально-техническое обеспечение	10
7. Методические и оценочные материалы	12

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Промышленная фронтенд-разработка» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 38.03.05 Бизнес-информатика, профиль Бизнес и аналитика, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 838 от 29.07.2020 года.

Изучение дисциплины (модуля) «Промышленная фронтенд-разработка» позволяет подготовить специалистов к участию в реальных промышленных проектах, где требуется не только техническая грамотность, но и понимание архитектуры, безопасности и процессов доставки ПО. Полученные знания и навыки обеспечивают быструю адаптацию в команде разработки и эффективный вклад в создание качественных веб-приложений.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 38.03.05 Бизнес-информатика, профиль Бизнес и аналитика и входит в часть, формируемую участниками образовательных отношений, Блока 1, как дисциплина по выбору.

Дисциплина (модуль) изучается на 3 или 4 курсе в 6,7 или 8 семестре на выбор. Доступна к изучению после успешного освоения дисциплины (модуля): «Основы промышленной разработки».

Цель изучения дисциплины (модуля): заключается в формировании навыков проектирования, разработки и сопровождения масштабируемых, производительных и безопасных фронтенд-приложений на React и TypeScript в соответствии с современными промышленными стандартами.

Задачи изучения дисциплины (модуля):

- изучить принципы компонентного подхода, управления состоянием и жизненным циклом компонентов в React;
- освоить современные практики клиент-серверного взаимодействия, роутинга и защиты пользовательских сценариев в SPA;
- развить навыки архитектурной организации кода и обеспечения поддерживаемости фронтенд-приложений;
- научиться писать тесты на уровне функций, компонентов и сквозных пользовательских сценариев;
- освоить инструменты сборки, деплоя, мониторинга и оптимизации производительности в production-среде.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- основные принципы разработки промышленных фронтенд-приложений на React и TypeScript;
- принципы компонентного подхода, декомпозиции интерфейса и управления жизненным циклом компонентов;
- механизмы управления состоянием клиентского приложения и различия между локальным, глобальным и серверным состоянием;
- подходы к организации роутинга, навигации и защищённых пользовательских сценариев в SPA;
- принципы клиент-серверного взаимодействия, обработки сетевых состояний, кэширования и синхронизации данных;
- основные угрозы безопасности фронтенд-приложений и подходы к защите пользовательских данных;

- подходы к архитектурной организации React-приложений и обеспечению поддерживаемости кода;
- принципы тестирования фронтенд-приложений на уровне функций, компонентов и пользовательских сценариев;
- основы серверного рендеринга, различия между CSR, SSR, SSG и другими моделями рендеринга;
- подходы к сборке, деплою, мониторингу и оптимизации производительности фронтенд-приложений.

уметь:

- создавать и настраивать React-приложение с использованием TypeScript и современных инструментов разработки;
- разрабатывать переиспользуемые компоненты пользовательского интерфейса и декомпозировать сложные экраны;
- использовать хуки React для управления состоянием, побочными эффектами, формами и пользовательским взаимодействием;
- организовывать состояние приложения с учётом масштаба проекта и характера данных;
- реализовывать маршрутизацию, защищённые маршруты и основные пользовательские сценарии SPA;
- подключать фронтенд-приложение к серверному API, обрабатывать загрузку, ошибки, кэширование и обновление данных;
- реализовывать базовые механизмы авторизации и учитывать требования безопасности при работе с пользовательскими данными;
- структурировать фронтенд-проект с разделением ответственности между компонентами, страницами, слоями и модулями;
- писать unit-, компонентные и E2E-тесты для ключевой логики и пользовательских сценариев;
- анализировать производительность приложения, выявлять лишние перерисовки, проблемы сборки и узкие места интерфейса;
- использовать SSR/Next.js для решения задач серверного рендеринга и сравнивать его с клиентским рендерингом;
- подготавливать фронтенд-приложение к публикации, настраивать сборку, мониторинг и базовые проверки качества.

владеть:

- инструментами промышленной фронтенд-разработки на React, TypeScript и современной JavaScript-экосистеме;
- навыками проектирования и разработки поддерживаемых компонентных интерфейсов;
- подходами к управлению состоянием, роутингом и клиент-серверным взаимодействием в сложных SPA;
- методами обеспечения безопасности пользовательского интерфейса и клиентской логики;
- подходами к архитектурной организации фронтенд-приложений и рефакторингу растущего проекта;
- навыками тестирования фронтенд-приложений на разных уровнях: от отдельных функций до сквозных пользовательских сценариев;
- инструментами отладки, профилирования и анализа производительности React-приложений;
- подходами к реализации серверного рендеринга и выбору подходящей модели рендеринга для проекта;

- методами подготовки фронтенд-приложения к production-среде: сборка, деплой, мониторинг, анализ качества и оптимизация;
- навыками анализа, объяснения и улучшения кода фронтенд-приложения на уровне, достаточном для участия в промышленной разработке.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности.
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем.
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации.
ОПК-2.	Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ОПК-2.1.	Знает основные тенденции и характеристики рынка информационных систем и информационно-коммуникационных технологий
		ОПК-2.2.	Умеет проводить исследование и анализ рыночной информации для оценки потребностей бизнеса и выбора оптимальных решений
		ОПК-2.3.	Имеет практический опыт в разработке и внедрении стратегий управления бизнесом на основе анализа рынка информационных технологий
ПК-2.	Способен использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования для решения задач профессиональной деятельности	ПК-2.1.	Знает основные математические методы и инструментальные средства, применяемые для обработки и анализа информации
		ПК-2.2.	Умеет эффективно использовать математический аппарат для систематизации данных и решения профессиональных задач
		ПК-2.3.	Имеет практический опыт работы с инструментами анализа информации в рамках исследовательских проектов

ПК-3.	Способен готовить научно-технические отчеты, презентации, научные публикации по результатам выполненных исследований	ПК-3.1.	Знает требования и стандарты оформления научно-технических отчетов, презентаций и публикаций
		ПК-3.2.	Умеет структурировать и представлять результаты исследований в ясной и доступной форме
		ПК-3.3.	Имеет практический опыт подготовки и публикации научных материалов, отражающих результаты выполненных исследований
ПК-8.	Способен под руководством специалиста более высокой категории осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-8.1.	Знает принципы и стандарты управления проектами
		ПК-8.2.	Умеет разрабатывать планы и организовывать проектную деятельность в соответствии с установленными стандартами
		ПК-8.3.	Имеет практический опыт участия в проектной работе, включая планирование и координацию задач

3. Тематический план

№п/ п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары			
1	Основы промышленной разработки на React	10	10		40	Домашние задания, Контрольная работа
2	Архитектура клиентского приложения	10	10	2	40	Домашние задания, Контрольная работа
3	Качество и сопровождение фронтенд-приложения	6	6	2	24	Домашние задания
4	Серверный рендеринг и production-подходы	4	4	4	14	Домашние задания, Проект
	Экзамен			4		
	Итого	30	30	12	118	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы промышленной разработки на React	Введение в React, компонентный подход и настройка проекта Состояние, хуки и жизненный цикл React-компонентов Формы, рефы, кастомные хуки и базовая оптимизация компонентов Инструменты разработки, профилирование и настройка production-сборки
2	Архитектура клиентского приложения	Управление состоянием в React-приложении Клиент-серверное взаимодействие и управление серверным состоянием Роутинг, навигация и организация пользовательских сценариев Безопасность фронтенд-приложений и реализация пользовательской авторизации
3	Качество и сопровождение фронтенд-приложения	Модульное и компонентное тестирование React-приложений E2E-тестирование и стратегия тестирования пользовательских сценариев Продвинутое паттерны React, архитектура проекта и доступность интерфейсов
4	Серверный рендеринг и production-подходы	Next.js и серверный рендеринг фронтенд-приложений Деплой, мониторинг и оптимизация фронтенд-приложения

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Хорстманн, К. С. Современный JavaScript для нетерпеливых : практическое руководство / К. С. Хорстманн ; пер. с англ. А. А. Слинкина. - Москва : ДМК Пресс, 2021. - 288 с. - ISBN 978-5-97060-177-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2109482>.

2. Черный, Б. Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений / Б. Черный. - Санкт-Петербург : Питер, 2021. - 352 с. - ISBN 978-5-4461-1651-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2142477>

Дополнительная литература:

1. Вандеркам, Д. Эффективный TypeScript: 62 способа улучшить код : практическое руководство / Д. Вандеркам. - Санкт-Петербург : Питер, 2021. - 288 с. - (Серия «Бестселлеры O'Reilly»). - ISBN 978-5-4461-1623-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1733517>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том

числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robot Operating System	зарубежное	свободно распространяемое

CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Промышленная фронтенд-разработка» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, контрольные работы, проекты, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения

каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Промышленная фронтенд-разработка»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *экзамена*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Промышленная фронтенд-разработка» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	20%	8	Набор задач по темам недели
Контрольная работа	30%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Проект	20%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Экзамен	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по курсу

Формула расчёта итоговой оценки по дисциплине (модулю) «Промышленная фронтенд-разработка»: $\langle 0,2 \times \text{среднее за домашние задания} + 0,2 \times \text{проект} + 0,3 \times \text{среднее за контрольные работы} + 0,3 \times \text{экзамен} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1.

1. Установите Node.js и создайте React-приложение с помощью create-react-app или Vite с поддержкой TypeScript. Опишите, зачем нужны эти инструменты и как они упрощают разработку.

2. Создайте компонент Header, Navbar, Main, Footer с использованием JSX. Реализуйте компонент NewsCard, отображающий заголовок, краткое описание и дату. Объясните, как работает компонентный подход и зачем нужна декомпозиция интерфейса.

3. Напишите функциональный компонент Counter, используя хук useState. Добавьте кнопки для увеличения и сброса счётчика. Объясните, как хуки заменяют жизненный цикл классовых компонентов.

4. Создайте кастомный хук useLocalStorage, который сохраняет значение в localStorage и синхронизирует его между сессиями. Используйте его в компоненте

ThemeSwitcher. Объясните, зачем нужны кастомные хуки и как они способствуют переиспользованию логики.

5. Настройте ESLint и Prettier в проекте. Проанализируйте код с помощью React DevTools и Profiler. Объясните, как инструменты помогают выявлять лишние перерисовки и улучшать производительность.

Домашнее задание 2.

1. Реализуйте управление состоянием приложения с помощью Context API или useReducer: создайте глобальное состояние для корзины товаров. Объясните, в каких случаях использовать локальное, контекстное и внешнее состояние.

2. Подключите приложение к публичному API (например, <https://fakestoreapi.com/products>). Используйте useEffect и fetch для загрузки данных. Реализуйте отображение состояний "загрузка", "ошибка", "данные получены". Объясните, как управлять серверным состоянием вручную.

3. Настройте маршрутизацию с помощью React Router. Создайте страницы: Главная, Каталог, Карточка товара, Корзина. Реализуйте защищённый маршрут (например, "Личный кабинет"), доступный только при авторизации. Объясните, как организуется навигация в SPA.

4. Реализуйте форму входа: ввод логина и пароля, проверка на стороне клиента, симуляция авторизации (например, через localStorage). Добавьте защиту от XSS: экранируйте вывод пользовательских данных. Объясните, какие угрозы существуют на фронтенде и как их минимизировать.

5. Оптимизируйте рендеринг: используйте React.memo, useCallback, useMemo для предотвращения лишних перерисовок в списке товаров. Объясните, как эти хуки помогают в производительности.

Домашнее задание 3.

1. Напишите unit-тесты с помощью Jest и @testing-library/react для компонента NewsCard и хука useLocalStorage. Проверьте корректность рендеринга и поведения. Объясните, зачем нужны модульные тесты и какие сценарии они покрывают.

2. Создайте компонент формы UserProfileForm и напишите компонентные тесты: проверьте валидацию, изменение полей, отправку формы. Используйте моки для имитации обработки события. Объясните, чем компонентное тестирование отличается от unit-тестирования.

3. Реализуйте E2E-тест с помощью Cypress: автоматизируйте сценарий "пользователь заходит на сайт, переходит в каталог, добавляет товар в корзину, переходит в корзину". Опишите, как E2E-тесты помогают проверять сквозные пользовательские сценарии.

4. Перепишите структуру проекта по архитектурному подходу (например, Feature-Sliced Design или Atomic Design): разделите код на папки components, features, entities, shared. Объясните, как такая организация улучшает поддерживаемость кода.

5. Проверьте доступность интерфейса с помощью axe или Lighthouse: найдите и исправьте ошибки (например, отсутствие alt у изображений, несемантические теги). Объясните, почему доступность важна и как её обеспечить в React-приложении.

Примерные задания для контрольной работы

Контрольная работа 1.

1. Объясните, что такое компонентный подход в React. Опишите различия между функциональными и классовыми компонентами и преимущества функциональных компонентов с хуками.

2. Создайте функциональный компонент UserProfile, который отображает имя и возраст пользователя. Используйте хук useState для реализации кнопки "Показать/скрыть возраст". Объясните, как useState управляет состоянием компонента.

3. Напишите кастомный хук `useInput`, который возвращает значение поля и обработчик его изменения. Используйте его в форме входа. Объясните, зачем нужны кастомные хуки и как они способствуют переиспользованию логики.

4. Реализуйте компонент `ProductList`, который рендерит список товаров. Оптимизируйте его с помощью `React.memo` и `useCallback`. Объясните, как эти инструменты помогают избежать лишних перерисовок.

5. Настройте проект на `React + TypeScript` с помощью `Vite`. Добавьте `ESLint` и `Prettier`. Опишите, как каждый из этих инструментов улучшает качество кода и поддерживаемость проекта.

6. Используя `React DevTools`, проанализируйте рендеринг компонентов в приложении. Объясните, как работает `Profiler` и как с его помощью выявлять узкие места производительности.

Контрольная работа 2.

1. Реализуйте управление состоянием корзины товаров с помощью `Context API` и `useReducer`. Объясните, когда следует использовать этот подход, а когда — внешние библиотеки (например, `Redux`).

2. Подключите приложение к API (например, <https://fakestoreapi.com/products>). Загрузите список товаров, обработайте состояния "загрузка", "ошибка", "данные получены". Объясните, как управлять серверным состоянием вручную без использования библиотек вроде `React Query`.

3. Настройте маршрутизацию с помощью `React Router`. Создайте страницы: Главная, Каталог, Товар, Корзина. Реализуйте защищённый маршрут Личный кабинет, доступный только авторизованным пользователям. Объясните, как организуется навигация и защита маршрутов в SPA.

4. Реализуйте форму авторизации: ввод логина и пароля, проверка на стороне клиента, симуляция входа (например, через `localStorage`). Добавьте защиту от XSS: экранируйте вывод и не используйте `dangerouslySetInnerHTML`. Объясните основные угрозы безопасности на фронтенде и способы их предотвращения.

5. Опишите архитектурный подход (например, `Feature-Sliced Design` или `Atomic Design`). Примените его к структуре проекта: разделите код на `entities`, `features`, `widgets`, `shared`. Объясните, как такая организация улучшает масштабируемость и сопровождаемость кода.

6. Сравните два подхода к управлению состоянием: локальное (`useState`) и глобальное (`Context`, `Redux`). Укажите плюсы и минусы каждого. В каких случаях предпочтительнее использовать каждый из них?

Примерное описание задания и критерии оценивания к проекту

Описание проектного задания:

Разработайте клиентское приложение на `React` с использованием `Next.js`, `TypeScript` и промышленных практик. Тематика — на выбор (например: интернет-магазин, блог, todo-приложение).

Основные требования к проекту:

- Серверный рендеринг (SSR/SSG) в `Next.js`.
- Архитектуру по признакам (например, `Feature-Sliced Design`).
- Маршрутизацию с динамическими и защищёнными страницами.
- Работу с API: загрузку данных, обработку состояний "загрузка", "ошибка".
- Управление состоянием (`Redux`, `Zustand` или `React Query`).
- Формы с валидацией и кастомными хуками (например, `useForm`).
- Авторизацию и защиту маршрутов.
- Модульные и E2E-тесты (`Jest + Testing Library`, `Cypress`).
- Доступность: семантика, `alt`, клавиатурная навигация.

- Настройку ESLint, Prettier, TypeScript.
- Деплой на Vercel/Netlify и, при желании, мониторинг (Sentry).

Проект сдаётся в виде репозитория на GitHub/GitLab. В README должен быть указан:

- краткое описание,
- инструкцию по запуску,
- ссылку на задеплойенное приложение,
- архитектурные решения,
- примеры тестов.

Критерии оценивания проекта:

- Архитектура: чёткая структура по признакам (features, entities), разделение ответственностей.
- UI/UX: понятный интерфейс, поддержка разных состояний, адаптивность.
- Состояние: корректное управление локальным, глобальным и серверным состоянием.
- API: работа с данными, обработка ошибок, кэширование (при использовании React Query).
- Роутинг: динамические и защищённые маршруты, навигация.
- Безопасность: защита от XSS, валидация форм, безопасное хранение токенов.
- Тестирование: модульные и E2E-тесты покрывают ключевые сценарии.
- Доступность: семантическая вёрстка, alt, aria, поддержка клавиатуры.
- SSR и оптимизация: используется SSR/SSG, высокие метрики Lighthouse.
- Качество кода: TypeScript, ESLint/Prettier, отсутствие дублирования, кастомные хуки.
- Деплой: приложение задеплойено и работает стабильно.
- Документация: README содержит всё необходимое для запуска и понимания проекта.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Назовите вид теста, при котором проверяется поведение отдельного компонента в изоляции.	Unit test / Модульный тест	УК-1
2.	Укажите подход, при котором данные из нескольких API объединяются в одном компоненте.	Синтез данных / Data aggregation	УК-1
3.	Выберите метод анализа производительности, доступный в React DevTools.	Профилирование / Profiling	УК-1
4.	Определите принцип, при котором система рассматривается как совокупность взаимодействующих элементов.	Системный подход	УК-1
5.	Укажите способ загрузки кода, при котором части приложения подгружаются по требованию.	Code splitting / Динамический импорт	ОПК-2

6.	Назовите архитектурный принцип, при котором логика и представление разделены.	Разделение ответственностей / Separation of concerns	ОПК-2
7.	Выберите структуру организации кода, при которой папки группируются по функциональности.	Feature-sliced design / FSD	ОПК-2
8.	Определите подход к управлению состоянием, подходящий для сложных приложений с множеством взаимосвязанных данных.	Redux / Zustand / MobX	ОПК-2
9.	Укажите атрибут, необходимый для доступности изображений при использовании скринридеров.	alt	ПК-2
10.	Назовите международный стандарт, регулирующий доступность веб-контента.	WCAG	ПК-2
11.	Выберите тег, используемый для создания семантической навигации по странице.	<code><nav></code>	ПК-2
12.	Определите принцип, обеспечивающий работу интерфейса без использования мыши.	Клавиатурная навигация / Keyboard navigation	ПК-2
13.	Укажите хук, с помощью которого в React создаётся изменяемое состояние.	useState	ПК-3
14.	Назовите хук, используемый для выполнения эффектов, таких как запросы к API.	useEffect	ПК-3
15.	Выберите инструмент, отвечающий за сборку и оптимизацию фронтенд-приложения.	Webpack / Vite / Rollup	ПК-3
16.	Определите библиотеку, упрощающую работу с серверным состоянием и кэшированием.	React Query / SWR	ПК-3
17.	Укажите метод в Next.js, используемый для генерации страниц на сервере при каждом запросе.	getServerSideProps	ПК-8
18.	Назовите тип рендеринга, при котором HTML формируется на стороне сервера.	SSR / Server-Side Rendering	ПК-8
19.	Выберите подход, при котором страницы генерируются на этапе сборки.	SSG / Static Site Generation	ПК-8
20.	Определите систему, используемую для отслеживания ошибок в боевом окружении.	Sentry / Error tracking / Datadog	ПК-8