
УТВЕРЖДЕНА

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

**Рабочая программа дисциплины (модуля)
«Фронтенд-разработка»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Машинное обучение в индустрии

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	3
3. Тематический план.....	9
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) «Фронтенд-разработка» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Машинное обучение в промышленности, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Фронтенд-разработка» позволяет сформировать системные знания и практические навыки современной клиентской разработки на стороне браузера. Дисциплина (модуль) охватывает полный цикл создания веб-интерфейсов: от основ веб-платформы и семантической вёрстки до написания типобезопасного TypeScript-кода, построения одностраничных приложений (SPA) и анализа производительности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Машинное обучение в промышленности и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений как дисциплина по выбору. Дисциплина (модуль) доступна к изучению на 2 курсе в 4 семестре.

Доступна для прохождения при условии успешного завершения дисциплины (модуля) «Промышленная разработка».

Цель изучения дисциплины (модуля): подготовить обучающихся к самостоятельной разработке, отладке и оптимизации современных веб-интерфейсов на ванильном JavaScript и TypeScript без использования фреймворков, с соблюдением принципов доступности, производительности и архитектурной чистоты.

Задачи изучения дисциплины (модуля):

- освоить принципы работы веб-платформы: HTTP, DNS, клиент-серверное взаимодействие, CORS, кэширование;
- научиться использовать инструменты разработчика браузера для анализа HTML, CSS, JavaScript и сетевых запросов;
- освоить семантическую вёрстку, адаптивный дизайн и основы доступности (a11y) и SEO;
- изучить современные возможности JavaScript и TypeScript, включая асинхронность, модули, дженерики и типобезопасность;
- развить навыки построения одностраничных приложений (SPA) на чистом TypeScript с клиентским роутингом, загрузкой данных и оптимизацией производительности.

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 вёрстка, 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-6.	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1.	Знает основные методы самооценки и анализа своей деятельности, а также принципы управления временем и целеполагания
		УК-6.2.	Умеет ставить реалистичные и достижимые цели, определять приоритеты в своей деятельности, а также разрабатывать и внедрять планы по совершенствованию своих навыков и компетенций на основе полученной самооценки
		УК-6.3.	Имеет практический опыт применения методов самооценки в своей профессиональной деятельности, включая участие в тренингах, семинарах и проектах, направленных на развитие личной эффективности и профессионального роста
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает основные методы и подходы к решению задач прикладной и компьютерной математики, включая алгоритмы, математическое моделирование и теорию оптимизации, а также современные инструменты и технологии, используемые в этой области
		ОПК-1.2.	Умеет анализировать и формулировать математические задачи, применять соответствующие методы и алгоритмы для их решения, а также интерпретировать и представлять результаты в

			понятной и доступной форме
		ОПК-1.3.	Имеет практический опыт работы над проектами или исследованиями в области прикладной и компьютерной математики, включая участие в конкурсах, олимпиадах или научных публикациях, где были решены актуальные и значимые задачи
ПК-1.	Способен определять общие формы и закономерности области машинного обучения в индустрии	ПК-1.1.	Знает основные теоретические концепции и принципы, относящиеся к области машинного обучения в индустрии, а также ключевые закономерности и модели, которые помогают в анализе и интерпретации данных
		ПК-1.2.	Умеет проводить систематический анализ области разработки, выявлять и формулировать общие закономерности и тенденции, а также применять методы исследования для получения новых знаний и понимания
		ПК-1.3.	Имеет практический опыт работы в области машинного обучения в индустрии, включая участие в научных проектах, исследованиях или практических заданиях, где были выявлены и описаны общие формы и закономерности
ПК-3.	Способен решать задачи профессиональной деятельности, формулировать результат, увидеть следствия полученного результата	ПК-3.1.	Знает основные принципы и методы решения задач профессиональной деятельности, а также способы формулирования и представления результатов, включая анализ последствий и их значимость в контексте проекта
		ПК-3.2.	Умеет применять математические и компьютерные методы для решения конкретных задач,

			формулировать четкие и обоснованные результаты, а также анализировать их последствия для дальнейших действий и решений
		ПК-3.3.	Имеет практический опыт в решении профессиональных задач, включая участие в проектах, где были получены результаты и проанализированы их следствия, что способствовало принятию обоснованных решений
ПК-6.	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере машинного обучения в индустрии	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в машинном обучении в индустрии
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

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 (практичес кие занятия)			
1	Веб-платформа		16		40	Домашние задания
2	JavaScript в браузере		18	4	44	Домашние задания Контрольная работа
3	TypeScript		18	4	44	Домашние задания Контрольная работа
	Зачет с оценкой			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
3	TypeScript	Инструментальная среда фронтенд-разработки TypeScript: основы типизации и компиляция TypeScript: дженерики и утилитарные типы Архитектура и паттерны 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) «Фронтенд-разработка» в рамках текущего контроля успеваемости используются такие виды учебной работы как семинары, домашние задания, контрольные работы, 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) «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 *зачета с оценкой*, 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 syllabus 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 (модуль). 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 решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
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
2	Не сдан	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
1	Не сдан	преподаватель задает дополнительные наводящие вопросы.

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

Активность	Вес	Количество	Описание
Домашние задания	40%	13	Набор задач по темам недели
Контрольная работа	30%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	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) «Фронтенд-разработка»: $0,4 \times \text{среднее за домашние задания} + 0,3 \times \text{среднее за контрольные работы} + 0,3 \times \text{зачет с оценкой}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1. Веб-платформа и вёрстка

1. Создайте семантическую разметку страницы новостного сайта: header, nav, main, article, aside, footer.
2. Добавьте форму входа с полями: email, пароль, чекбокс «Запомнить меня». Обеспечьте валидацию (required, type).
3. Свяжите метки с полями формы (<label for="...">).
4. Напишите CSS:
 - Адаптивная сетка (Grid/Flexbox),
 - Медиа-запросы для мобильных устройств,
 - CSS-переменные для цветовой схемы.
5. Проверьте доступность: добавьте alt, aria-label, проверьте контрастность.

\

Домашнее задание 2. JavaScript и DOM

1. Напишите скрипт, который:
 - Динамически создаёт список элементов из массива объектов (например, товаров),
 - Добавляет их в DOM,
 - При клике на элемент — удаляет его.
2. Реализуйте форму фильтрации: при вводе текста — фильтруются элементы списка.
3. Используйте делегирование событий.
4. Напишите функцию-обёртку для fetch, возвращающую Promise с обработкой ошибок.
5. Загрузите данные с JSON-эндпоинта (например, https://jsonplaceholder.typicode.com/posts?_limit=5) и отобразите.

Домашнее задание 3. TypeScript и SPA

1. Настройте проект с Vite + TypeScript:
 - Инициализируйте npm init,
 - Подключите TypeScript, ESLint, Prettier.

2. Создайте типы: User, Post, ApiResponse<T>.
3. Напишите типизированную функцию fetchData<T>(url), возвращающую Promise<T>.
4. Реализуйте клиентский роутинг:
 - /users — список пользователей,
 - /posts — список постов,
 - Используйте window.location.hash.
5. Добавьте ладер, обработку ошибок, оптимизируйте загрузку (ленивая подгрузка модулей — опционально).

Примерные задания для контрольных работ

Контрольная работа 1.

1. Напишите HTML-код для формы регистрации с полями: имя, email, пароль. Укажите атрибуты для валидации.
2. Напишите CSS-правило, которое выравнивает блок по центру страницы с использованием Flexbox.
3. Напишите JavaScript-код, который находит все ссылки на странице и добавляет им атрибут target="_blank".
4. Напишите функцию, которая принимает строку и возвращает её длину. Обработайте случай, если аргумент — null.
5. Напишите код, который отправляет GET-запрос к <https://jsonplaceholder.typicode.com/todos/1> и выводит заголовок в консоль.

Контрольная работа 2.

1. Напишите TypeScript-интерфейс Product с полями: id (число), name (строка), price (число), inStock (булево).
2. Напишите generic-функцию getLast<T>(arr: T[]): T | undefined, возвращающую последний элемент массива.
3. Укажите имя команды, с помощью которой компилируется TypeScript-файл в JavaScript.
4. Напишите код, который реагирует на изменение хеша в URL и выводит новое значение в консоль.
5. Напишите CSS-свойство, которое отключает перенос текста в блоке.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Как приоритизировать задачи в веб-разработке?	По бизнес-ценности, срокам и техническим рискам	УК-6
2.	Как планировать свое профессиональное развитие в веб-разработке?	Через изучение новых технологий, пет-проекты, курсы	УК-6
3.	Как оценить свою эффективность после спринта?	По выполненным задачам, качеству кода, обратной связи	УК-6
4.	Как управлять временем при оценке задач?	Декомпозиция, учет рисков, буфер времени	УК-6

5.	Какой протокол лежит в основе обмена данными между браузером и сервером?	HTTP	ОПК-1
6.	Как называется модель взаимодействия в вебе, при которой клиент инициирует запрос?	Модель запрос–ответ	ОПК-1
7.	Как называется объектная модель документа в браузере?	DOM	ОПК-1
8.	Как называется механизм наследования в JavaScript через цепочку прототипов?	Прототипное наследование	ОПК-1
9.	Как называется архитектурный подход, при котором всё приложение работает без перезагрузки страницы?	Одностраничное приложение (SPA)	ПК-1
10.	Какой тренд позволяет загружать части приложения по мере необходимости?	Ленивая загрузка (Lazy Loading)	ПК-1
11.	Что такое микрофронтенды в современной разработке?	Архитектура, разделяющая фронтенд на независимые модули	ПК-1
12.	Какая технология позволяет веб-приложению работать офлайн?	PWA (Progressive Web App)	ПК-1
13.	Какой атрибут формы связывает подпись с полем ввода для повышения доступности?	for	ПК-3
14.	Какой механизм CSS определяет приоритет применения стилей?	Каскадирование	ПК-3
15.	Какой метод используется для поиска элемента по CSS-селектору?	querySelector	ПК-3
16.	Как называется процесс преобразования TypeScript-кода в JavaScript?	Компиляция (транспилиция)	ПК-3
17.	Как называется механизм выполнения кода после завершения длительной операции без блокировки потока?	Асинхронность	ПК-6
18.	Какой синтаксис упрощает работу с промисами и делает код похожим на синхронный?	async/await	ПК-6
19.	Как называется шаблон проектирования, разделяющий данные, отображение и логику управления?	MVC	ПК-6
20.	Какой инструмент добавляет статическую типизацию в JavaScript?	TypeScript	ПК-6