
УТВЕРЖДЕНА

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

**Рабочая программа дисциплины (модуля)
«Технологии фронтенд-разработки»**

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)	8
5. Учебно-методическое обеспечение	9
6. Материально-техническое обеспечение	9
7. Методические и оценочные материалы	11

1. Краткая характеристика дисциплины (модуля)

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

Дисциплина «Технологии фронтенд-разработки» обеспечивает подготовку специалистов, способных создавать современные, быстрые и удобные пользовательские интерфейсы.

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) является выборной и доступна для изучения на 3 или 4 курсе в 6, 7 или 8 семестре на выбор после успешного освоения дисциплины «Основы промышленной разработки».

Цель изучения дисциплины (модуля): сформировать у студентов системные знания и практические навыки создания интерактивных web-интерфейсов, отвечающих требованиям качества, производительности, безопасности и доступности, а также подготовить их к самостоятельной работе в современных командах разработки.

Задачи изучения дисциплины (модуля):

- овладеть базовыми технологиями разметки и стилизации;
- освоить язык программирования JavaScript и его типизацию в TypeScript;
- изучить и применять архитектурные паттерны клиентского приложения;
- получить практические навыки работы с современными фронтенд-фреймворками.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- принципы работы веб-платформы: HTTP/HTTPS, DNS, клиент-серверное взаимодействие, cookies, кэширование, CORS.
- назначение и возможности инструментов разработчика браузера, включая анализ HTML, CSS, сетевых запросов и ошибок JavaScript.
- основы семантической HTML-разметки, форм, доступности, SEO и структуры веб-страницы.
- основные принципы CSS: каскад, специфичность, box model, адаптивная верстка, Flexbox, Grid, анимации и CSS-переменные.
- базовые и продвинутые особенности JavaScript: типы данных, функции, области видимости, замыкания, прототипы, объекты, классы и контекст this.
- принципы работы DOM API, событийной модели браузера и рендеринга страницы.
- механизмы асинхронного программирования в JavaScript: event loop, callbacks, Promise, async/await, Fetch API.
- современные возможности JavaScript: модули, итераторы, генераторы, Map, Set, Symbol, Proxy, Reflect.
- основы инструментальной среды фронтенд-разработки: Node.js, npm, Vite, сборка проекта, линтеры и форматтеры.
- основы TypeScript: статическая типизация, компиляция, tsconfig, интерфейсы, типы, объединения, сужение типов и структурная типизация.
- продвинутые возможности TypeScript: дженерики, utility types, mapped types,

conditional types и типобезопасная работа с данными.

- базовые архитектурные подходы и паттерны организации фронтенд-кода без использования фреймворков.

- принципы построения SPA на ванильном TypeScript, клиентского роутинга, загрузки модулей и оптимизации производительности.

- основные метрики производительности веб-приложений и подходы к их анализу с использованием Lighthouse и Web Vitals.

уметь:

- анализировать работу веб-страниц и сетевых запросов с помощью DevTools.

- создавать семантическую, доступную и валидную HTML-разметку.

- верстать адаптивные интерфейсы с использованием современных возможностей CSS.

- работать с макетом в Figma и переносить базовые визуальные параметры в код.

- писать JavaScript-код с учётом особенностей языка, областей видимости, замыканий, прототипов и контекста выполнения.

- работать с DOM: создавать, изменять и удалять элементы, обрабатывать события, управлять формами и интерактивностью страницы.

- реализовывать асинхронную логику с использованием Promise, async/await, Fetch API и AbortController.

- использовать современные структуры и возможности JavaScript для решения прикладных задач.

- создавать и настраивать фронтенд-проект с использованием Node.js, npm, Vite, ESLint и Prettier.

- переводить JavaScript-код на TypeScript и типизировать данные, функции, DOM-операции и ответы внешних API.

- использовать дженерики и utility types для построения типобезопасных абстракций.

- организовывать код фронтенд-приложения по модулям и применять базовые архитектурные паттерны.

- проектировать и реализовывать SPA на ванильном TypeScript.

- работать с внешними API, обрабатывать JSON-данные и ошибки сетевого взаимодействия.

- анализировать производительность веб-приложения и применять базовые методы оптимизации загрузки и рендеринга.

владеть:

- инструментами анализа и отладки веб-приложений в браузере.

- навыками разработки адаптивных, доступных и интерактивных веб-интерфейсов.

- навыками уверенного программирования на JavaScript в браузерной среде.

- методами работы с DOM, событиями, формами, браузерными API и асинхронными запросами.

- подходами к написанию поддерживаемого JavaScript/TypeScript-кода.

- инструментами современной фронтенд-разработки: Node.js, npm, Vite, ESLint, Prettier.

- навыками типизации фронтенд-приложений с использованием TypeScript.

- методами построения типобезопасных функций, моделей данных и обёрток для работы с API.

- подходами к модульной организации кода и применению архитектурных паттернов во фронтенд-разработке.

- навыками построения одностраничного приложения на ванильном TypeScript без использования фреймворков.

- методами анализа и оптимизации производительности веб-приложений.

— навыками чтения, анализа, рефакторинга и объяснения фронтенд-кода.

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 аналитики, 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,	ПК-3.1.	Знает требования и стандарты оформления научно-технических

	презентации, научные публикации по результатам выполненных исследований		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	Самосто ятельна я работа	
		Лекции	Семинары (практическ ие занятия)			
1	Веб-платформа		20	2	40	Домашнее задание, контрольная работа
2	JavaScript в браузере		20	2	40	Домашнее задание, контрольная работа
3	TypeScript		12	4	48	Домашнее задание, контрольная работа
	Экзамен			2		
Итого:			52	10	128	
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	Веб-платформа	Как работает веб HTML: семантика, формы, доступность Таблицы стилей CSS
2	JavaScript в браузере	JavaScript: основы языка JavaScript: объекты, прототипы, ООП DOM API и работа с браузером JavaScript: асинхронность JavaScript: продвинутые возможности Контрольная работа
3	TypeScript	Инструментальная среда фронтенд-разработки TypeScript: основы типизации и компиляция TypeScript: дженерики и утилитарные типы Архитектура и паттерны Контрольная работа SPA на ванильном TypeScript, производительность и оптимизация

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

Robotic 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) «Технологии фронтенд-разработки» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, кеймы, коллоквиум, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

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 и продемонстрировать навыки их практического применения.

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) «Технологии фронтенд-разработки»

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	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	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	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	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 решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	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	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Технологии фронтенд-разработки» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	40%	13	Набор задач по темам недели
Контрольная работа	30%	2	Примерные задания для контрольной работы
Экзамен	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Технологии фронтенд-разработки»: $0,4 \times \text{среднее за домашние задания} + 0,3 \times \text{среднее за контрольные работы} + 0,3 \times \text{экзамен}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

ДЗ 1. Семантическая разметка и адаптивный дизайн

1. **Задача** – создать простую «визит-карточку» компании.

2. Требования

— HTML-разметка должна быть полностью семантической (header, nav, main, article, footer).

— Стилизация только через CSS-файлы (без инлайн-стилей).

— Реализовать адаптивность: при ширине экрана $> 1024 \text{ px}$ – две колонки, $< 1024 \text{ px}$ – одна колонка.
-овать Flexbox или CSS-Grid, не таблицы.

3. **Критерии оценки** – корректность разметки, отсутствие лишних элементов, чистый CSS, отсутствие горизонтального скролла, валидность (W3C Validator).

ДЗ 2. JavaScript ES6 + простая интерактивность

1. **Задача** – написать небольшую игру «Угадай число».

2. Требования

- Код в отдельном `.js`-файле, подключённом к HTML-странице.
- Использовать **let/const**, стрелочные функции, шаблонные строки, деструктуризацию.
- Взаимодействие с пользователем через **prompt/alert** или динамические элементы DOM.
- Добавить возможность выбора диапазона чисел (по умолчанию 1-100).

3. **Критерии оценки** – применение современных возможностей ES6, читаемость кода, отсутствие глобальных переменных, корректность логики.

ДЗ 3. TypeScript + React (или Vue/Angular) – компонент со списком задач

1. **Задача** – реализовать «Todo-list» со следующими функциями:

- Добавление задачи, удаление, отметка выполненной.
- Сохранение списка в **localStorage** (чтение при загрузке).

2. Требования

- Проект сгенерировать через **create-react-app** (или аналогичный CLI-инструмент).
- Написать все файлы в TypeScript, объявив типы для состояния и пропсов.
- Использовать функциональные компоненты + хуки (**useState, useEffect**).
- Оформить стили через CSS-модули или styled-components.

3. **Критерии оценки** – корректный типизированный код, работа с **localStorage**, чистая архитектура (разделение UI и логики), грамотная стилизация.

ДЗ 4. Взаимодействие с API и обработка ошибок

1. **Задача** – создать страницу, показывающую текущую погоду для выбранного города.

2. Требования

- Использовать открытый API (например, OpenWeatherMap).
- Реализовать запрос через **fetch** (или **axios**) с **async/await**.
- Обрабатывать ошибки сети и неверный ввод (показать сообщение пользователю).
- Добавить индикатор загрузки (spinner).

3. **Критерии оценки** – корректный запрос, надёжная обработка исключений, пользовательский UI-фидбек.

ДЗ 5. Тестирование и CI

1. **Задача** – покрыть тестами любой из предыдущих проектов (Todo-list или погодный сервис).

2. Требования

- Юнит-тесты с **Jest** (или **Mocha/Chai**).
- Один e2e-тест с **Cypress** или **Playwright** (проверка полной пользовательской истории).
- Добавить в репозиторий GitHub Actions workflow, который при каждом push запускает тесты.

3. **Критерии оценки** – наличие тестов, покрытие > 70 % (по **jest --coverage**)

Примерные задания для контрольной работы

1. Что такое **семантическая разметка** и зачем она нужна? Приведите три примера семантических тегов.

2. Объясните разницу между **Flexbox** и **CSS-Grid**. Когда предпочитаем один подход другому?

3. Что такое **box-model** и как влияет **box-sizing: border-box** на расчёт размеров элемента?

4. Опишите основные особенности **ES6-модулей** (import/export). Чем они лучше старых скриптов-тегов?
5. Перечислите три способа **управления состоянием** в React и дайте короткую характеристику каждому.
6. Что такое **CORS** и как браузер реагирует на запросы, нарушающие политику кросс-доменных запросов?
7. В чем состоит цель **Service Worker** и какие примеры задач решаются с его помощью?
8. Какие типы тестов существуют в фронтенд-разработке? Дайте определение unit-test, integration-test, e2e-test.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Укажите основную архитектурную модель современных веб-приложений.	Трёхуровневая клиент-серверная модель (presentation-logic-data) / одностраничное приложение (SPA) с бек-эндом-API (micro-services).	УК-1
2	Назовите типичный способ взаимодействия клиентской части с сервером.	HTTP/HTTPS – REST-API, GraphQL; для двусторонней связи – WebSocket, Server-Sent Events, gRPC-Web.	УК-1
3	Укажите модель построения пользовательского интерфейса в современных фреймворках.	Компонентный (component-based) подход, виртуальный DOM (React, Vue, preact).	УК-1
4	Назовите основной способ организации стилей в масштабируемом проекте.	CSS-Modules, BEM, CSS-in-JS (styled-components, emotion).	ОПК-2
5	Укажите алгоритм/паттерн управления глобальным состоянием в SPA.	Redux (Flux), MobX, Context + useReducer, Pinia (Vue).	ОПК-2
6	Назовите свойство, гарантирующее сохранность данных в браузере между сессиями.	Персистентное хранилище – LocalStorage, SessionStorage, IndexedDB; кэш Service Worker.	ОПК-2
7	Укажите способ обнаружения и обработки ошибок запросов к API.	Централизованный обработчик (interceptor) + retry с экспоненциальным бэк-офф; отображение сообщения пользователю.	ОПК-2
8	Перечислите один из методов обеспечения согласованности UI при конкурентных запросах.	Optimistic UI-updates с откатом (rollback) либо отмена запросов через AbortController / Cancellation Token.	ОПК-2
9	Укажите математическую модель, используемую для оценки производительности клиентского кода.	Оценка сложности – Big O для рендеринга и diff; метрики Critical Rendering Path, Time-to-Interactive.	ПК-2
10	Назовите алгоритм оптимизации загрузки ресурсов в браузере.	Lazy-loading, code-splitting (dynamic import), tree-shaking, pre-fetch / pre-load.	ПК-2
11	Укажите принцип организации ввода-вывода в браузерном окружении.	Event-loop + асинхронные callbacks/Promises; Web Workers – внешний поток выполнения.	ПК-2

12	Перечислите методы защиты веб-приложения от XSS/CSRF.	Content-Security-Policy, HttpOnly + SameSite-cookies, санитизация ввода, токен-по-запросу (CSRF-token).	ПК-2
13	Укажите стандарт описания архитектуры фронтенд-приложения в технической документации.	IEEE 1471 / ISO/IEC/IEEE 42010 (Architecture Description) + ADR (Architectural Decision Records).	ПК-3
14	Укажите способ структурирования отчёта о результатах тестирования UI.	По типу тестов: unit-tests, integration-tests, e2e-tests; таблицы/графики покрытости, pass/fail, дефекты.	ПК-3
15	Укажите инструмент для визуального анализа производительности веб-страницы.	Chrome DevTools → Performance, Lighthouse, WebPageTest, GTmetrix.	ПК-3
16	Укажите формат представления результатов нагрузочного тестирования фронтенда.	HTML-/PDF-отчёт + JSON/CSV-файлы с метриками (latency, RPS, error-rate) и графиками.	ПК-3
17	Укажите принцип проектирования масштабируемого фронтенда.	Горизонтальное масштабирование через code-splitting, микрофронтенды, распространение статических файлов через CDN.	ПК-8
18	Назовите подход к управлению версиями UI-компонентов.	Semantic Versioning (SemVer) в npm-пакетах, changelog, автоматическое публикация (semantic-release).	ПК-8
19	Укажите этап жизненного цикла разработки, на котором внедряется CI/CD.	После написания кода и тестов – непрерывная интеграция (build + test) и непрерывный деплой (stage / prod).	ПК-8
20	Укажите инструмент, позволяющий выполнять статический анализ кода фронтенд-проекта.	ESLint + Stylelint, SonarQube, Prettier (для авто-форматирования).	ПК-8