
УТВЕРЖДЕНА

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

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

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Прикладной искусственный интеллект

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	12
4. Содержание дисциплины (модуля)	13
5. Учебно-методическое обеспечение	14
6. Материально-техническое обеспечение	14
7. Методические и оценочные материалы	16

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 02.03.01 «Математика и компьютерные науки», профиль «Прикладной искусственный интеллект», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Технологии фронтенд-разработки» формирует прочную фундаментальную базу для успешной работы в современной веб-разработке, позволяя глубоко понимать принципы работы браузера, языков и инструментов. Полученные знания и навыки обеспечивают возможность эффективного освоения любых фронтенд-фреймворков и создания масштабируемых, производительных и поддерживаемых веб-приложений.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Прикладной искусственный интеллект» и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) доступна для изучения на 3 или 4 курсе в 5, 6, 7 или 8 семестре на выбор после успешного освоения дисциплины (модуля): «Основы промышленной разработки».

Цель изучения дисциплины (модуля): заключается в формировании системного понимания веб-платформы и навыков создания современных, производительных, доступных и поддерживаемых веб-интерфейсов с использованием HTML, CSS, JavaScript и TypeScript без зависимости от фреймворков.

Задачи изучения дисциплины (модуля):

- изучить принципы работы веб-платформы, включая HTTP/HTTPS, DNS, клиент-серверное взаимодействие, механизмы безопасности (CORS, cookies) и процессы кэширования;
- освоить современные подходы к верстке с использованием семантического HTML, адаптивного дизайна, Flexbox, Grid и практик доступности (a11y) и SEO;
- развить навыки программирования на JavaScript и TypeScript, включая работу с DOM, асинхронным кодом, модулями и современными языковыми возможностями;
- научиться проектировать и реализовывать одностраничные приложения (SPA) на ванильном TypeScript с клиентским роутингом и типобезопасной работой с API;
- освоить инструментальную экосистему фронтенд-разработки, включая Node.js, npm, Vite, линтеры, форматтеры и инструменты анализа производительности (Lighthouse, Web Vitals).

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:

Код и содержание компетенции	Индикатор компетенции и 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.
УК-3. Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде (соответствует УНК-02)	УК-3.1. (УНК-02.01.) Принимает участие в групповом взаимодействии в ходе профессиональной деятельности (базовый) УНК-02.01.01. У-1. Проявляет понимание своей роли в команде УНК-02.01.01. У-2. Умеет ясно выражать свои мысли в группе УНК-02.01.01. У-3. Способен слушать и учитывать мнения других участников команды. УНК-02.01.01. У-4. Умеет давать конструктивную обратную связь УНК-02.01.01. У-5. Умеет работать в команде, проявлять инициативу и поддерживать коллег УНК-02.01.01. У-6. Обладает навыками разрешения конфликтных ситуаций и поиска компромиссов УНК-02.01.01. З-1. Знает основы командной работы, роли и ответственности каждого участника.
	УК-3.2. (УНК-02.02.) Проявляет лидерство и осуществляет наставничество (базовый) УНК-02.02.01. У-1. Умеет мотивировать и вдохновлять команду для достижения общих целей УНК-02.02.01. У-2. Способен организовывать работу группы, распределять обязанности и контролировать выполнение задач УНК-02.02.01. У-3. Умеет делиться знаниями и опытом, оказывать поддержку и осуществлять наставничество коллегам УНК-02.02.01. У-4. Способен принимать ответственные решения и демонстрировать пример профессионализма УНК-02.02.01. З-1. Знает основы лидерства, мотивации и командообразования УНК-02.02.01. З-2. Знает принципы эффективного наставничества и развития персонала УНК-02.02.01. З-3. Знает методы оценки эффективности работы команды и индивидуальных участников
ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы,	ОПК-6.1. Знает принципы алгоритмизации, основные структуры данных, парадигмы программирования, методы разработки, тестирования и отладки компьютерных программ.

пригодные для практического применения	ОПК-6.2. Умеет разрабатывать, документировать, тестировать и отлаживать алгоритмы и компьютерные программы, пригодные для практического применения, с использованием современных языков и сред разработки.
	ОПК-6.3. Имеет практический опыт разработки и сопровождения компьютерных программ и программных комплексов для решения прикладных задач.
ППК-ДА3. Способен проводить аналитические исследования с применением технологий больших данных	ППК-ДА3.2. – Обеспечивает соответствие результатов анализа бизнес-задачам заказчика (средний) ППК-ДА3.2. 3-1. Знает методы перевода бизнес-требований в аналитические задачи ППК-ДА3.2. 3-2. Знает ключевые бизнес-метрики в предметной области
ППК-Р5. Способен разрабатывать требования и проектировать программное обеспечение	ППК-Р5.1. Анализирует возможности реализации требований к компьютерному программному обеспечению (средний) ППК-Р5.1. 3-1. Знает возможности существующей программно-технической архитектуры. ППК-Р5.1. 3-2. Знает возможности современных и перспективных средств разработки программных продуктов, технических средств. ППК-Р5.1. 3-3. Знает методологии разработки компьютерного программного обеспечения и технологии программирования. ППК-Р5.1. У-1. Умеет проводить сбор и систематизацию требований к компьютерному программному обеспечению. ППК-Р5.1. У-2. Умеет выявлять взаимосвязи и документировать требования к компьютерному программному обеспечению ППК-Р5.1. У-3. Умеет вырабатывать варианты реализации требований к компьютерному программному обеспечению.
	ППК-Р5.2. Разрабатывает технические спецификации на программные компоненты и их взаимодействие (средний) ППК-Р5.2. 3-1. Знает методы и средства проектирования компьютерного программного обеспечения. ППК-Р5.2. 3-2. Знает методы и средства проектирования программных интерфейсов. ППК-Р5.2. У-1. Умеет выбирать средства реализации требований к компьютерному программному обеспечению. ППК-Р5.2. У-2. Умеет вырабатывать варианты реализации компьютерного программного обеспечения. ППК-Р5.2. У-3. Умеет проводить оценку и обоснование рекомендуемых решений.
	ППК-Р5.3. Проектирует компьютерное программное обеспечение (продвинутый) ППК-Р5.3. 3-1. Знает принципы построения и виды архитектуры компьютерного программного обеспечения. ППК-Р5.3. 3-2. Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения. ППК-Р5.3. 3-3. Знает нормативно-технические документы (стандарты), определяющие требования к технической документации на компьютерное программное обеспечение. ППК-Р5.3. 3-4. Знает методы и средства проектирования

	компьютерного программного обеспечения. ППК-Р5.3. У-1. Умеет разрабатывать и изменять архитектуру компьютерного программного обеспечения и согласовывать ее с системным аналитиком и архитектором программного обеспечения. ППК-Р5.3. У-2. Умеет проектировать структуры данных. ППК-Р5.3. У-3. Умеет проектировать программные интерфейсы. ППК-Р5.4. Выполняет логическое проектирование системы (продвинутый) ППК-Р5.4. З-1. Знает устройство и функционирование ИТ-систем/продуктов ППК-Р5.4. З-2. Знает методы моделирования и описания устройства и функционирования ИТ-систем/продуктов, их частей, обеспечения и окружения ППК-Р5.4. З-3. Знает базовые технологии взаимодействия и интеграции систем и компонентов ППК-Р5.4. З-4. Знает методы функциональной декомпозиции ИТ-систем ППК-Р5.4. У-1. Умеет декомпонировать ИТ-системы и ИТ-продукты на подсистемы и элементы поставки ППК-Р5.4. У-2. Умеет описывать интерфейсы пользователя на логическом уровне ППК-Р5.4. У-3. Умеет описывать интеграции со смежными системами на логическом уровне
ППК-Р6. Способен участвовать в промышленной разработке программного обеспечения	ППК-Р6.1. Работает в соответствии с промышленными методологиями разработки (средний) ППК-Р6.1. З-1. Знает принципы Agile и их применение в промышленных проектах. ППК-Р6.1. З-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership). ППК-Р6.1. У-1. Умеет оценивать объем задачи и срок ее выполнения, участвовать в планировании спринтов. ППК-Р6.1. У-2. Умеет работать в команде с использованием инструментов управления проектами. ППК-Р6.2. Использует инструменты промышленной разработки (средний) ППК-Р6.2. З-1. Знает принципы Continuous Integration and Continuous Delivery (CI/CD). ППК-Р6.2. З-2. Знает системы мониторинга и логирования в продуктивной среде. ППК-Р6.2. У-1. Умеет настраивать потоки работ CI/CD. ППК-Р6.2. У-2. Умеет работать с контейнеризацией и оркестрацией. ППК-Р6.2. У-3. Умеет настраивать мониторинг в продуктивной среде. ППК-Р6.3. Разрабатывает масштабируемый и поддерживаемый код (продвинутый) ППК-Р6.3. З-1. Знает принципы чистого кода, SOLID, DRY, KISS и др. ППК-Р6.3. З-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения. ППК-Р6.3. З-3. Знает паттерны проектирования и антипаттерны. ППК-Р6.3. У-1. Умеет разрабатывать модульный и тестируемый программный код. ППК-Р6.3. У-2. Умеет выполнять модульное, интеграционное и

	нагрузочное тестирование. ППК-Р6.3. У-3. Умеет проводить рефакторинг для повышения качества кода. ППК-Р6.3. У-4. Умеет применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах.
	ППК-Р6.4. Участвует в развертывании и поддержке программного обеспечения (продвинутой) ППК-Р6.4. З-1. Знает стратегии развертывания промышленного программного обеспечения. ППК-Р6.4. З-2. Знает основы работы с облачными платформами. ППК-Р6.4. У-1. Умеет развертывать приложения в облаке. ППК-Р6.4. У-2. Умеет обнаруживать и устранять инциденты с работой продуктивного программного обеспечения.
ППК-Р7. Способен применять искусственный интеллект (ИИ) для генерации и отладки программного кода	ППК-Р7.1. Применяет ИИ-инструменты для генерации программного кода (базовый) ППК-Р7.1. З-1. Знает принципы работы современных генеративных ИИ-моделей для генерации кода. ППК-Р7.1. З-2. Знает ограничения и риски использования ИИ-генерации (безопасность, качество кода, лицензирование). ППК-Р7.1. У-1. Умеет формулировать корректные текстовые запросы (промты) для генерации кода. ППК-Р7.1. У-2. Умеет интегрировать ИИ-инструменты в среду разработки.
	ППК-Р7.2. Использует ИИ для анализа и отладки кода (средний) ППК-Р7.2. З-1. Знает методы ИИ-анализа кода. ППК-Р7.2. З-2. Знает форматы и инструменты для автоматизированного тестирования с ИИ. ППК-Р7.2. У-1. Умеет настраивать ИИ-инструменты для поиска уязвимостей. ППК-Р7.2. У-2. Умеет интерпретировать рекомендации ИИ по исправлению кода.
	ППК-Р7.3. Оптимизирует код с помощью ИИ (продвинутой) ППК-Р7.3. З-1. Знает методы ИИ-оптимизации. ППК-Р7.3. З-2. Знает критерии качества кода, применяемые ИИ-системами. ППК-Р7.3. У-1. Умеет использовать ИИ для рефакторинга. ППК-Р7.3. У-2. Умеет проверять корректность оптимизаций, предложенных ИИ.
	ППК-Р7.4. Оценивает этические и профессиональные аспекты применения ИИ в разработке (продвинутой) ППК-Р7.4. З-1. Знает этические нормы использования ИИ (конфиденциальность, плагиат кода и т.п.). ППК-Р7.4. З-2. Знает лицензионные ограничения сгенерированного кода. ППК-Р7.4. У-1. Умеет проверять код на соответствие стандартам после ИИ-генерации. ППК-Р7.4. У-2. Умеет документировать использование ИИ в разработке.

ППК-ДА4. Способен выявлять бизнес-проблемы или бизнес-возможности	ППК-ДА4.2. Выявление истинных бизнес-проблем или бизнес-возможностей (продвинутый) ППК-ДА4.2. 3-1. Знает предметную область и специфику деятельности организации в объеме, достаточном для решения задач бизнес-анализа ППК-ДА4.2. 3-2. Знает инструменты, техники анализа бизнес-ситуации и предметной области, включая анализ данных. ППК-ДА4.2. 3-3. Знает перспективные и существующие цифровые технологии и цифровые возможности для бизнеса в контексте предметной области и специфики деятельности организации.
РГ-1. Способен руководить процессами разработки компьютерного программного обеспечения	РГ-1.1. Руководит разработкой программного кода (средний) РГ-1.1. 3-1. Знает методы принятия управленческих решений. РГ-1.1. 3-2. Знает основные принципы и методы управления персоналом РГ-1.1. 3-1 Знает технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии РГ-1.1. У-1. Способен распределять задачи на разработку программного кода между исполнителями. РГ-1.1. У-2. Применяет лучшие мировые практики оформления программного кода. РГ-1.1. У-3. Оценивает качество и эффективность программного кода.
	РГ-1.2. Руководит проверкой работоспособности компьютерного программного обеспечения (средний) РГ-1.2. 3-1. Знает основные методы измерения и оценки характеристик компьютерного программного обеспечения РГ-1.2. 3-2. Знает методы и средства проверки работоспособности компьютерного программного обеспечения РГ-1.2. 3-3. Знает методы и средства рефакторинга и оптимизации программного кода РГ-1.2. У-1. Умеет распределять задачи на проверку работоспособности компьютерного программного обеспечения между исполнителями. РГ-1.2. У-2. Умеет оценивать результаты проверки работоспособности компьютерного программного обеспечения
	РГ-1.3. Руководит интеграцией программных модулей и компонентов компьютерного программного обеспечения (продвинутый) РГ-1.3. 3-2. Знает методы и программные интерфейсы взаимодействия компьютерного программного обеспечения с внешними программными компонентами РГ-1.3. 3-3. Знает методы проектирования и разработки программных интерфейсов взаимодействия внутренних модулей компьютерного программного обеспечения РГ-1.3. У-1. Умеет назначать задания на разработку процедур интеграции, сборку, подключение к внешней среде, проверку работоспособности выпусков программного продукта
РГ-4. Способен организовать промышленную разработку программного обеспечения	РГ-4.1. Внедряет промышленные методологии разработки (средний) РГ-4.1. 3-1. Знает принципы Agile и их применение в промышленных проектах. РГ-4.1. 3-2. Знает процессы code review, принципы

	коллективного владения кодом (collective code ownership). РГ-4.1. У-1. Умеет оценивать объем задач и срок их выполнения. РГ-4.1. У-2. Умеет распределять задачи между участниками команды. РГ-4.1. У-3. Умеет организовать работу в команде с использованием инструментов управления проектами.
--	---

3. Тематический план

№п/ п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Практические занятия			
1	Веб-платформа		12		40	Домашние задания
2	JavaScript в браузере		20	4	44	Домашние задания, Контрольная работа
3	TypeScript		20	4	44	Домашние задания, 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
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
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) «Технологии фронтенд-разработки» в рамках текущего контроля успеваемости используются такие виды учебной работы, как семинары, домашние задания и 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 (модуля) 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	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 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) «Технологии фронтенд-разработки» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашнее задание	40%	13	Набор задач по темам недели
Контрольная работа	30%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
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) «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

Домашнее задание 1.

1. Объясните, как работает веб: из каких этапов состоит запрос к веб-сайту, начиная с ввода URL и заканчивая отображением страницы в браузере. Укажите роль DNS, HTTP, HTML, CSS.

2. Создайте семантическую HTML-разметку для страницы новостного сайта. Используйте теги `<header>`, `<nav>`, `<main>`, `<article>`, `<aside>`, `<footer>`. Объясните, зачем нужна семантика и как она влияет на доступность и SEO.

3. Добавьте на страницу форму обратной связи с полями: имя, email, сообщение. Укажите правильные атрибуты (type, required, placeholder) и объясните, как форма обеспечивает доступность (например, через `<label>` и `aria-*` атрибуты).

4. Напишите CSS-стили для созданной страницы: задайте цвета, шрифты, отступы. Используйте каскад и специфичность, чтобы продемонстрировать, как браузер выбирает стиль для элемента.

5. Сделайте страницу адаптивной: используйте медиазапросы, чтобы изменить компоновку на мобильных устройствах. Примените Flexbox или Grid для построения сетки. Объясните, как работает адаптивная верстка.

Домашнее задание 2.

1. Напишите JavaScript-код, который при загрузке страницы добавляет заголовок `<h1>` в `<main>` с текстом "Добро пожаловать!". Объясните, как работает DOM API и как найти/создать/вставить элементы.

2. Создайте объект user с полями name, age, isActive. Напишите функцию, которая принимает такой объект и возвращает строку вида "Пользователь: Имя, возраст: 25, активен: да/нет". Используйте прототипы, чтобы добавить метод getInfo() к объекту.

3. Найдите на странице кнопку и привяжите к ней обработчик события click, который выводит сообщение в консоль. Добавьте возможность отключить обработчик после трёх нажатий. Объясните, как работает событийная модель браузера.

4. Напишите асинхронную функцию, которая с помощью `fetch` получает данные с публичного API (например, <https://jsonplaceholder.typicode.com/users>). Выведите имена пользователей на страницу. Используйте `async/await` и обработайте возможные ошибки.

5. Создайте итератор для массива пользователей, который возвращает одного пользователя за раз. Реализуйте простой генератор, который фильтрует активных пользователей. Объясните разницу между итераторами и генераторами.

Домашнее задание 3.

1. Установите Node.js и инициализируйте проект с помощью `npm init`. Настройте Vite для работы с TypeScript. Добавьте ESLint и Prettier. Опишите, зачем нужны эти инструменты и как они улучшают разработку.

2. Перепишите код из второго домашнего задания на TypeScript: определите интерфейс `User` с типами полей, типизируйте функции и ответ API. Объясните, как статическая типизация помогает избежать ошибок.

3. Создайте дженерик-функцию `filterByProperty<T>(array: T[], key: keyof T, value: any)`, которая фильтрует массив по свойству. Используйте utility types, например `Partial<User>`, чтобы типизировать форму редактирования. Объясните преимущества дженериков и утилитарных типов.

4. Реализуйте простой одностраничный роутер на ванильном TypeScript: при изменении хэша в URL (`#home`, `#about`) отображается соответствующий контент. Используйте модули для разделения логики. Опишите принцип работы клиентского роутинга.

5. Проанализируйте производительность своей страницы с помощью Lighthouse. Оцените метрики Web Vitals (FCP, LCP, CLS). Предложите 3 способа оптимизации загрузки и рендеринга (например, ленивая загрузка, минификация, кэширование).

Примерные задания для контрольной работы

Контрольная работа 1.

1. Напишите краткое описание основных особенностей языка JavaScript: типы данных, области видимости, замыкания, прототипное наследование. Объясните, чем отличается `var`, `let`, `const`.

2. Создайте объект `Product` с полями `name`, `price`, `inStock`. Добавьте метод `getInfo()` через прототип. Реализуйте функцию-конструктор или класс для создания нескольких экземпляров. Объясните, как работает прототипная цепочка.

3. Напишите JavaScript-код, который при загрузке страницы находит элемент по ID и изменяет его содержимое. Добавьте обработчик события `click`, который скрывает или показывает другой элемент. Объясните, как работает DOM API и события.

4. Реализуйте функцию, которая с помощью `fetch` получает данные с API (например, список пользователей). Используйте `async/await`, обработайте ошибки. Выведите результат на страницу. Объясните, как работает `event loop` и асинхронность в JavaScript.

5. Создайте итератор для массива товаров, который возвращает по одному элементу за раз. Реализуйте генератор, который фильтрует товары по цене. Объясните разницу между итераторами и генераторами.

6. Напишите функцию, которая использует замыкание для создания счётчика. Объясните, как замыкания помогают инкапсулировать данные и почему они важны в JavaScript.

Контрольная работа 2.

1. Опишите основные преимущества использования TypeScript по сравнению с JavaScript. Объясните, как работает компиляция, `tsconfig.json` и статическая типизация.

2. Создайте интерфейс User с полями id, name, email, isActive. Напишите функцию createUser, принимающую объект типа User. Передайте в неё объект с лишним полем — объясните, как TypeScript обрабатывает это, и как включить строгую проверку.

3. Реализуйте дженерик-функцию filterBy<T>(array: T[], key: keyof T, value: any), которая фильтрует массив по свойству. Используйте Partial<T>, Pick<T> или Omit<T> для типизации формы редактирования. Объясните, зачем нужны утилитарные типы.

4. Настройте простой проект с помощью Vite и TypeScript. Добавьте ESLint и Prettier. Опишите, зачем нужны эти инструменты и как они улучшают качество кода и командную разработку.

5. Реализуйте клиентский роутинг на ванильном TypeScript: при изменении # в URL (#home, #profile) отображается соответствующий контент. Используйте модули для разделения логики. Объясните принцип работы SPA без фреймворков.

6. Опишите один из архитектурных паттернов (например, Module Pattern, Observer, Repository). Покажите пример его использования во фронтенд-приложении. Объясните, как паттерны улучшают структуру кода.

7. Проанализируйте производительность веб-приложения с помощью Lighthouse. Оцените метрики Web Vitals (LCP, FID, CLS). Предложите три способа оптимизации: ленивая загрузка, минификация, кэширование. Объясните, как каждое улучшение влияет на пользовательский опыт.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция	Индикатор
1	Опишите роли участников команды при разработке фронтенд-приложения и распределении задач по компонентам интерфейса.	Разработчик пишет код, тестировщик проверяет, дизайнер проектирует интерфейс, все несут ответственность за этап	УК-3, УК-1	УК-3.1, УК-1.2
2	Как мотивировать команду и распределить ресурсы при оптимизации производительности фронтенда с ограниченным временем?	Четкое распределение задач, контроль сроков, пример профессионализма, оценка трудоемкости этапов проекта	УК-3, РГ-1	УК-3.2, РГ-1.1
3	Подготовьте презентацию результатов оптимизации веб-приложения для стейкхолдеров и руководства компании.	Структура: цель, методы, результаты, выводы, рекомендации с учетом целевых установок заинтересованных сторон	УК-1, ППК-ДА3	УК-1.3, ППК-ДА3.2
4	Оформите документацию по фронтенд-проекту согласно профессиональным стандартам и требованиям к техническим спецификациям.	Титульный лист, содержание, введение, основная часть, выводы,	УК-1, ППК-Р5	УК-1.3, ППК-Р5.3

		список источников, приложения по ГОСТ		
5	Дайте определение семантической верстке и запишите её смысл в контексте доступности веб-приложений.	Семантическая верстка это использование HTML тегов по назначению для улучшения доступности и SEO	УК-1, ОПК-6	УК-1.2, ОПК-6.1
6	Проанализируйте экономическую информацию для выбора между фреймворками для клиентской части приложения.	Сравнение затрат, производительности, рисков, учет специфики задач, влияние на эффективность бизнес-решений	УК-1, ОПК-6	УК-1.2, ОПК-6.2
7	Обоснуйте экономическое решение о выборе архитектуры SPA на основе анализа затрат и выгод.	Решение выбирается если учитывает интересы всех сторон, повышает защиту инвестора от неблагоприятных изменений	УК-1, ОПК-6	УК-1.3, ОПК-6.3
8	Запишите формулу для расчета времени загрузки страницы и выберите метод анализа данных для оценки.	Формула времени загрузки, метод статистического анализа для оценки значимости улучшений производительности	ОПК-6, ППК-ДА3	ОПК-6.2, ППК-ДА3.2
9	Оптимизируйте выбор структуры компонентов и обоснуйте влияние на бизнес показатели компании.	Выбор структуры с максимальной эффективностью, оценка воздействия на ключевые показатели компании, скорость отрисовки	ОПК-6, ППК-Р5	ОПК-6.3, ППК-Р5.3
10	Обоснуйте экономическое решение в учебном проекте с учетом системного мышления при проектировании фронтенд-архитектуры.	Использование модели с учетом взаимосвязей между компонентами, состоянием и бизнес-целями компании	ОПК-6, РГ-4	ОПК-6.3, РГ-4.1
11	Назовите правовые нормы регулирующие использование лицензий при разработке фронтенд-ПО и их смысл.	Лицензии MIT, Apache, требования к открытому коду, ответственность за нарушение лицензионных соглашений	ППК-Р5, УК-1	ППК-Р5.3, УК-1.1

12	Примените правовые нормы при оценке рисков нарушения лицензий при работе с npm-пакетами.	Анализ рисков несоблюдения лицензионных требований, оценка вероятности и пороговых значений, меры минимизации штрафов	ППК-Р5, ППК-Р6	ППК-Р5.3, ППК-Р6.3
13	Дайте правовую оценку ситуации с нарушением требований по лицензированию во фронтенд-проекте.	Определение нарушений лицензионных актов, применение санкций, влияние на репутацию организации	ППК-Р5, УК-1	ППК-Р5.3, УК-1.3
14	Выберите метод статистического анализа для проверки гипотез о влиянии оптимизации на производительность веб-приложения.	А/В тест, проверка гипотез о значимости различий, расчет доверительных интервалов для оценок времени загрузки	ППК-ДА3, ППК-ДА4	ППК-ДА3.2, ППК-ДА4.2
15	Адаптируйте аналитическую модель оценки фронтенд-модуля под бизнес потребности и оцените влияние решений.	Перевод технических метрик в бизнес показатели, оценка воздействия на ключевые показатели компании, оптимизация процессов	ППК-ДА3, ППК-Р6	ППК-ДА3.2, ППК-Р6.3
16	Выберите тип визуализации для представления данных о результатах тестирования и коммуникации с руководством.	График времени загрузки и доверительных интервалов, интерактивный дашборд для публичной дискуссии о результатах анализа	ППК-ДА3, УК-1	ППК-ДА3.2, УК-1.3
17	Выявите бизнес проблему на основе анализа отклонений в производительности фронтенд-кода и примите решение.	Проблема низкой скорости отрисовки, выбор стратегии реагирования, решение о корректировке архитектуры или алгоритма	ППК-ДА4, ППК-Р7	ППК-ДА4.2, ППК-Р7.2
18	Классифицируйте бизнес возможности при масштабировании фронтенд-решения с учетом стратегического планирования.	Оптимизация вычислительных ресурсов, разработка плана действий с учетом	ППК-ДА4, РГ-4	ППК-ДА4.2, РГ-4.1

		целей компании, выявление резервов роста		
19	Сформулируйте рекомендации на основе статистических выводов о результатах тестирования для управления продуктом.	При значимом улучшении внедрение изменения, при отсутствии эффекта отказ от изменения для эффективности системы	ППК-Р6, ППК- Р7	ППК-Р6.3, ППК-Р7.3
20	Оцените риски реализации фронтенд-проекта и разработайте меры управления.	Анализ вероятности срывов сроков, мониторинг индикаторов успешности, пересмотр плана при выявлении проблем	РГ-1, УК-1	РГ-1.3, УК- 1.3