

Программа профессионального обучения по курсу «Разработка сайтов и веб-приложений» (16+) (редакция июль 2026 года)

Программа рассчитана на 18 месяцев и предназначена для тех, кто хочет системно изучить frontend-разработку и научиться создавать современные сайты и веб-приложения.

За время обучения вы пройдёте путь от вёрстки страниц на HTML и CSS до разработки сложных интерфейсов на JavaScript, React и TypeScript.

Вы научитесь работать с макетами, создавать адаптивные сайты, добавлять интерактивность, подключаться к внешним API, управлять состоянием приложений, тестировать и оптимизировать код, а также публиковать готовые проекты в интернете.

После прохождения основных трёх модулей вы сможете создавать полноценные frontend-приложения, близкие по структуре к проектам, встречающимся в современной веб-разработке.

Дополнительно можно пройти четвёртый модуль по PHP, MySQL и Laravel. Он позволит освоить серверную разработку и создавать полноценные Full-Stack-приложения, объединяющие frontend, backend и базу данных.

Обучение начинается с нуля. От вас требуется уверенно пользоваться компьютером, регулярно заниматься и выполнять задания.

Подробнее о курсе: <https://itpark32.ru/web.html>

Отзывы об обучении: https://vk.com/topic-58525095_28834899

1-й модуль: «Верстка сайтов HTML и CSS»

Длительность освоения модуля: 6 месяцев

Этот модуль нужен, чтобы научиться создавать структуру и внешний вид современных сайтов.

Вы изучите HTML и CSS, научитесь работать с текстом, изображениями, таблицами и формами, создавать адаптивные страницы и верстать сайты по готовым макетам.

Также вы познакомитесь с Git, Figma, Bootstrap, SCSS, методологией БЭМ и публикацией готовых проектов в интернете.

Список изучаемых тем в модуле:

HTML

1. Как функционирует Internet
2. Что такое HTML
3. Структура HTML документа
4. Теги
5. Разметка текста
6. Атрибуты тегов
7. Разметка страницы
8. Работа с изображениями
9. Работа с файловой системой, пути к файлам
10. Работа со ссылками
11. Работа со списками
12. Работа с таблицами
13. Работа с формами
14. HTML – video, iframe, embed

15. SVG-графика
16. Оформление сайта
17. Табличная верстка

CSS

1. Что такое CSS
2. Базовые CSS-свойства
3. Базовые селекторы
4. Псевдоэлементы и псевдоклассы
5. Продвинутые селекторы атрибутов и состояний
6. Наследование и каскадирование
7. Оформление текста
8. Блочная модель документа
9. Позиционирование элементов
10. Работа с фонами
11. Градиентная заливка
12. Переходы
13. Трансформации
14. Анимация
15. Декоративные эффекты
16. CSS Переменные
17. Flexbox
18. CSS Grid Layout
19. Адаптивный дизайн, медиа запросы

Расширенные возможности и инструменты автоматизации

1. Автоматизированная верстка с помощью фреймворков (на примере Bootstrap)
2. Препроцессоры SASS, SCSS
3. Работа с Git
4. Деплой проектов
5. Работа с графическими редакторами (на примере Figma)
6. Pixel Perfect верстка

7. Технология БЭМ

Чему вы научитесь:

- Создавать правильную и понятную структуру веб-страницы с помощью HTML.
- Верстать адаптивные сайты, которые корректно отображаются на компьютерах, планшетах и смартфонах.
- Работать с Flexbox, CSS Grid, анимациями и современными возможностями CSS.
- Переносить дизайн-макеты из Figma в готовые веб-страницы.
- Организовывать CSS-код с помощью БЭМ и SCSS.
- Хранить проекты в Git и публиковать готовые сайты в интернете.

Примеры типовых проектов:

1. Одностраничные сайты-лэндинги с фиксированными размерами. Например, сайты барбершопов, курсов и онлайн-образования, сервисы заказа еды.
2. Многостраничные сайты с возможностью путешествия между страницами. Например, интернет-магазины, социальные сервисы.
3. Адаптивные сайты с применением продвинутых технологий языка CSS, а также внешних библиотек и фреймворков для верстки.

2-й модуль: «Frontend разработка на JavaScript»

Длительность освоения модуля: 6 месяцев

Этот модуль нужен, чтобы перейти от статичных страниц к интерактивным сайтам и веб-приложениям.

Вы изучите JavaScript с нуля, научитесь составлять алгоритмы, работать с данными, изменять содержимое веб-страницы, обрабатывать действия пользователей и сохранять информацию в браузере.

Также вы разберётесь в асинхронном программировании, клиент-серверном взаимодействии и подключении к внешним интернет-сервисам через API.

Список изучаемых тем в модуле:

Основы JavaScript

1. Введение в JavaScript
2. Использование консоли
3. JavaScript в браузере
4. Переменные и константы – var, let & const
5. Примитивные типы данных
6. Приведение типов
7. Числа и математические функции
8. Методы работы со строками
9. Условные операторы и операторы сравнения
10. Сложные типы данных
11. Массивы и методы работы с массивами
12. Объекты и методы работы с объектами
13. Циклы
14. Функции
15. Контекст функций
16. Области видимости
17. Встроенные объекты и их методы
18. Встроенные в JavaScript функции
19. Отладка JavaScript-кода
20. Подъем (hoisting) переменных и функций
21. Работа с метриками
22. Валидация форм
23. Валидация с помощью регулярных выражений

DOM и браузер

- 24. Браузерная модель документа (BOM)
- 25. Объектная модель документа (DOM)
- 26. Перемещение по DOM дереву
- 27. Манипуляции с DOM деревом. Добавление, изменение и удаление DOM-элементов
- 28. Подходы к созданию DOM-элементов
- 29. Коллекции и свойства элементов
- 30. Работа с браузерными событиями
- 31. Делегирование событий
- 32. Обработка исключений

Хранение данных

- 33. Продвинутая работа с массивами.
- 34. Колбэки (функции обратного вызова)
- 35. Деструктуризация
- 36. Spread-синтаксис
- 37. Локальное хранение данных
- 38. Спецификация ECMAScript
- 39. Новые возможности ES6+

Асинхронность и работа с сетью

- 40. Асинхронность
- 41. Протокол HTTP и форматы данных
- 42. Инструменты для работы с сетевыми запросами
- 43. Работа с API, клиент-серверная связь, RestAPI

Архитектура JavaScript-приложений

- 44. Объектно-ориентированное программирование
- 45. Архитектура приложений и современные подходы к их созданию
- 46. Модули ES и Webpack
- 47. SPA

Чему вы научитесь:

- С помощью JavaScript Вы изучите алгоритмику и исследуете типы данных, научитесь работать с условиями, циклами и функциями. Сделаете свои проекты динамичными.
- Сможете добавить открывающееся меню, диалоговые окна, слайдеры и другие интерактивные элементы на сайт.
- Вы освоите сложные концепции языка: объектно-ориентированное программирование, асинхронность и обмен данными с сервером.
- Научитесь работать с сетью, отправлять запросы на сервер, подключаться к сторонним API, проектировать REST API приложения.
- Эти инструменты помогут вам эволюционировать из верстальщика в веб-разработчика.

Примеры типовых проектов:

1. Сайты с возможностью купить товар или услугу, записаться на мероприятие.
2. Приложения для сайта, например, калькулятор бюджета, менеджер задач, парсер.
3. Визуальные анимированные компоненты для сайта, типа галерея изображений, слайдер, меню, форма, модальное окно.
4. Мини-игры, реализующие логику JavaScript, например, игра в кубики, угадай число, платформер.
5. RESTful API интерфейсы, реализующие такие расширенные возможности для вашего сайта или приложения, как перевод статей, получение новостей, сохранение данных на удаленный хостинг, авторизация пользователей.

3-й модуль: «Frontend разработка на React и TypeScript»

Длительность освоения модуля: 5 месяцев

Этот модуль нужен, чтобы перейти от приложений на чистом JavaScript к разработке современных компонентных frontend-приложений.

Вы изучите React и его экосистему, освоите TypeScript, Redux Toolkit, маршрутизацию, работу с формами, авторизацию и взаимодействие с REST API.

Также вы научитесь проектировать структуру крупных frontend-приложений, тестировать компоненты, анализировать производительность и оптимизировать готовые проекты.

Список изучаемых тем в модуле:

React и его экосистема

1. Основы React:
2. Введение в React и его экосистему.
3. Установка и настройка среды разработки.
4. JSX: синтаксис и особенности.
5. Компоненты: функциональные и классовые.
6. Пропсы и состояние (props и state).
7. Работа с компонентами
8. Управление состоянием компонентов.
9. Передача данных между компонентами.
10. Жизненный цикл компонентов.
11. События и обработка событий в React.
12. Динамическое рендеринг и условный рендеринг.
13. React Hooks
14. Введение в хуки (Hooks) и их преимущества.
15. useState и управление состоянием.

16. `useEffect` и управление побочными эффектами.
17. Дополнительные хуки: `useContext`, `useReducer`, `useMemo` и `useCallback`.
18. Создание пользовательских хуков.
19. Управление состоянием в приложении
20. Подходы к управлению состоянием в React-приложениях.
21. Использование Context API для глобального состояния.
22. Введение в Redux: назначение и принципы работы.
23. Использование Redux Toolkit для упрощения работы с Redux.
24. Маршрутизация в React-приложениях
25. Введение в React Router.
26. Настройка маршрутов и навигация между страницами.
27. Динамические маршруты и маршрутизация с параметрами.
28. Обработка ошибок и редиректы.
29. Защищенные маршруты и аутентификация.
30. Работа с формами и валидация:
31. Создание и управление формами в React.
32. Контролируемые и неконтролируемые компоненты формы.
33. Валидация форм с использованием библиотек, таких как Formik и Yup.
34. Отправка данных формы и работа с API.
35. Подключение к API и работа с данными
36. Fetch API и Axios для выполнения HTTP-запросов.
37. Управление асинхронными операциями в React.
38. Использование хуков для работы с данными (`useEffect`, `useState`).
39. Обработка ошибок и загрузка данных.
40. Кеширование данных с использованием React Query.
41. Продвинутое концепции React
42. Оптимизация производительности компонентов.
43. Мемоизация компонентов и хуков.
44. Паттерны проектирования в React (HOC, Render Props, Container-Presenter).
45. Управление сторонними библиотеками и их интеграция в проект.
46. Согласование и управление состоянием при изменениях.
47. Инструменты и экосистема React
48. Настройка и использование Create React App.

49. Конфигурация Webpack и Babel для React-проектов.
50. Использование ESLint и Prettier для автоматизации кода.
51. Развертывание и публикация React-приложений

Typescript

1. Основы TypeScript
2. Установка и настройка TypeScript.
3. Типизация: основные типы данных (string, number, boolean и др.).
4. Компиляция TypeScript в JavaScript.
5. Типы данных и аннотации типов
6. Прimitives типы данных.
7. Типы для массивов, объектов и функций.
8. Псевдонимы типов (type aliases).
9. Интерфейсы (interfaces) и их использование.
10. Объектно-ориентированное программирование с TypeScript
11. Классы и объекты.
12. Модификаторы доступа (public, private, protected).
13. Наследование и абстракция.
14. Интерфейсы и реализации.
15. Абстрактные классы.
16. Расширенные типы
17. Объединенные типы (union types).
18. Пересекающиеся типы (intersection types).
19. Типы-кортежи (tuples).
20. Enum (перечисления).
21. Типы утверждений (type assertions).
22. Функции и обобщенные типы
23. Аннотация типов для функций.
24. Функции с неопределенным числом аргументов (rest parameters).
25. Функции обратного вызова (callback functions).
26. Обобщенные типы (generics) и их использование в функциях и классах.
27. Работа с модулями
28. Экспорт и импорт модулей.

29. Пространства имен (namespaces).
30. Подключение внешних модулей и библиотек с использованием TypeScript.
31. Типы Utility и другие полезные инструменты
32. Частичные типы (Partial).
33. Обязательные типы (Required).
34. Типы для работы с интерфейсами (Pick, Omit).
35. Типы условных выражений (Conditional types).
36. Встроенные и пользовательские utility типы.
37. Работа с DOM и типизация в браузере
38. Типизация событий и элементов DOM.
39. Работа с API браузера с использованием TypeScript.
40. Интеграция TypeScript с проектами на JavaScript и React
41. Подключение TypeScript в существующий проект на JavaScript.
42. Настройка TypeScript в проектах на React.
43. Типизация компонентов, пропсов и стейтов в React.
44. Инструменты разработки и тестирования
45. Настройка и использование ESLint и Prettier для TypeScript.
46. Автоматизация сборки с помощью Webpack и других инструментов.
47. Тестирование TypeScript-кода с использованием Jest.

Тестирование и оптимизация производительности

11. Введение в тестирование приложений
12. Зачем нужно тестирование: основные цели и задачи.
13. Обзор различных видов тестирования (юнит-тесты, интеграционные тесты, end-to-end тесты).
14. Настройка окружения для тестирования:
15. Установка и настройка тестовых фреймворков (Jest, React Testing Library).
16. Юнит-тестирование компонентов React
17. Написание тестов для функциональных и классовых компонентов.
18. Мокающие и шпионские функции.
19. Проверка рендеринга, пропсов и состояния компонентов.

20. Тестирование пользовательских взаимодействий:
21. Симуляция событий (клики, ввод текста, прокрутка и т.д.).
22. Тестирование изменения состояния при взаимодействии с компонентами.
23. Написание тестов для форм, кнопок и других интерактивных элементов.
24. Тестирование асинхронных операций
25. Тестирование компонентов, использующих `fetch` или `axios` для взаимодействия с API.
26. Использование мока для имитации API-запросов.
27. Обработка ошибок и тестирование сценариев сбоев.
28. Интеграционное тестирование в React
29. Тестирование взаимодействия нескольких компонентов.
30. Тестирование маршрутизации (React Router).
31. Проверка взаимодействия с глобальным состоянием (Redux, Context API).
32. Оптимизация производительности React-приложений
33. Основы оптимизации производительности.
34. Использование `React.memo`, `useMemo`, `useCallback` для оптимизации рендеринга.
35. Ленивая загрузка компонентов и кода (React.lazy, Suspense).
36. Анализ и мониторинг производительности
37. Введение в инструменты для анализа производительности (React Profiler, Chrome DevTools).
38. Использование Lighthouse для аудита производительности.
39. Метрики производительности: Time to Interactive (TTI), First Contentful Paint (FCP), Largest Contentful Paint (LCP).
40. Оптимизация загрузки и рендеринга:
41. Оптимизация загрузки ресурсов (изображения, шрифты, скрипты).
42. Уменьшение размера бандла с помощью code splitting и tree shaking.
43. Оптимизация использования стилей и CSS-in-JS.
44. Кэширование и управление состоянием:
45. Использование кэширования данных для оптимизации производительности.

46. Работа с локальным состоянием и глобальными состояниями в контексте производительности.

Чему вы научитесь:

- Вы освоите React и его экосистему, научитесь создавать приложения из компонентов, которые можно переиспользовать на разных страницах и в разных проектах.
- Сможете разрабатывать сложные интерактивные интерфейсы, работать с формами, маршрутизацией, авторизацией, состоянием приложения и внешними API.
- Освоите TypeScript и научитесь использовать строгую типизацию, интерфейсы, дженерики и другие возможности языка для создания надёжного и поддерживаемого кода.
- Научитесь интегрировать TypeScript в проекты на JavaScript и React, типизировать компоненты, данные, события и ответы сервера.
- Освоите модульное и интеграционное тестирование React-приложений, научитесь проверять компоненты, пользовательские действия и работу с API.
- Сможете анализировать производительность приложения, находить причины лишних перерисовок и медленной загрузки страниц.
- Научитесь использовать кеширование, ленивую загрузку компонентов, разделение кода и другие способы оптимизации приложения.
- Полученные навыки позволят вам создавать масштабируемые frontend-приложения, которые проще развивать, тестировать и поддерживать.

Примеры типовых проектов:

1. Планировщик задач с категориями, статусами, фильтрацией и сохранением данных.

2. Платформа для бронирования услуг или товаров с календарём, формами и личным кабинетом.
3. Интернет-магазин с каталогом, поиском, фильтрацией, сортировкой, корзиной и авторизацией.
4. Интерактивная доска объявлений или система управления контентом.
5. Приложение для поиска и подбора фильмов, книг или музыки с подключением к внешнему API.
6. Информационная панель с таблицами, графиками, статистикой и большим объёмом данных.
7. SPA-приложение с маршрутизацией, глобальным состоянием, тестированием и оптимизацией производительности.

После окончания 3-го модуля курса идёт выпускной проект. Выпускной проект по курсу предполагает выполнение предложенного на выбор итогового задания с целью подтверждения полученных знаний и полного освоения программы обучения по всем 3-м модулям.

Примеры выпускных проектов:

- SPA-приложение
- Многостраничный сайт
- Веб-сервис

4-й модуль: «Подготовка к трудоустройству в IT»

Длительность освоения модуля: 1 месяц

Этот модуль проводится во время написания вами выпускного проекта и помогает подготовиться к поиску первой работы или стажировки в сфере Frontend React-разработки.

Вы узнаете, как устроена работа современной IT-компании, подготовите резюме и портфолио, оформите профиль GitHub, разработаете стратегию поиска вакансий и потренируетесь проходить основные этапы отбора.

Список изучаемых тем в модуле:

1. Жизненный цикл IT-проекта и основные этапы разработки программного продукта.
2. Методологии Agile, Scrum и Kanban.
3. Роли и профессии специалистов в IT-команде.
4. Типы IT-компаний: продуктовые компании, аутсорсинг, стартапы, веб-студии и внутренние IT-отделы.
5. Процесс разработки новой функции: постановка задачи, проектирование, разработка, Code Review, тестирование и публикация.
6. Структура резюме начинающего Frontend-разработчика и описание проектного опыта.
7. Стратегия поиска работы, подбор вакансий и использование карьерных ресурсов.
8. Оформление профиля GitHub и подготовка проектов для портфолио.
9. Выполнение и презентация тестовых заданий.
10. Первое общение с HR и подготовка к техническому собеседованию.
11. Самопрезентация, обсуждение условий работы, заработной платы и предложения о трудоустройстве.

12. Первый рабочий день, онбординг, взаимодействие с наставником и правила общения внутри команды.

Чему вы научитесь

- Понимать, как устроены IT-компании: разбираться в жизненном цикле проекта, методологиях разработки, ролях специалистов и процессе создания новой функции — от постановки задачи до выпуска готового решения.
- Составлять резюме Frontend-разработчика, правильно описывать обучение, технический стек, выполненные проекты и полученный проектный опыт.
- Оформлять профиль GitHub и портфолио: выбирать лучшие проекты, составлять понятные README, описывать архитектуру, используемые технологии и порядок запуска приложений.
- Искать подходящие вакансии, выполнять тестовые задания и проходить общение с HR и техническими специалистами, отвечая на вопросы по HTML, CSS, JavaScript, TypeScript, React и веб-разработке.
- Обсуждать условия работы, заработную плату и предложение о трудоустройстве, а также правильно вести себя во время первого рабочего дня и адаптации в команде.

По итогам 3-х модулей Вы будете знать и уметь:

1. Создавать семантические HTML-страницы.
2. Верстать адаптивные сайты на CSS.
3. Использовать Flexbox, CSS Grid, Bootstrap и SCSS.
4. Работать с макетами в Figma.
5. Программировать на JavaScript.
6. Работать с условиями, циклами, функциями, массивами и объектами.

7. Изменять страницы с помощью DOM.
8. Обрабатывать браузерные события.
9. Работать с LocalStorage и SessionStorage.
10. Выполнять асинхронные запросы.
11. Подключаться к внешним API.
12. Создавать приложения на чистом JavaScript.
13. Создавать компонентные приложения на React.
14. Использовать React Hooks.
15. Реализовывать маршрутизацию, формы и авторизацию.
16. Использовать TypeScript в React-проектах.
17. Управлять состоянием с помощью Redux Toolkit.
18. Работать с React Query и Redux Toolkit Query.
19. Писать модульные и интеграционные тесты.
20. Анализировать и оптимизировать производительность.
21. Работать с Git и удалёнными репозиториями.
22. Публиковать готовые проекты в интернете.

Сферы трудоустройства React/TypeScript Frontend Developer:

1. IT-компании и стартапы
2. Финансовые учреждения и банки
3. Государственные учреждения и некоммерческие организации
4. Web-студии

React frontend-разработчик может работать в IT-компании, в продуктовой команде, в офисе, удалённо или участвовать в проектной разработке.

5-й модуль: «Backend разработка на PHP, MySQL и Laravel» (Дополнительный модуль.)

Проходится по желанию ученика после завершения основной программы.

Длительность освоения модуля: 6 месяцев

Этот модуль нужен, чтобы научиться создавать серверную часть сайтов и перейти от frontend-разработки к созданию полноценных Full-Stack-приложений.

Вы изучите PHP, MySQL и Laravel, научитесь обрабатывать запросы пользователей, хранить данные, создавать REST API, реализовывать авторизацию и связывать backend с React-приложениями.

PHP

1. Что такое хостинг и веб-сервер
2. Знакомство с OpenServer
3. Базовые операторы и функции
4. Переменные и константы
5. Работа с числами
6. Работа со строками
7. Работа с логическим типом данных
8. Условные операторы
9. Работа с массивами
10. Циклы
11. Функции: создание и использование
12. Связка PHP + HTML
13. Работа с формами. POST и GET запросы
14. Работа с файловой системой
15. Обработка ошибок
16. Регулярные выражения
17. Связка PHP + MySQL
18. Подготовленные запросы (PDO)

19. CRUD
20. Протокол HTTP
21. HTTP заголовки
22. Сессии
23. COOKIE
24. Взаимодействие JavaScript и PHP
25. AJAX технология
26. JSON
27. Парсинг сайтов
28. Работа с API
29. Клиент-серверное взаимодействие
30. ООП
31. Классы, методы и объекты
32. Наследование и абстрактные классы
33. Интерфейсы
34. Модификаторы доступа, геттеры и сеттеры
35. Паттерн проектирования MVC
36. Концепция MVC
37. Настройка и конфигурация сервера (.htaccess)
38. Автозагрузчик
39. Пространство имен
40. ЧПУ
41. Composer и подключаемые библиотеки
42. Работа с консолью

SQL

1. Теория баз данных
2. Знакомство с SQL
3. MySQL + phpMyAdmin
4. Структура БД
5. Работа с таблицами
6. Типы данных в SQL
7. Простые типы запросов

8. Сложные типы запросов
9. Отношения и соединение таблиц
10. Группировка
11. Модификация данных
12. Подзапросы
13. Функции и выражения, агрегация данных

Laravel

1. Что такое Laravel
2. Введение, установка и первичная настройка
3. Структура проекта на Laravel
4. Роутинг
5. Контроллеры и экшены
6. Контроллеры ресурсов
7. Миграции
8. Выполнение «сырых» SQL запросов
9. Конструктор запросов
10. Модели
11. Работа с базами данных. ORM-система Eloquent
12. Связи моделей
13. Работа с шаблонами. Шаблонизатор Blade
14. Макеты (Layout)
15. Работа с формами
16. Работа с файлами и изображениями
17. Управляющие конструкции
18. Валидация
19. Сессии
20. Работа с кэшем и Куки
21. Формирование ответа (Response)
22. Обработка запроса (Request)
23. Встроенные возможности и библиотеки Laravel
24. Посредники
25. Аутентификация и авторизация пользователей

26. Работа с админ панелью

Чему вы научитесь:

- Разрабатывать серверную часть современных сайтов и веб-приложений на PHP и Laravel.
- Обработать HTTP-запросы пользователей и формировать ответы для клиентской части приложения.
- Работать с файлами: загружать, сохранять, читать и изменять данные.
- Подключать MySQL к проекту и выполнять основные SQL-запросы для получения, добавления, изменения и удаления данных.
- Проектировать структуру базы данных и взаимодействовать с ней через Eloquent ORM.
- Применять принципы объектно-ориентированного программирования и организовывать код с помощью классов, интерфейсов и архитектуры MVC.
- Использовать маршрутизацию, контроллеры, модели, миграции, шаблоны Blade и систему лейаутов Laravel.
- Настраивать приложение с помощью переменных окружения и подключать сторонние библиотеки.
- Реализовывать регистрацию, аутентификацию и разграничение прав пользователей.
- Работать с сессиями, cookie, токенами и другими средствами хранения данных пользователя.
- Валидировать пользовательский ввод и обеспечивать базовую безопасность сайта или приложения.
- Создавать административные панели, REST API и связывать backend с frontend-приложением.

Примеры типовых проектов:

1. Интернет-магазин – платформа для продажи товаров и услуг.

2. Форум – ресурс для обсуждения темы или в чате по определенной тематике.
3. CRM-система – программный продукт для автоматизации управления отношениями с клиентами.
4. Парсеры данных для получения информации со сторонних сайтов.
5. Сайты-блоги, форумы.
6. Приложения для сайтов.

После окончания 4-го модуля курса идёт выпускной проект по всему курсу. Выпускной проект по курсу предполагает выполнение предложенного на выбор итогового задания с целью подтверждения полученных знаний и полного освоения программы обучения по всем 4-м модулям.

Примеры выпускных проектов:

- Интернет магазин
- CRM система
- Социальная сеть
- Веб-сервис

По итогам всего курса Вы будете знать и уметь:

1. HTML, CSS и JavaScript - языки, используемые для создания веб-страниц и интерактивных элементов на страницах.
2. Базы данных и языки запросов - SQL и другие, позволяющие хранить и обрабатывать данные на сервере.
3. Серверные технологии - PHP и другие, используемые для создания серверной инфраструктуры.
4. API и форматы данных - RESTful API, JSON и другие, используемые для обмена данными между клиентом и сервером.

5. Фреймворки и библиотеки - React и другие, используемые для ускорения процесса разработки и улучшения качества кода.
6. Версионирование кода и совместная работа - Git, GitHub и другие, используемые для управления и совместной разработки проектов.
7. Безопасность веб-приложений - знания об уязвимостях и методах защиты от них.
8. Оптимизация и тестирование - знания о методах оптимизации и тестирования веб-приложений для улучшения производительности и качества.

Сферы трудоустройства Full-stack web Developer:

1. IT-компании и стартапы
2. Финансовые учреждения и банки
3. Государственные учреждения и некоммерческие организации
4. Web-студии

Full-stack web разработчик может работать в IT-компании, в продуктовой команде, в офисе, удалённо или участвовать в проектной разработке.