

Мидл фронтенд-разработчик

- 01 Кому подойдёт курс
- Курс рассчитан на разработчиков с опытом во фронтенде больше полугода. Однако он также подойдёт тем, кто только переходит во фронтенд, но у кого есть программистский бэкграунд.
-
- 02 Чему научитесь
- Если вы уже знакомы с фронтендом, то на курсе вы углубите знания о современных фронтенд-технологиях, освоите продвинутые инструменты и подходы, научитесь работать с архитектурой приложений и оптимизировать код. Всё это повысит вашу востребованность на рынке.
- Если вы только переходите во фронтенд, то быстро адаптируетесь к новой экосистеме, систематизируете знания в HTML, CSS и JavaScript и освоите современные практики и инструменты для работы.
-
- 03 Как проходит обучение
- Курс «Мидл фронтенд-разработчик» состоит из двух частей, в каждой из которых студенты фокусируются на одном большом проекте. Первый (мессенджер) выполняется индивидуально, а второй (игра) — уже в команде. Так за время учёбы вы разработаете два полноценных веб-приложения, которые можно будет положить в портфолио.
- Если вы выбрали дополнительный модуль по React, то вам предстоит поработать и над третьим заданием — пользовательским приложением для космической бургерной. Вас ждёт живая практика, основанная на решении реальных рабочих задач.
- На всех тарифах у вас будут опытные наставники и менторы, с которыми вы будете общаться в чате и на вебинарах. С ними можно будет разбирать сложности в теории и обсуждать промежуточные результаты работы над проектами.
-

Мидл фронтенд-разработчик

Сравнение версий

	Название тарифа	Название тарифа
	«Мидл фронтенд-разработчик»	«Мидл фронтенд-разработчик» + модуль по React
Для кого	Для тех, кто хочет прокачать навыки фронтенд-разработки до мидл-уровня, чтобы уверенно решать сложные рабочие задачи и успешно расти в профессии.	Для тех, кто хочет глубже освоить React, прокачав свои знания и навыки в веб-разработке: применять современные подходы и создавать сложные интерфейсы.
Длительность обучения	5 месяцев	8,5 месяцев
Практические работы с ревью	2 проекта и 9 ревью	3 проекта и 15 ревью
Преимущества тарифа	<p>На тарифе «Мидл фронтенд-разработчик» вы научитесь разрабатывать современные веб-приложения с помощью JavaScript, TypeScript, HTML и CSS, освоите инструменты сборки и автоматизации и глубоко разберётесь в клиент-серверном взаимодействии, протоколах HTTP, WebSocket и принципах REST. Вы познакомитесь с тем, как тестировать приложения и защищать их от кибератак, настраивать серверный рендеринг (SSR) и офлайн-режим через Service Workers, подключать PostgreSQL и работать с базой данных через ORM. Вы также освоите DevOps-инструменты: Docker, Nginx, CI/CD, OAuth2 и JWT.</p> <p>По итогам курса вы разработаете два полноценных приложения (индивидуально и в команде) и прорешаете алгоритмические задачи, что станет вашим преимуществом при прохождении собеседований.</p>	<p>На тарифе с React вы изучите дополнительные технологии на продвинутом уровне. Так что, помимо программы по фронтенду, вам предстоят темы по Redux, TypeScript, WebSocket и роутингу. Вы научитесь типизировать компоненты и состояние с помощью TypeScript, настраивать аутентификацию и авторизацию пользователей, реализовывать онлайн-взаимодействие через WebSocket, деплоить и поддерживать React-приложения в рабочем окружении, а также автоматизированно их тестировать. Ещё вас ждёт больше практики, так что вы создадите дополнительное многопользовательское приложение на React.</p>

Мидл фронтенд-разработчик

5–8,5 месяцев

продолжительность курса

Тарифы «Мидл фронтенд-разработчик»
и «Мидл фронтенд-разработчик + модуль по React»

2 ЧАСА

00

Бесплатный вводный модуль

2 НЕДЕЛИ

01

Сборка проекта и шаблонизаторы

2 НЕДЕЛИ

02

TypeScript и компонентный подход

2 НЕДЕЛИ

03

Роутинг и работа с запросами

2 НЕДЕЛИ

04

Безопасность и эксплуатация

2 НЕДЕЛИ

★

Каникулы.
Командная работа

2 НЕДЕЛИ

05

Основы React

2 НЕДЕЛИ

06

Продвинутый React, Redux и тестирование React-приложений

2 НЕДЕЛИ

07

SSR и браузерные хранилища

2 НЕДЕЛИ

08

Контейнеризация и базы данных

2 НЕДЕЛИ

09

CI/CD и подготовка к защите

40 ЧАСОВ

★

Алгоритмы

Мидл фронтенд-разработчик

8,5 месяцев

продолжительность курса

Только тариф «Мидл фронтенд-разработчик и модуль по React»

<p>3 НЕДЕЛИ</p> <p>01</p> <p>Основы React</p>	<p>2 НЕДЕЛИ</p> <p>02</p> <p>Управление состоянием и пользовательский опыт</p>	<p>2 НЕДЕЛИ</p> <p>03</p> <p>Роутинг и безопасность</p>	<p>2 НЕДЕЛИ</p> <p>04</p> <p>TypeScript</p>
<p>2 НЕДЕЛИ</p> <p>05</p> <p>Использование WebSocket в React-приложении</p>	<p>2 НЕДЕЛИ</p> <p>06</p> <p>Организация, тестирование и деплой React-приложений</p>		

Бесплатная часть. Как всё устроено

2 часа

Вы подробно познакомитесь с форматом курса и программой. В частности, узнаете, как будут проходить спринты и практика и кто будет сопровождать вас во время учёбы.

Ещё вас ждут два урока из основной части курса. Так вы поймёте, что из себя представляет интерактивный учебник Практикума и подходит ли он вам.

Вы также пройдёте входной тест. С его помощью вы оцените сложность курса, а мы будем уверены, что у будущих студентов хватит знаний и навыков, чтобы пройти курс.

Всё вместе поможет понять, насколько вам подходит курс, и выбрать оптимальный тариф.

Первая часть. Самостоятельный проект «Веб-мессенджер»

8 недель

В первой части курса студентам предстоит самостоятельно создать полноценный SPA — веб-мессенджер со стандартными функциями (регистрация, авторизация, список чатов и обмен сообщениями). Для этого вы разработаете собственный фреймворк с использованием JavaScript, TypeScript, CSS и браузерного API. А вот использовать сторонние библиотеки не получится, зато так вы поймёте, что для веб-приложения достаточно нативного JavaScript.

Содержание модуля

01. Сборка проекта и шаблонизаторы	<ul style="list-style-type: none">• Что такое Frontend• Модули• Node.js. Управление зависимостями через npm• Разворачивание проекта с нуля с помощью Vite• Хостинг для кода и проекта• Архитектура CSS• Структурирование кода по методологии БЭМ• Доступные интерфейсы: A11y• Шаблонизаторы: Handlebars• Практика: обращение к элементам массива	<p>В первом спринте вам предстоит узнать, как устроен курс и учебный план, какие задачи у вас будут, какие возможности даёт платформа Практикума, к кому обращаться за помощью и какие есть форматы поддержки. Ещё в первом спринте вы выясните, чем занимается фронтенд-разработчик, и познакомитесь с ключевыми понятиями. А потом начнётся собственно учёба.</p> <p>Вы поймёте, что такое модули и зачем они нужны в JavaScript, научитесь подключать сторонние библиотеки, управлять компонентами проекта и разрабатывать удобные для всех сайты. Настроите рабочее окружение и автоматизируете работу с повторяющимися элементами интерфейса. Познакомитесь с платформами для хранения кода и публикации проектов, методами организации CSS-кода и популярной методологией БЭМ для читаемого и поддерживаемого CSS. Вас также ждут практические задачи на обращение к элементам массива в JavaScript.</p> <p>Проектная работа: Создайте проект с использованием Handlebars, разработаете прототипы и основные страницы веб-мессенджера, настройте сборку через Vite, стилизуете проект и организуете автодеплой на Netlify.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none">• Node.JS• npm• Vite• PostCSS• БЭМ• A11y• Handlebars	2 недели
------------------------------------	---	--	----------

02. TypeScript и компонентный подход

- Продвинутый JavaScript
- ООП в JavaScript
- Начало знакомства с TypeScript
- Архитектура фронтенд-приложений
- Браузерные события
- Компоненты
- Codestyle и линтеры

Во втором спринте вы начнёте изучать продвинутые возможности JavaScript и современные подходы к разработке интерфейсов. Лексическое окружение, функции, асинхронные операции, структуры данных и паттерны проектирования. Вы научитесь реализовывать свои промисы и преобразовывать колбэки, а ещё познакомитесь с концепциями абстракции, инкапсуляции, наследования и полиморфизма. Особое внимание в спринте будет уделено TypeScript и методам поддержки качества кода с помощью линтеров и форматеров. Познакомитесь с архитектурой приложений и принципами жизненного цикла компонентов и освоите работу с браузерными событиями и создание вложенных и реактивных компонентов.

Проектная работа:

Переведёте клиентский код на TypeScript, освоите компонентный подход и соберёте базовые компоненты будущего веб-мессенджера. Вам также предстоит реализовать страницу чатов с формами и их валидацией и выстроить проект с чёткой структурой и понятной архитектурой.

Инструменты и технологии:

- Продвинутый JavaScript
 - ООП
 - TypeScript
 - Архитектура фронтенд-приложений
 - Работа браузера
 - Prettier
 - ESLint
 - Stylelint
-

2 недели

03. Роутинг и работа с запросами	<ul style="list-style-type: none"> • Роутинг • Клиент-серверное взаимодействие (HTTP, REST и API) • Работа с запросами • Использование API-чатов • Практика: isEqual и queryString 	<p>Третий спринт посвящён ключевым темам современной фронтенд-разработки: роутингу, клиент-серверному взаимодействию и работе с API. Сначала вы разберётесь с маршрутизацией: реализуете переходы между страницами без перезагрузки, освоите History API и построите собственный роутер. Затем перейдёте к протоколу HTTP: узнаете, из чего состоят запрос и ответ, какими бывают методы, заголовки и статусы, чем различаются версии протокола и как HTTPS обеспечивает безопасное соединение. Ещё вы научитесь тестировать запросы, ускорять передачу данных и применять принципы REST при проектировании клиент-серверных приложений. Вас ждёт работа с реальными запросами: вы создадите собственные классы и утилиты для работы с API, наладите обработку и хранение данных и отправку запросов на сервер. Особое внимание уделите авторизации, работе с файлами и интеграции API для чатов.</p> <p>Проектная работа: По итогам третьего спринта вы реализуете маршрутизацию веб-мессенджера: переходы между страницами с сохранением истории и корректным обновлением. Настройте работу с HTTP-API для регистрации, авторизации, управления профилем и чатами, обеспечив взаимодействие с сервером через собственный класс для запросов.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • History API • HTTP • REST • CORS 	2 недели
----------------------------------	---	---	----------

04. Безопасность и эксплуатация

- WebSocket
- Продолжение знакомства с TypeScript
- Тестирование с помощью Mocha и Chai
- Эксплуатация фронтенд-приложений
- Вопросы информационной безопасности
- Размещение статики: CDN
- Создание собственных утилит и функций (omit)

В этом спринте вы разберётесь с протоколом WebSocket: особенностями двусторонней связи между клиентом и сервером и реализацией обмена сообщениями в режиме реального времени. Вы научитесь тестировать мессенджер с Mocha и Chai и освоите основы информационной безопасности. На реальных кейсах из индустрии разберёте, как подходить к защите комплексно. Вновь поработаете с TypeScript. На этот раз вас ждут объединения типов, обработка данных и внедрение стандартов ESNEXT, чтобы ваш код был надёжным и масштабируемым. Отдельно разберётесь с эксплуатацией фронтенд-приложения: автоматизацией запуска скриптов, проверками состояния проекта перед публикацией и подготовкой приложения к устойчивой работе. И наконец, вы научитесь публиковать статические ресурсы через CDN, чтобы ускорить загрузку и повысить устойчивость к нагрузкам.

2 недели

Проектная работа:

Вы разработаете систему тестирования ключевых модулей веб-мессенджера и обеспечите их качество с помощью Mocha и Chai. Реализуете полную функциональность чата с обменом сообщениями в реальном времени и внедрите защиту от XSS-атак и продвинутую обработку ошибок.

Инструменты и технологии:

- TypeScript
- WebSocket
- Mocha
- Chai
- Эксплуатация
- Precommit
- Инъекции
- XSS
- CSRF

Задание на каникулы: модуль «Командная работа»

2 часа

- Как быть в команде
- Командные проекты и стартовый план действий
- Командный Git
- Как проводить ревью друг друга

В этом модуле вы познакомитесь с принципами командной работы и инструментами совместной разработки, включая командный Git и ревью кода. Освоите базовый план запуска проекта и начнёте вместе разрабатывать концепцию будущей игры.

Вторая часть. Командный проект «Веб-игра»

10 недель

Вторая часть курса будет посвящена созданию полноценной браузерной игры. Это будет командный проект. Вы познакомитесь с дополнительными для фронтенда направлениями: прототипированием интерфейсов, основами информационной безопасности, DevOps и принципами объектно-ориентированного программирования. Работа над проектом поможет развить навыки взаимодействия в команде и научит брать ответственность за продукт и его развитие.

Содержание модуля

05. Основы React	<ul style="list-style-type: none">• Сбор требований и проектирование функционала приложения• Как ставить задачи по проекту• Проработка пользовательских сценариев• Canvas API• Погружение в React• React Router• Игровые механики• Собственная реализация throttle	<p>В пятом спринте вас ждёт комплексная разработка современных веб-приложений: от постановки задач и проработки пользовательских сценариев до внедрения игровых механик. Вы научитесь декомпозировать задачи, распределять зоны ответственности, определять приоритеты и формулировать критерии готовности фич — всё, чтобы сделать процесс разработки управляемым и прозрачным.</p> <p>Затем вы познакомитесь с Canvas API. Узнаете, как создавать графику и работать с анимациями в браузере. В дальнейшем это позволит внедрять интерактивные элементы и реализовать базовые игровые механики.</p> <p>Ключевой блок спринта — погружение в React. Создание интерфейса на компонентах, связывание экранов через React Router, настройка роутинга, работа с хуками и обработка событий — это станет прочным фундаментом для динамичных и отзывчивых приложений.</p> <p>Специальные блоки спринта будут посвящены игровым механикам. Вы научитесь внедрять базовые элементы игрового процесса. Например, реализовать throttle, чтобы управлять частотой событий и повышать отзывчивость интерфейса.</p> <p>Проектная работа: В этой проектной работе начинается разработка веб-игры: настройте маршрутизацию, создадите главную страницу (мини-лендинг) и страницы ошибок 400 и 500, логина, форума, регистрации, лидерборда и профиля. Реализуете авторизацию и валидацию форм. Особое внимание предстоит также уделить механике игры и экранам начала и завершения.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none">• Планирование• Управление проектами• User Scenario• User Story• User Story Map• User Story Mapping• Canvas API• React• React Router• GameDev	2 недели
------------------	---	---	----------

<p>06. Продвинутый React, Redux и тестирование React-приложений</p>	<ul style="list-style-type: none"> • React под капотом: HOC, контексты и оптимизация • Хранение данных: Redux и селекторы • Тестирование в React • Service Workers • Web API для игры 	<p>Теперь настало время углубиться в продвинутые возможности React и архитектуру современного фронтенда. Сначала предстоит разобраться, как работает React под капотом: компоненты высшего порядка (HOC), использование контекста для управления данными между уровнями приложения и эффективные техники оптимизации рендеринга с помощью мемоизации и хуков.</p> <p>Затем вы научитесь хранить и структурировать данные. Для этого освоите работу с Redux и создание селекторов, которые помогают удобно извлекать информацию из состояния. Нельзя обойтись без тестирования в React. Вы познакомитесь с основными подходами и инструментами и напишете тесты с Jest, чтобы проверить логику компонентов и игровых механик.</p> <p>Завершится спринт Service Workers. Вы узнаете, как повысить производительность и надёжность приложения, обеспечив автономную работу и кеширование данных, а также попытаетесь подключать выбранные Web API, чтобы расширить возможности вашей браузерной игры.</p> <p>Проектная работа: Вам предстоит реализовать ключевую игровую механику и ввести состояния старта и конца игры для управления процессом. Сфокусируетесь на визуальной части и внедрите Service Workers для повышения производительности. Добавьте хранилище игрового состояния и покроете движок тестами для стабильной работы.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • React • Стейт-менеджмент • Redux • Jest • Service Workers • Web API
---	--	---

07. SSR и браузерные хранилища	<ul style="list-style-type: none"> • Хранилища в Web • Web API для игры (продолжение) • Server Side Rendering • Настройка аутентификации и авторизации 	<p>Седьмой спринт будет посвящён современным способам работы с данными, расширению функциональности браузерных игр и профессиональным инструментам рендеринга и безопасности.</p> <p>Начнётся всё с работы с хранилищами в веб-приложениях. Вы узнаете, как сохранять и извлекать данные через локальное хранилище, имитировать работу СУБД в браузере и обмениваться данными между разными вкладками. Эти инструменты позволят создавать надёжные и удобные интерфейсы. Затем вы продолжите знакомство с современными Web API для игровых и мультимедийных задач. Освойте Media API, работу с файлами через FileSystem API, подключение игровых контроллеров с Gamepad API и управление курсором через Pointer Lock API. Вас также ждут технологии для 3D-графики (WebGL, WebVR API) и повышения производительности с помощью Web Workers и Ambient Light Events API.</p> <p>Важный этап — настройка Server Side Rendering (SSR). Вы сгенерируете разметку на стороне сервера и разберётесь, как это повышает скорость загрузки страниц и улучшает SEO. Ещё вам предстоит внедрить SSR с использованием Express, Redux и современных инструментов для управления стилями и роутингом. Вы сравните SSR и Client Side Rendering, изучите лучшие практики объединения данных, работы с метатегами и подключением промежуточного ПО для обработки запросов.</p> <p>Отдельно остановимся на настройке аутентификации и авторизации, работе с популярными паттернами (OAuth и JWT), а также на более нестандартных подходах к безопасности веб-приложений.</p> <p>Проектная работа: Вы настроите Express для серверного рендеринга (SSR), интегрируете Redux и React Router в SSR для управления состоянием и маршрутизацией, реализуете авторизацию по OAuth и подключите API лидерборда.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • localStorage • sessionStorage • Express • SSR • Серверный роутинг • Helmet • Аутентификация и авторизация 	2 недели
--------------------------------	--	--	----------

<p>08. Контейнеризация и базы данных</p>	<ul style="list-style-type: none"> • Работа с базами данных • Подготовка окружения: Docker, Docker Compose и PostgreSQL • Основы PostgreSQL • Интеграция PostgreSQL с Node.js • Темизация приложения 	<p>В восьмом спринте вы погрузитесь в разработку серверной части, управление базами данных и подготовку профессионального окружения для приложения. Вы разберётесь, зачем фронтенд-разработчику вообще нужны базы данных, освоите синтаксис SQL и познакомитесь с подходами NoSQL. Научитесь создавать таблицы, работать с типами данных, использовать связи и индексы, применять подзапросы и операции JOIN, чтобы строить сложные, связанные структуры данных. Следующий шаг — подготовить окружение для разработки. Здесь вас поджидает контейнеризация с Docker и Docker Compose, управление переменными окружения, запуск нескольких контейнеров в общей сети и основы виртуализации.</p> <p>Отдельный блок будет посвящён PostgreSQL: типам данных, созданию и модификации таблиц, настройке ключей, работе с транзакциями и интеграции с Node.js. Вы освоите управление базой через ORM, моделирование сущностей и связей для сложных запросов и реализуете CRUD-операции.</p> <p>В завершение спринта речь пойдёт про темизацию приложения. Вы допишете необходимые API-эндпоинты, настроите запуск и поддержку серверной части в изолированном окружении и получите инструменты для развития полнофункционального веб-приложения на современной архитектуре.</p> <p>Проектная работа: Работа над игрой подходит к концу. Осталось настроить Docker для запуска и управления PostgreSQL, обеспечить взаимодействие между пользователями и обмен сообщениями, внедрить переключение тем для удобства интерфейса, реализовать интерфейс с поддержкой эмодзи и настроить серверную инфраструктуру для стабильной работы форума.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • SQL • Контейнеризация • Docker • Docker Compose • PostgreSQL • ORM • Темизация 	<p>2 недели</p>
--	---	--	-----------------

09. CI/CD и подготовка к защите	<ul style="list-style-type: none"> • Погружение в Nginx • Настройка инфраструктуры в Яндекс Облаке • Командный зачёт 	<p>Игра почти готова: осталось только внести финальные правки, развернуть проект в облачной среде, настроить автоматическую сборку и обеспечить устойчивость и безопасность приложения. Этому и будет посвящён финальный спринт.</p> <p>Вы погрузитесь в работу с Nginx: узнаете, зачем он нужен, запустите сервер, настройте раздачу статических файлов, проксирование запросов, логирование, балансировку между сервисами, кеширование, сжатие данных и работу по HTTPS и HTTP/2.</p> <p>Дальше нужно будет настроить инфраструктуру в Яндекс Облаке. Однако сначала нужно разобраться с основами: облачными хранилищами, доменами, лимитами и автоматизацией развёртывания. Затем вы настройте необходимые облачные ресурсы, научитесь управлять расходами, деплоить проект через CI/CD и обеспечивать постоянную интеграцию и доставку новых версий. Отдельно остановитесь на подходах Infrastructure as Code для тиражирования и масштабирования инфраструктуры.</p> <p>И конечно, вас ждёт защита командного проекта.</p> <p>Проектная работа: Чтобы завершить разработку игры, вам нужно настроить политику безопасности для ограничения источников ресурсов, обеспечить надёжное и защищённое соединение сервера и расширить функциональность, добавив дополнительные сервисы. Также в задачах проектной работы — повысить безопасность и стабильность проекта, описав уязвимости и методы их устранения.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • Nginx • Деплой • CI/CD • Яндекс Облако 	2 недели
---------------------------------	---	--	----------

Дополнительный модуль. Алгоритмы

40 часов

Это необязательный, но очень полезный курс. Он охватывает широкий спектр тем от базовых структур данных до хеш-таблиц и алгоритмов сжатия и поиска. Курс можно проходить параллельно с основным в своём темпе. Теорию вы будете закреплять в тренажёре на практических задачах.

Особое внимание будет уделено алгоритмам сортировки, поисковым задачам и методам работы со строками и их свойствами. Вы познакомитесь с алгоритмами сжатия (например, реализуете алгоритм Хаффмана) и разберёте типовые задачи с собеседований, отрабатывая решение нестандартных комплексных кейсов.

К концу курса вы научитесь писать эффективный, оптимизированный код, это поможет уверенно решать широкий спектр алгоритмических задач, проходить технические собеседования и строить карьеру.

Содержание модуля

- Основные функции сложности
- Введение в структуры данных
- Оценка сложности алгоритма: асимптотический анализ
- Основные понятия: передача данных и рекурсия
- Пузырьковая сортировка
- Анализ сложности алгоритма
- Сортировка вставками
- Связный список
- Стек и очередь
- Быстрая сортировка
- Сортировка слиянием
- Алгоритмы и сортировки: повторение
- Хеш-таблицы
- Методы разрешения коллизий и операции с хеш-таблицей
- Деревья
- Операции с бинарным деревом поиска и бинарной кучей
- Графы
- Структуры данных: повторение
- Сжатие данных. Алгоритм Хаффмана
- Алгоритмы поиска. Бинарный поиск
- Поиск в ширину (breadth-first search, BFS)
- Поиск в глубину (depth-first search, DFS)
- Реальные задачи на собеседованиях

Модуль по React

3,5 месяца

В этом модуле вы будете разрабатывать многопользовательское приложение — «Бургерная на краю Вселенной». Конечно, оно будет на React. Вы познакомитесь с библиотекой, научитесь управлять состоянием и маршрутизацией, обеспечивать безопасность, работать с TypeScript и WebSocket, а также организовывать, тестировать и развёртывать приложение.

Содержание модуля

01. Основы React	<ul style="list-style-type: none">• SPA и фреймворки• Настройка и сборка фронтенд-проекта• Основы React• Состояние и эффекты в React• Компоненты и жизненный цикл• Изоляция компонентов и Storybook• Продвинутый React	<p>В первом спринте курса по React вы познакомитесь с архитектурой одностраничных приложений и популярными фронтенд-фреймворками: React, Angular и Vue. Освоите настройку фронтенда на Vite, а также работу с виртуальным DOM, JSX, компонентами, пропсами, списками и событиями.</p> <p>Далее вы перейдёте к хукам: разберёте базовые и кастомные, паттерны передачи данных между компонентами, контроль их жизненного цикла и управление состоянием. Особое внимание уделите обработке событий, формам, коллекциям и стилизации с помощью CSS-модулей.</p> <p>Вы освоите Storybook — инструмент изолированной разработки компонентов, что позволит создавать масштабируемые и независимые части интерфейса.</p> <p>Отдельный блок будет посвящён продвинутым аспектам React: синтетическим событиям, порталам, компонентам высшего порядка (HOC), мемоизации, конкурентному рендерингу и современным инструментам оптимизации производительности и работе с внешними источниками данных.</p> <p>К концу первого спринта вы получите фундаментальные навыки проектирования современных SPA, научитесь строить масштабируемые интерфейсы и оптимизировать рабочие процессы.</p> <p>Проектная работа: В проектной работе первого спринта начинается работа над приложением бургерной. Вы создадите его главную страницу и модальные окна, получите первые данные с сервера и оптимизируете вычисления в компонентах.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none">• JSX• Хуки• Vite• Stylelint, ESLint, Prettier• Storybook• CSS-модули• React DevTools	3 недели
------------------	--	--	----------

02. Управление состоянием и пользовательский опыт

- Глобальное состояние
- Redux
- Redux Toolkit
- RTK Query
- Drag-and-drop

Теперь настало время учиться управлять глобальным состоянием на практике. Вы отработаете хуки, контекст и редьюсеры, а также освоите Redux (actions, reducers, middleware) и работу с DevTools. В блоке Redux Toolkit вас ждут инструменты быстрой интеграции: слайсы для состояния, асинхронные экшены и работа с формами. Разберётесь с RTK Query, чтобы получать и обновлять данные и обрабатывать ошибки. Наконец, перейдёте к drag-and-drop: изучите основы перетаскивания в React, примените библиотеку React DnD, свяжете её с Redux и добавите продвинутые интерактивные механики. В результате вы получите ключевые стратегии управления состоянием для современного React-проекта и сможете реализовывать сложные сценарии работы с данными и интерфейсом.

Проектная работа:

Работа над приложением продолжается. Теперь вы добавите глобальное состояние, соберёте первый бургер с перетаскиванием и точно так же настроите сортировку ингредиентов.

Инструменты и технологии:

- React Context
 - State Reducer
 - Кастомные хуки
 - Redux
 - Redux Toolkit
 - Redux DevTools
 - RTK Query
 - Drag-and-drop
-

2 недели

03. Роутинг
и безопасность

- React Router
- Авторизация

Третий спринт будет посвящён навигации в приложении и внедрению аутентификации для управления доступом. Сначала вы разберётесь с основами маршрутизации: чем отличается клиентская и серверная, как реализовывать переходы через History API. А дальше вы научитесь создавать вложенные маршруты, регистрировать ссылки, передавать параметры между страницами и работать с модальными окнами. В результате вы сможете настраивать продвинутую навигацию, включая ленивую загрузку компонентов для оптимизации производительности, и познакомитесь с современными подходами к рендерингу на сервере и клиенте.

Во второй части спринта сосредоточимся уже на авторизации. Вас ждёт разбор процесса входа и выхода из системы, токены доступа и их обновление. Ещё вы изучите защищённые маршруты и управление доступом по ролям с использованием JWT, cookie и localStorage.

Проектная работа:

В этой проектной работе вам предстоит добавить в приложение авторизацию: создать страницы входа, регистрации, восстановления пароля и профиля. Ещё декларативно или через массивы вы опишете маршруты, настроите поддержку JWT-токенов и защиту маршрутов и доработаете модальные окна.

Инструменты и технологии:

- React Router
 - Redux Toolkit
 - RTK Query
 - History API
 - React Suspense
 - Методы аутентификации и авторизации
 - Redux
-

2 недели

- Основы TypeScript
- Типизация кода
- TypeScript в React
- Типизация Redux

В четвертом спринте вы погрузитесь в работу с TypeScript и его интеграцию в React-проекты. Разберётесь с принципами динамической и статической типизации, научитесь создавать типы и интерфейсы и подключать TypeScript к проекту.

Дальше поговорим про базовые типы и типизацию функций, массивов и объектов, использование специальных и пользовательских типов и реализацию интерфейсов для расширения возможностей приложения.

Отдельный блок — работа с дженериками и приведением типов, типизация DOM-элементов, стандартных объектов JavaScript и условных типов, а также создание и применение файлов деклараций (d.ts) для сторонних библиотек.

Вы научитесь типизировать разные части React-приложения: функциональные и классовые компоненты, хуки, обработку форм и событий, контекст, а также продвинутые паттерны — `forwardRef`, `createRef`, `Error Boundaries` и компоненты высшего порядка (HOC).

Вы примените принципы типизации в рутинге и интегрируете сложные типы непосредственно в интерфейс приложения.

Также вы научитесь грамотно типизировать все ключевые элементы Redux: опишете экшены, слайсы и стор на TypeScript, корректно типизируете хуки `useDispatch` и `useSelector` для безопасной и удобной работы с глобальным состоянием приложения.

Проектная работа:

Так как спринт посвящён TypeScript, в проектной работе вы переведёте разрабатываемое приложение на этот язык и проверите корректность и полноту типов, чтобы повысить удобство, безопасность и поддерживаемость кода.

Инструменты и технологии:

- TypeScript
 - Типизация (`useContext`, `forwardRef`, `createRef`, `Error Boundaries`, HOC)
-

<p>05. Использование WebSocket в React-приложении</p>	<ul style="list-style-type: none"> • WebSockets • WS и Redux: middleware • WS и RTK Query 	<p>В пятом спринте важным шагом станет освоение WebSocket: вы узнаете, как открывать и закрывать соединения, обрабатывать события и передавать данные в реальном времени между клиентом и сервером. Рассмотрите отладку соединения и тестирование WebSocket через инструменты разработчика.</p> <p>Далее пойдёт интеграция WebSocket с Redux: вы реализуете цепочку работы через middleware, создадите необходимые экшены и слайсы, научитесь подключать middleware к стору для управления событиями обмена данными. Особое внимание вы уделите обработке разрыва соединения и обновлению токенов для безопасности сессии.</p> <p>В завершении спринта разберётесь, как подключать WebSocket к RTK Query: изучите, как управлять жизненным циклом кеша, настроить обмен сообщениями и грамотно обрабатывать ошибки и разрывы соединений.</p> <p>Проектная работа: Работа над приложением бургерной начинает подходить к концу. Теперь вам нужно будет добавить динамику и настроить работу с WebSocket с использованием Redux Toolkit или RTK Query. Кроме того, вы реализуете ленты заказов и проверите типизацию глобального хранилища, чтобы обеспечить стабильную работу приложения.</p> <p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • TypeScript • Redux • WebSocket • RTK Query • Redux Toolkit 	<p>2 недели</p>
---	--	--	-----------------

06. Организация, тестирование и деплой React-приложений	<ul style="list-style-type: none"> • Организация кода и архитектура • Тестирование: Vitest и Playwright • Подготовка к продакшену • Деплой на GitHub Pages • Деплой на сервер 	<p>В финальном спринте вы освоите ключевые принципы современной архитектуры React-приложения: научитесь структурировать проект по слоям, фичам и модулям, создавать чистую архитектуру, организовывать компоненты и сервисы, следовать лучшим практикам раздельной логики и UI, а также внедрять строгий режим для безопасности и контроля кода.</p> <p>Блок тестирования познакомит с Vitest и Playwright. Вы отработаете создание юнит-, интеграционных и системных тестов, моки и проверки состояний и автоматизируете контроль качества.</p> <p>В разделах, посвящённых продакшену, погрузитесь в оптимизацию (Helmet, динамический импорт, бандлинг, аналитику, раздельные dev/prod-окружения) и CI/CD для автоматизированных тестов и публикации. Научитесь выкладывать проекты на GitHub Pages через Vite и выполнять финальный деплой на сервер: домен, HTTPS с Let's Encrypt, Nginx и завершение развёртывания.</p>	2 недели
		<p>Проектная работа: Чтобы довести приложение бургерной до готовности, осталось только повысить надёжность разработки. Для этого вам сначала предстоит написать юнит-тесты на Vitest, а затем — интеграционные. В финале вы задеплоите приложение, выбрав удобный способ развёртывания.</p>	
		<p>Инструменты и технологии:</p> <ul style="list-style-type: none"> • FSD • TDD • Vitest • Playwright • MSW • fixtures • Helmet • CI 	
