МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной директором ФГАОУ ВО РУТ
(МИИТ) Париновым Д.В.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Разработка клиентской части веб-приложений

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки программного обеспечения

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: директор центра Проневич Ольга Борисовна
Дата: 12.11.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина посвящена разработке клиентской части веб-приложений с использованием HTML, CSS, JavaScript, базовых средств TypeScript и начальных возможностей React. В ходе изучения рассматриваются устройство веб-страницы, семантическая разметка, формы, каскадные таблицы стилей, блочная модель, позиционирование, Flexbox, Grid, адаптивность, анимации, язык JavaScript, работа с DOM, события, валидация форм, поиск и изменение данных на странице. На лабораторных занятиях обучающиеся последовательно создают клиентское веб-приложение от статической страницы до интерактивного интерфейса с проверкой ввода, управлением состоянием элементов и начальной компонентной организацией.

Целью освоения дисциплины является формирование способности проектировать, реализовывать, тестировать и документировать клиентскую часть веб-приложений, обеспечивающую корректную структуру, адаптивное отображение, интерактивное поведение и подготовку к интеграции с серверными сервисами.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности – анализировать требования к клиентскому интерфейсу, создавать семантическую HTML-разметку и формы, проектировать оформление и адаптивную раскладку средствами CSS, реализовывать интерактивное поведение на JavaScript, работать с DOM, событиями, массивами, объектами и строками, применять базовую типизацию TypeScript, создавать начальные компоненты React, проводить проверку клиентских сценариев и готовить техническую документацию.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ОПК-6 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения;

ППК-Р1 - Способен разрабатывать и отлаживать программный код;

ППК-Р3 - Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- принципы клиент-серверной архитектуры веб-приложений и роль браузера в обработке HTML, CSS и JavaScript
- основы HTTP, адресации ресурсов, запросов, ответов, заголовков, кодов состояния и загрузки ресурсов веб-страницы
- назначение HTML, структура HTML-документа, элементы, атрибуты, ссылки, изображения, списки, таблицы и формы
- принципы семантической разметки, включая header, main, footer, aside, article, nav, section и элементы текстового выделения
- устройство HTML-форм, включая input, select, textarea, fieldset, legend, проверку обязательных полей и связь формы с пользовательским сценарием
- назначение CSS, способы подключения стилей, каскад, наследование, специфичность, селекторы, псевдоклассы и псевдоэлементы
- основы оформления текста, цвета, шрифтов, размеров, отступов, границ, скруглений и стилизации таблиц
- блочную модель CSS, свойства display, float, clear, position и правила формирования потока документа
- принципы построения гибких раскладок средствами Flexbox, включая направление, выравнивание, порядок, рост и сжатие элементов
- принципы построения двумерных раскладок средствами Grid, включая строки, столбцы, области, промежутки, доли пространства и объединение ячеек
- основы адаптивной верстки, медиавыражения, контрольные ширины экрана и изменение структуры интерфейса для разных устройств
- назначение переходов и ключевых кадров CSS при создании анимаций, движения, появления, масштабирования и визуальной обратной связи
- основы JavaScript, включая переменные, типы данных, операции, сравнения, ветвления, циклы, преобразование типов и работу со строками
- массивы и объекты JavaScript, методы изменения и обхода массивов, сортировку, вложенные массивы, JSON и представление структурированных данных
- функции JavaScript, включая объявления, стрелочные функции, параметры по умолчанию, остаточные параметры, область видимости и побочные эффекты
- объектную модель документа, выбор элементов, чтение и изменение содержимого, классов, атрибутов, значений форм и структуры страницы

- обработку событий браузера, включая события ввода, изменения, нажатия, загрузки страницы и применение addEventListener

- основы проверки пользовательского ввода, блокировки действий, поиска по таблице, фильтрации данных и изменения списка элементов

- начальные возможности TypeScript, включая аннотации типов, интерфейсы, типизацию функций, массивов, объектов и элементов DOM

- начальные принципы React, включая компонент, свойства, состояние, обработку событий, условное отображение и списки элементов

- методы проверки клиентской части веб-приложения, включая ручные сценарии, инструменты разработчика браузера, проверку адаптивности и тестирование пользовательского ввода

Уметь:

- уметь создавать структуру веб-страницы при помощи HTML в условиях представления текстового, графического и навигационного содержания

- уметь размечать содержимое страницы при помощи семантических элементов HTML в условиях доступности, читаемости и поисковой обработки документа

- уметь создавать формы ввода при помощи HTML-элементов форм в условиях регистрации, расчета, выбора параметров и отправки пользовательских данных

- уметь оформлять веб-страницы при помощи CSS в условиях управления цветом, типографикой, отступами, границами, состояниями наведения и таблицами

- уметь управлять расположением элементов при помощи блочной модели, display, float и position в условиях типовых задач компоновки страницы

- уметь разрабатывать гибкие раскладки при помощи Flexbox в условиях выравнивания, распределения и изменения порядка элементов

- уметь разрабатывать двумерные раскладки при помощи Grid в условиях создания областей страницы, колонок, строк и адаптивной сетки

- уметь создавать адаптивные страницы при помощи медиавыражений CSS в условиях отображения на смартфонах, планшетах и настольных экранах

- уметь применять переходы и анимации CSS при помощи transition и keyframes в условиях визуальной обратной связи и изменения состояния элементов

- уметь реализовывать вычислительные сценарии при помощи JavaScript в условиях работы с переменными, строками, числами, ветвлениями, циклами и преобразованием типов

- уметь обрабатывать массивы и объекты при помощи JavaScript в условиях сортировки, поиска, фильтрации, преобразования и представления данных в JSON

- уметь проектировать функции при помощи JavaScript в условиях разделения вычислений, обработки параметров и ограничения побочных эффектов

- уметь изменять веб-страницу при помощи DOM API в условиях выбора элементов, изменения текста, классов, значений форм и структуры документа

- уметь реализовывать обработку событий при помощи addEventListener в условиях проверки форм, поиска по таблице, добавления элементов и блокировки недопустимых действий

- уметь типизировать небольшие фрагменты клиентского кода при помощи TypeScript в условиях описания данных, параметров функций и элементов DOM

- уметь создавать начальные компоненты интерфейса при помощи React в условиях вывода списков, передачи свойств, изменения состояния и обработки событий

- уметь проверять клиентские сценарии при помощи инструментов разработчика браузера и контрольных пользовательских действий в условиях разных размеров экрана и ошибочного ввода

- уметь готовить техническую документацию по клиентской части веб-приложения в условиях описания структуры страниц, сценариев, ограничений и результатов проверки

Владеть:

- навыком создания семантической HTML-разметки, таблиц, форм и навигации веб-страницы

- навыком оформления веб-страниц средствами CSS с учетом типографики, блочной модели и визуальных состояний

- навыком разработки адаптивных раскладок средствами Flexbox, Grid и медиавыражений

- навыком реализации анимаций и визуальной обратной связи средствами CSS

- навыком программирования клиентских сценариев на JavaScript с использованием массивов, объектов, функций и строк

- навыком работы с DOM, событиями, формами, таблицами и динамическим изменением элементов страницы

- навыком применения начальной типизации TypeScript и начальной компонентной организации React

- навыком проверки клиентской части веб-приложения и подготовки технической документации

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 7 з.е. (252 академических часа(ов).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов		
	Всего	Семестр	
		№5	№6
Контактная работа при проведении учебных занятий (всего):	144	80	64
В том числе:			
Занятия лекционного типа	64	32	32
Занятия семинарского типа	80	48	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 108 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Клиентская часть веб-приложения и работа браузера Рассматриваемые вопросы: - роль клиентской части в клиент-серверной архитектуре; - загрузка HTML, CSS, JavaScript и дополнительных ресурсов браузером; - связь пользовательского действия, страницы, запроса и ответа сервера.
2	HTTP и структура веб-документа Рассматриваемые вопросы: - адрес ресурса, HTTP-запрос, HTTP-ответ, заголовки и коды состояния; - структура HTML-документа, дерево элементов и подключение внешних ресурсов; - требования к именованию файлов, путям и организации клиентского проекта.
3	Семантическая разметка HTML Рассматриваемые вопросы: - элементы заголовков, абзацев, списков, ссылок, изображений и текстового выделения; - семантические области страницы и их роль в структуре документа; - таблицы, объединение ячеек и представление табличных данных.
4	Формы и пользовательский ввод в HTML Рассматриваемые вопросы: - элементы input, select, textarea, fieldset и legend; - типы полей, обязательность ввода, ограничения и начальная проверка формы; - связь формы с пользовательским сценарием расчета, регистрации или выбора параметров.
5	Основы CSS и каскадное оформление страницы Рассматриваемые вопросы: - способы подключения CSS, каскад, наследование и специфичность; - селекторы, псевдоклассы, псевдоэлементы и состояния элементов; - оформление цвета, шрифтов, размеров, отступов, границ и таблиц.
6	Блочная модель и позиционирование элементов Рассматриваемые вопросы: - блочные, строчные и строчно-блочные элементы; - свойства display, float, clear и position; - влияние потока документа на расположение элементов страницы.
7	Гибкие и сеточные раскладки CSS Рассматриваемые вопросы: - построение одномерных раскладок средствами Flexbox; - построение двумерных раскладок средствами Grid; - выбор способа раскладки для типовых областей веб-страницы.
8	Адаптивность и анимация клиентского интерфейса Рассматриваемые вопросы: - медиавыражения и контрольные ширины для разных устройств; - изменение количества колонок, размеров и поведения элементов; - переходы, ключевые кадры и визуальная обратная связь.
9	Основы языка JavaScript Рассматриваемые вопросы: - подключение сценариев и выполнение кода в браузере; - переменные, типы данных, операции, сравнения и преобразование типов; - ветвления, циклы и базовые вычислительные сценарии.
10	Строки, массивы и объекты JavaScript Рассматриваемые вопросы: - методы строк, шаблонные строки и регулярные выражения; - массивы, обход, сортировка, вложенные массивы и преобразование данных; - объекты, свойства, JSON и представление предметных данных.

№ п/п	Тематика лекционных занятий / краткое содержание
11	Функции и организация клиентского кода JavaScript Рассматриваемые вопросы: - объявление функций, стрелочные функции и параметры по умолчанию; - остаточные параметры, arguments, область видимости и возвращаемые значения; - чистые функции, побочные эффекты и разбиение клиентского кода на части.
12	Объектная модель документа и события браузера Рассматриваемые вопросы: - выбор элементов страницы и изменение текста, значений, классов и атрибутов; - события ввода, изменения, нажатия, загрузки страницы и addEventListener; - построение интерактивного поведения без перезагрузки страницы.
13	Проверка форм и обработка данных на странице Рассматриваемые вопросы: - проверка электронной почты, пароля, флажков и обязательных полей; - блокировка действий при недопустимом вводе; - поиск, фильтрация и выделение строк таблицы средствами JavaScript.
14	Работа с внешними данными в JavaScript Рассматриваемые вопросы: - асинхронное выполнение и получение данных через Fetch API; - обработка успешного ответа, ошибки сети и некорректных данных; - обновление элементов DOM по результатам запроса к сервису.
15	Основы TypeScript в клиентской разработке Рассматриваемые вопросы: - назначение TypeScript и связь с JavaScript; - аннотации типов, интерфейсы, типизация функций, массивов и объектов; - типизация элементов DOM и обработчиков событий.
16	Введение в компонентную разработку с React Рассматриваемые вопросы: - компонент, свойства, состояние и повторное использование элементов интерфейса; - условное отображение, списки и обработка событий; - перенос части интерактивной страницы в начальную компонентную структуру.
17	Введение в React: JSX и Virtual DOM Рассматриваемые вопросы: - назначение React и место библиотеки в клиент-серверной архитектуре SPA; - синтаксис JSX, правила преобразования в вызовы createElement; - принцип работы Virtual DOM и алгоритм согласования (Reconciliation); - отличие декларативного и императивного подходов к построению интерфейса.
18	Компоненты, Props и однонаправленный поток данных Рассматриваемые вопросы: - функциональные компоненты и правила их именования; - передача свойств, деструктуризация props и children; - defaultProps, проверка типов через PropTypes и TypeScript; - паттерны композиции компонентов и разделения ответственности.
19	Локальное состояние и базовые хуки Рассматриваемые вопросы: - назначение хука useState и правила его вызова; - асинхронность обновления состояния и пакетная обработка; - функциональное обновление состояния через предыдущее значение; - назначение хука useRef для доступа к DOM и хранения мутирующих значений.
20	Побочные эффекты и жизненный цикл компонента Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- назначение хука useEffect и массив зависимостей; - функция очистки (cleanup) и отмена подписок; - обработка гонок данных (race conditions) при асинхронных запросах; - типичные сценарии: загрузка данных, подписки на события, таймеры.
21	Оптимизация рендеров и мемоизация Рассматриваемые вопросы: - понятие ссылочного равенства объектов и функций в JavaScript; - назначение хуков useMemo и useCallback; - компонент React.memo и поверхностное сравнение props; - правила применения мемоизации и случаи избыточной оптимизации.
22	Контекст и глобальное состояние приложения Рассматриваемые вопросы: - назначение хука useContext и создание провайдеров; - разделение стейта на несколько контекстов; - проблемы производительности Context API при частых обновлениях; - обзор альтернатив: Zustand, Redux Toolkit, Jotai и выбор решения под задачу.
23	Маршрутизация в React-приложениях Рассматриваемые вопросы: - назначение библиотеки React Router и настройка BrowserRouter; - компоненты Routes, Route, Link и NavLink; - вложенные маршруты и динамические параметры вида :id; - программная навигация через useNavigate и работа с query-параметрами.
24	Продвинутая маршрутизация и защита маршрутов Рассматриваемые вопросы: - layout-роуты и общие обвязки страниц; - защищенные маршруты (Private Routes) на основе роли пользователя; - обработка страницы 404 и редиректы через Navigate; - загрузка данных до отображения маршрута (loader, action).
25	Работа с формами в React Рассматриваемые вопросы: - контролируемые (Controlled) и неконтролируемые (Uncontrolled) компоненты; - обработка ввода, валидация и отображение ошибок; - назначение библиотек React Hook Form и Zod для валидации схем; - динамические поля формы и массовая обработка массивов данных.
26	Взаимодействие с REST API Рассматриваемые вопросы: - настройка экземпляра Axios и базовые конфигурации; - interceptors для добавления заголовков и обработки ошибок; - хранение JWT-токена и управление сессией пользователя; - отображение состояний загрузки, успеха и ошибки (loading, error, success).
27	Серверное состояние и кеширование данных Рассматриваемые вопросы: - назначение библиотеки TanStack Query (React Query); - хуки useQuery и useMutation, ключи запросов; - инвалидация кеша, фоновое обновление и stale-время; - бесконечная прокрутка (Infinite Query) и оптимистичные обновления UI.
28	Кастомные хуки и переиспользование логики Рассматриваемые вопросы: - правила создания кастомных хуков и именование use*; - извлечение бизнес-логики из UI-компонентов;

№ п/п	Тематика лекционных занятий / краткое содержание
	- типовые хуки: useFetch, useDebounce, useLocalStorage, useMediaQuery; - композиция хуков и передача зависимостей.
29	Стилилизация React-приложений Рассматриваемые вопросы: - CSS Modules и изоляция стилей компонентов; - подход CSS-in-JS на примере Styled Components; - utility-first подход с использованием Tailwind CSS; - динамические классы и библиотека clsx для условных стилей.
30	Тестирование React-приложений Рассматриваемые вопросы: - unit-тестирование хуков и утилитарных функций; - integration-тестирование компонентов через React Testing Library; - настройка Jest/Vitest и мокирование модулей; - мокирование серверных запросов с помощью msw (Mock Service Worker).
31	Оптимизация и Production-сборка Рассматриваемые вопросы: - ленивая загрузка компонентов через React.lazy и Suspense; - Code Splitting на уровне маршрутов и компонентов; - анализ размера бандла средствами rollup-plugin-visualizer; - оптимизация изображений, шрифтов и ресурсов третьих сторон.
32	Деплой и основы CI/CD для React-приложения Рассматриваемые вопросы: - переменные окружения и файлы .env для разных сред; - настройка сборки через Vite и публикация на Vercel/Netlify; - основы настройки GitHub Actions для автоматической сборки; - подготовка проекта к защите: README, демо-ссылка, описание сценариев.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Создание HTML-разметки текстовой страницы Студент создает HTML-документ с заголовками, абзацами, списками, ссылками и изображениями. В разметку добавляются section, span, strong и em. Страница проверяется в браузере на корректность структуры и отображения.
2	Создание семантической HTML-структуры и форм Студент размечает страницу с header, main, footer, aside, article и nav. Добавляются таблицы с объединением ячеек, форма расчета и форма регистрации. Поля формы группируются с использованием fieldset и legend.
3	Оформление страницы средствами CSS Студент подключает стили внешним файлом и задает цвета, шрифты, размеры, отступы, границы и скругления. Для интерактивных элементов добавляется состояние наведения. Таблица оформляется с чередованием строк.
4	Управление потоком и позиционированием элементов Студент изменяет поведение элементов через display, float, position и clear. Для очистки потока используется псевдоэлемент after. Результат проверяется на нескольких вариантах ширины окна браузера.

№ п/п	Наименование лабораторных работ / краткое содержание
5	Построение гибкой раскладки средствами Flexbox Студент создает блоки страницы с использованием display flex. Настраиваются направление, выравнивание, распределение свободного пространства, порядок, рост и сжатие элементов. Макет проверяется при изменении размеров окна.
6	Построение двумерной раскладки средствами Grid Студент создает сетку страницы с областями, строками, столбцами и промежутками. Используются единицы fr и объединение ячеек. В сетку помещаются основные области интерфейса.
7	Создание адаптивной страницы Студент добавляет медиавыражения для смартфона, планшета и настольного экрана. В зависимости от ширины меняются количество колонок, расположение блоков и размеры элементов. Поведение страницы проверяется средствами разработчика браузера.
8	Создание переходов и анимаций CSS Студент добавляет переходы для изменения состояния элементов. Через keyframes создаются появление, масштабирование и перемещение элементов. Для анимированного элемента задается понятная визуальная обратная связь.
9	Обработка данных средствами JavaScript Студент реализует вычисления с переменными, строками, числами и логическими выражениями. Используются if, тернарный оператор, циклы, Math, toFixed, typeof и преобразование типов. Результат выводится на страницу и проверяется на ошибочных данных.
10	Работа с массивами JavaScript Студент выполняет добавление, удаление, обход, сортировку и объединение элементов массива. Реализуются операции с двумерным массивом, включая заполнение и поворот. Результаты отображаются в виде таблицы или списка.
11	Проектирование функций JavaScript Студент создает обычные и стрелочные функции с параметрами по умолчанию и остаточными параметрами. В коде разделяются чистые вычисления и действия, изменяющие страницу. Для функций подбираются контрольные входные данные.
12	Работа с объектами и структурированными данными Студент создает объекты и массивы объектов для хранения предметных данных. Реализуются добавление, поиск, сортировка и преобразование данных в JSON. На странице отображается отформатированный результат обработки.
13	Подключение JavaScript и обработка простых событий Студент подключает сценарий к странице и выводит диагностические сообщения в консоль. Добавляются обработчики onclick, oninput, onchange, onload и addEventListener. Содержимое выбранных элементов изменяется через textContent и value.
14	Управление DOM, формами и таблицами Студент реализует калькулятор с обработкой ввода, форму регистрации с проверкой электронной почты, пароля и флажка, поиск по строкам таблицы и добавление пунктов в выпадающий список. Для изменения отображения используются children, classList, contains и создание элементов. Кнопки блокируются при недопустимых данных.
15	Начальная типизация клиентского кода средствами TypeScript Студент переносит часть JavaScript-кода на TypeScript. Для данных формы, массива объектов, функций и элементов DOM задаются типы и интерфейсы. Ошибки типизации исправляются до запуска страницы в браузере.
16	Создание начального интерфейса на React Студент создает небольшое React-приложение с несколькими компонентами. В компонентах используются свойства, состояние, список элементов, условное отображение и обработчики событий. Часть ранее созданного интерактивного сценария переносится в компонентную структуру.

№ п/п	Наименование лабораторных работ / краткое содержание
17	Инициализация React-проекта и базовая структура страниц Студент создает React-приложение средствами Vite с подключением TypeScript, ESLint и Prettier. Настраиваются алиасы путей, переменные окружения и структура каталогов проекта. Создается layout-компонент с боковым меню, шапкой и основной областью. Подключается React Router и настраиваются базовые маршруты страниц. Результат проверяется в браузере на корректность отображения и переходов.
18	Создание переиспользуемых UI-компонентов с типизацией Студент разрабатывает набор UI-компонентов: кнопки, карточки, бейджи, поля ввода и модальные окна. Для каждого компонента задаются строгие типы Props с использованием Generics и Union Types. Реализуются варианты отображения через props variant и size. Типизация проверяется компилятором TypeScript до запуска приложения.
19	Локальное состояние и интерактивные элементы интерфейса Студент реализует переключение светлой и темной темы, аккордеоны, выпадающие списки и модальные окна с использованием useState. Добавляется локальная фильтрация и сортировка таблиц. Функциональное обновление состояния применяется для счетчиков и списков. Поведение компонентов проверяется при многократных взаимодействиях пользователя.
20	Побочные эффекты и работа с DOM и событиями Студент применяет useEffect для установки заголовка вкладки document.title, отслеживания прокрутки страницы и автофокуса на полях ввода. Реализуются подписки на глобальные события клавиатуры с функцией очистки cleanup. Хук useRef используется для хранения ссылок на DOM-элементы и мутирующих значений. Корректность очистки эффектов проверяется средствами React DevTools.
21	Оптимизация рендеров и мемоизация компонентов Студент выявляет избыточные перерисовки компонентов средствами React DevTools Profiler. Применяет React.memo для чистых компонентов, useMemo для тяжелых вычислений при фильтрации больших списков и useCallback для стабилизации ссылок на функции-обработчики. Сравнивается производительность до и после оптимизации. Обосновывается целесообразность применения мемоизации для каждого случая.
22	Глобальное состояние приложения через Context и Zustand Студент создает контексты для профиля пользователя, настроек интерфейса и выбранных элементов. Реализуются провайдеры и хук useConsumer для доступа к данным. Для решения проблемы производительности при частых обновлениях подключается Zustand с разделением стейта на отдельные срезы. Корректность обновления компонентов проверяется при изменении глобального состояния.
23	Многостраничное приложение с вложенной маршрутизацией Студент настраивает вложенные маршруты с общим layout для раздела заявок. Реализуются страницы списка заявок и детальной страницы заявки с динамическим параметром :id. Применяются useNavigate для программной навигации и useSearchParams для работы с query-параметрами. Добавляется обработка страницы 404 через маршрут *. Переходы проверяются на корректность сохранения состояния.
24	Защищенные маршруты и разграничение доступа Студент создает компонент ProtectedRoute, проверяющий наличие токена авторизации и роль пользователя. Реализуются редиректы неавторизованных пользователей на страницу входа через Navigate. Разграничивается доступ к разделам для ролей «Админ» и «Пассажир». Поведение приложения проверяется при входе под разными учетными записями и при прямом обращении к защищенным URL.
25	Разработка сложных форм с валидацией Студент интегрирует библиотеку React Hook Form для управления состоянием формы создания заявки. Подключается Zod для описания схемы валидации и отображения ошибок полей. Реализуются динамически добавляемые поля и условное отображение блоков формы. Форма

№ п/п	Наименование лабораторных работ / краткое содержание
	проверяется на корректность обработки валидных и невалидных данных, а также на блокировку отправки при ошибках.
26	Интеграция REST API и управление сессией Студент настраивает экземпляр Axios с базовым URL и interceptors для добавления заголовков авторизации. Реализуются страницы входа и регистрации с сохранением JWT-токена в localStorage. Добавляются индикаторы загрузки Skeleton и обработка состояний loading, error, success. Корректность запросов проверяется при успешной авторизации, истечении токена и ошибках сети.
27	Кеширование данных и мутации через TanStack Query Студент переносит загрузку списков заявок на хук useQuery с настройкой ключей запросов и stale-время. Реализуются мутации создания и удаления заявок через useMutation с инвалидацией кеша. Добавляется бесконечная прокрутка через useInfiniteQuery и оптимистичное обновление UI. Поведение кеша проверяется при повторных переходах между страницами и фоновом обновлении данных.
28	Создание кастомных хуков для переиспользования логики Студент извлекает бизнес-логику из компонентов в кастомные хуки useDebounce для поиска, useClickOutside для закрытия выпадающих списков, useLocalStorage для сохранения настроек и useMediaQuery для адаптивного поведения. Хуки типизируются через TypeScript и покрываются сценариями использования. Переиспользование логики проверяется в нескольких компонентах проекта.
29	Стилилизация и адаптивное отображение интерфейса Студент подключает Tailwind CSS и переводит компоненты на utility-first подход. Реализуется адаптация табличных данных под мобильные устройства с превращением строк в карточки. Применяется библиотека clsx для условных классов. Отображение проверяется на контрольных ширинах экрана смартфона, планшета и настольного монитора средствами разработчика браузера.
30	Тестирование компонентов и сценариев приложения Студент настраивает Vitest и React Testing Library для интеграционного тестирования компонентов. Пишутся тесты на сценарии авторизации, создания заявки и обработки ошибок API. Подключается Mock Service Worker для мокирования серверных запросов. Тесты запускаются и проверяют корректность отображения и взаимодействия компонентов.
31	Ленивая загрузка и оптимизация сборки Студент применяет React.lazy и Suspense для ленивой загрузки тяжелых страниц и компонентов. Настраивается Code Splitting на уровне маршрутов. Анализируется размер бандла и выявляются крупные зависимости. Оптимизируются изображения и шрифты. Сборка проверяется на корректность загрузки компонентов при переходах между маршрутами.
32	Деплой приложения и подготовка к защите Студент настраивает переменные окружения для сред разработки и продакшена через файлы .env. Выполняется сборка проекта средствами Vite и публикация на платформу Vercel или Netlify. Настраивается GitHub Actions для автоматической сборки при пуше в основную ветку. Подготавливается README с описанием сценариев, демо-ссылкой и инструкцией по запуску. Проект защищается перед преподавателем.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Архитектура DOM и делегирование событий. Студент реализует динамическое добавление/удаление элементов (список задач) без фреймворков. Применяет делегирование событий на родительском элементе. Анализирует производительность и перерисовку дерева.
2	Продвинутая работа с массивами и объектами. Студент выполняет трансформацию данных (map, filter, reduce) для подготовки JSON-ответов

№ п/п	Тематика практических занятий/краткое содержание
	сервера к отображению в таблице. Реализует группировку и сортировку вложенных структур данных.
3	Асинхронный JavaScript и Fetch API. Студент настраивает взаимодействие с mock-API. Реализует обработку состояний: загрузка, успех, ошибка. Использует Promise.all для параллельных запросов и AbortController для отмены запросов.
4	Замыкания, контекст и функциональное программирование. Студент пишет функции высшего порядка, реализует паттерн «модуль» и каррирование. Разбирает проблемы утечек памяти и области видимости, что является базой для понимания хуков React.
5	Валидация форм и паттерны проектирования UI. Студент создает сложную форму с зависимыми полями, кастомной валидацией и блокировкой кнопки отправки. Применяет паттерны Debounce и Throttle для полей поиска.
6	Основы TypeScript для веб-интерфейсов. Студент типизирует ответы API, описывает интерфейсы для компонентов DOM, использует Generics для универсальных функций обработки данных и Union Types для состояний загрузки.
7	Модульная архитектура и сборка (Vite/npm). Студент инициализирует проект через Vite, настраивает ESLint и Prettier. Разделяет код на ES-модули, настраивает алиасы путей и переменные окружения (.env).
8	Подготовка к фреймворку: эмуляция реактивности. Студент пишет мини-библиотеку, которая отслеживает изменения в объекте состояния и автоматически обновляет соответствующие узлы DOM (понимание Virtual DOM и реактивности).

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным работам.
3	Выполнение курсовой работы.
4	Подготовка к промежуточной аттестации.
5	Подготовка к текущему контролю.

4.4. Примерный перечень тем курсовых работ

Разработка клиентской части веб-приложения для каталога учебных материалов.

Разработка клиентской части веб-приложения для расписания занятий.

Разработка клиентской части веб-приложения для регистрации участников мероприятия.

Разработка клиентской части веб-приложения для личного кабинета обучающегося.

Разработка клиентской части веб-приложения для учета заявок на обслуживание.

Разработка клиентской части веб-приложения для витрины транспортных сервисов.

Разработка клиентской части веб-приложения для поиска и фильтрации маршрутов.

Разработка клиентской части веб-приложения для управления списком задач.

Разработка клиентской части веб-приложения для интерактивного справочника.

Разработка клиентской части веб-приложения для формы обратной связи.

Разработка клиентской части веб-приложения для учета бронирований.

Разработка клиентской части веб-приложения для панели уведомлений.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Янцев, В. В. Разработка web-страниц на HTML, CSS и JavaScript : учебное пособие для вузов / В. В. Янцев. — Санкт-Петербург : Лань, 2024. — 148 с. — ISBN 978-5-507-49640-2. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/422462 (дата обращения: 19.06.2026)
2	Диков, А. В. Web-программирование на JavaScript : учебное пособие для вузов / А. В. Диков. — Санкт-Петербург : Лань, 2026. — 436 с. — ISBN 978-5-507-54625-1. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/518225 (дата обращения: 19.06.2026)
3	Диков, А. В. JavaScript. Взаимодействие с веб-страницей и сервером, асинхронное программирование : учебное пособие для вузов / А. В. Диков. — Санкт-Петербург : Лань, 2026. — 212 с. — ISBN 978-5-507-54188-1. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/515055 (дата обращения: 19.06.2026)
4	Государев, И. Б. Введение в веб-разработку на языке JavaScript : учебное пособие / И. Б. Государев. — Санкт-Петербург : Лань, 2022. — 144 с. — ISBN 978-5-8114-3539-5. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206588 (дата обращения: 19.06.2026)
5	Петракова, Н. В. Основы CSS : учебно-методическое пособие / Н. В. Петракова. — Брянск : Брянский ГАУ, 2023 — Часть 2 — 2023. — 107 с. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/385640 (дата обращения: 19.06.2026)

6	Никулова, Г. А. Web-дизайн. Приемы адаптивного Web-дизайна: технологии Flexbox и CSS Grid : учебное пособие / Г. А. Никулова, А. С. Терлецкий. — Липецк : Липецкий ГПУ, 2021. — 69 с. — ISBN 978-5-907461-41-3. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/228698 (дата обращения: 19.06.2026)
7	Сан, Ф. М. Разработка приложений с Quarkus и React : руководство / Ф. М. Сан ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2023. — 294 с. — ISBN 978-5-93700-207-5. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/455309 (дата обращения: 19.06.2026)
8	Розенталс, Н. Изучаем Typescript 3 / Н. Розенталс ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2019. — 608 с. — ISBN 978-5-97060-757-2. — Текст : электронный	Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131712 (дата обращения: 19.06.2026)

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

ЭБС Лань – https://e.lanbook.com/.

Образовательная платформа Юрайт – https://urait.ru/.

Единый реестр российских программ для ЭВМ и баз данных – https://reestr.digital.gov.ru/reestr/.

Профессиональные стандарты и квалификации, справочная информация КонсультантПлюс – https://www.consultant.ru/document/cons_doc_LAW_157436/.

Документация HTML MDN – https://developer.mozilla.org/en-US/docs/Web/HTML.

Документация CSS MDN – https://developer.mozilla.org/en-US/docs/Web/CSS.

Документация JavaScript MDN – https://developer.mozilla.org/en-US/docs/Web/JavaScript.

Документация TypeScript – https://www.typescriptlang.org/docs/.

Документация React – https://react.dev/learn.

Документация Git – https://git-scm.com/doc.

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Операционные системы – Astra Linux, ALT Linux, РЕД ОС, Debian GNU/Linux.

Среда разработки – Visual Studio Code, WebStorm Community, Chromium или Firefox Developer Edition.

Технологии клиентской разработки – HTML, CSS, JavaScript, TypeScript, React.

Средства сборки и проверки – Node.js, npm, Vite, ESLint, Git.

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для лаьораторных занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Курсовая работа в 5 семестре.

Экзамен в 5 семестре.

Зачет в 6 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»
Института железнодорожного
транспорта Е.А. Заманова

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Директор ЦС IT - программ ВИШ О.Б. Проневич

Председатель учебно-методической
комиссии Н.А. Андриянова