МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 6-05-0611-01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Скриптовые языки программирования»

Тема: «Web-сайт «Образовательная платформа “Future Academy”»»

**Исполнитель**

студент 2 курса 1 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Васильев

подпись, дата

**Руководитель**

доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Жиляк

должность, учен. степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ Н. А. Жиляк

подпись дата инициалы и фамилия

**Оглавление**

[1. Постановка задачи 4](#_Toc188531289)

[2. Проектирование страниц веб-сайт. 5](#_Toc188531290)

[2.1. Выбор способа вёрстки 5](#_Toc188531291)

[2.2. Выбор шрифтового оформления 5](#_Toc188531292)

[2.3. Разработка пользовательских элементов 6](#_Toc188531293)

[2.4. Выводы 10](#_Toc188531294)

[3. Реализация структуры веб-сайта. 11](#_Toc188531295)

[3.1 Компонент App.js 11](#_Toc188531296)

[3.2 Диаграмма использования 12](#_Toc188531297)

[4. Тестирование веб-сайта 14](#_Toc188531298)

[4.1 Адаптивный дизайн веб-сайта 14](#_Toc188531299)

[5.Заключение 17](#_Toc188531300)

**Введение**

В условиях стремительного развития информационных технологий и глобального перехода в цифровую среду образовательные платформы играют всё более важную роль в жизни современного общества. Они позволяют людям, независимо от их географического положения, получать доступ к образовательным материалам, проходить курсы и развивать свои навыки. Создание образовательных платформ актуально в условиях востребованности непрерывного обучения и переквалификации.

Образовательная платформа **«Future Academy»** предоставляет пользователям возможность получать информацию о различных направлениях, отслеживать их прогресс и взаимодействовать с преподавателями. Для реализации данной платформы необходим функциональный, адаптивный и удобный веб-сайт, который будет отвечать современным требованиям пользовательского интерфейса и безопасности.

**Цель курсовой работы** — разработать веб-сайт для образовательной платформы **«Future Academy»**, информирующий пользователя о данной платформе, о преимуществах и недостатках тех или иных образовательных курсах.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Спроектировать структуру веб-сайта, включающую основные страницы: Главная, Курсы, Регистрация, Авторизация, Личный кабинет.
2. Реализовать пользовательский интерфейс с использованием современных веб-технологий (React, CSS, HTML).
3. Обеспечить адаптивность сайта для корректного отображения на различных устройствах (ПК, планшетах, смартфонах).
4. Провести тестирование функционала, кросс-браузерности и безопасности веб-сайта.

**Методы разработки** включают использование современных инструментов и технологий, таких как:

* React для клиентской части;
* HTML5 и CSS3 для разработки интерфейса;
* REST API для взаимодействия между клиентом и сервером.

В результате выполнения данной курсовой работы будет создан веб-сайт, который предоставляет пользователям доступ к образовательным материалам, позволяет регистрироваться, авторизоваться и управлять своими данными в удобном интерфейсе.

# **Постановка задачи**

Целью данного курсового проекта является разработка функционального и удобного веб-сайта для образовательной платформы **«Future Academy»**, который будет предоставлять пользователям доступ к информации о онлайн-курсах, регистрацию и авторизацию, а также общедоступную информацию о данной Академии.

Для достижения цели необходимо реализовать следующие функциональные возможности:

1. **Главная страница**:
   * Отображение общей информации о платформе.
   * Быстрые ссылки для регистрации и авторизации.
2. **Регистрация и авторизация**:
   * Форма регистрации новых пользователей с валидацией введенных данных.
   * Форма авторизации для существующих пользователей.
3. **Форма обратной связи**:
   * Возможность отправить запрос или сообщение через контактную форму.
   * Отправка данных формы на сервер с последующим их обработкой.
4. **Дополнительные требования**:
   * Кросс-браузерная совместимость (работа в Chrome, Firefox, Safari).
   * Адаптивный дизайн для корректного отображения на мобильных устройствах и планшетах.

# **Проектирование страниц веб-сайт.**

Этот раздел описывает процесс проектирования страниц веб-сайта и принятие ключевых решений, включая выбор технологий, стилистики, шрифтов, пользовательских элементов, а также выводы по итогам проектирования.

## **2.1. Выбор способа вёрстки**

Для реализации страниц веб-сайта **«Future Academy»** была выбрана адаптивная вёрстка с использованием технологий **HTML5** и **CSS3**. Адаптивная вёрстка обеспечивает корректное отображение сайта на устройствах с разными экранами: от настольных компьютеров до смартфонов.

Для организации элементов страниц использовались следующие подходы:

* **Flexbox** — для гибкой компоновки блоков.
* **Grid Layout** — для построения сложных макетов (например, размещения карточек курсов).
* **Медиа-запросы** — для настройки отображения блоков на различных экранах (например, изменение размеров шрифта, скрытие или изменение расположения элементов).

**Обоснование выбора:**

* **Flexbox** удобен для управления горизонтальными и вертикальными выравниваниями, что делает его идеальным для построения шапок, подвалов и простых интерфейсов.
* **Grid Layout** позволяет легко реализовать сложные макеты, такие как галерея курсов или список мероприятий.
* Использование медиа-запросов делает сайт адаптивным и удобным для работы на устройствах любого размера.

## **2.2. Выбор шрифтового оформления**

Для шрифтового оформления веб-сайта был выбран шрифт **Montserrat**, который хорошо читается на экранах и имеет современный вид.

**Шрифтовая схема:**

* **Заголовки**: Montserrat, полужирный шрифт, размер шрифта варьируется от 24px до 32px.
* **Основной текст**: Montserrat, обычный шрифт, размер шрифта 16px.
* **Дополнительный текст (подписи, ссылки)**: Montserrat, курсив или полужирный, размер шрифта 14px.

**Обоснование выбора:**

* Шрифт Montserrat отличается современным стилем, что соответствует общей концепции образовательной платформы.
* Единообразное использование шрифта на всех страницах улучшает визуальное восприятие.
* Регулируемый межстрочный интервал (line-height) и размер шрифта обеспечивают удобство чтения на всех устройствах.

## **2.3. Разработка пользовательских элементов**

Мы решили разбить страницы на компоненты по причине удобства разработки и поддержки проекта. Такой подход позволяет упростить повторное использование кода, так как компоненты можно применять в разных частях приложения, избегая дублирования. Разделение на компоненты делает проект более читаемым и структурированным, так как каждая часть интерфейса представлена логически обособленным блоком. Это упрощает тестирование и отладку, поскольку можно тестировать каждую часть отдельно, что значительно ускоряет поиск и исправление ошибок. Кроме того, компоненты позволяют нескольким разработчикам работать одновременно над разными частями проекта, не мешая друг другу. Такой подход обеспечивает гибкость и масштабируемость: добавление новых функций или изменения дизайна становятся проще, так как модификации затрагивают только определённые компоненты, не влияя на весь проект. Разделение на компоненты помогает создать модульный, поддерживаемый и расширяемый код.

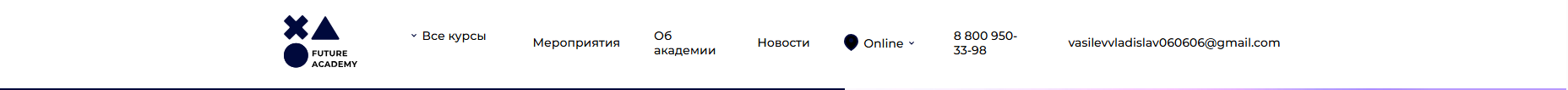


Рисунок 2.1— Компонент Header

Навигационное меню (Рисунок 2.1) позволяет пользователям легко переходить между основными страницами сайта, такими как Главная, Мероприятия, Об Академии, Новости и Регистрация. Оно реализовано с использованием Flexbox для горизонтального выравнивания элементов, а на мобильных устройствах трансформируется в бургер-меню для компактного отображения.

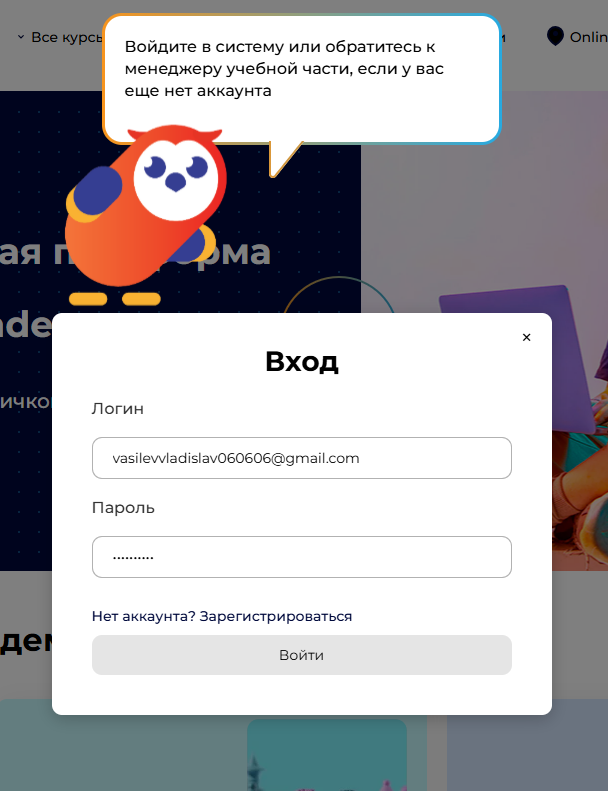


Рисунок 2.2 – Компонент Auth.js(Авторизация)

Формы Авторизации (Рисунок 2.2) и Регистрации (Рисунок 2.3) включают текстовые поля для ввода логина и пароля, а также валидацию данных для обеспечения корректного ввода. Для этих форм предусмотрены стилизованные кнопки с активными состояниями, которые меняют цвет и масштаб при наведении, добавляя интерактивности.

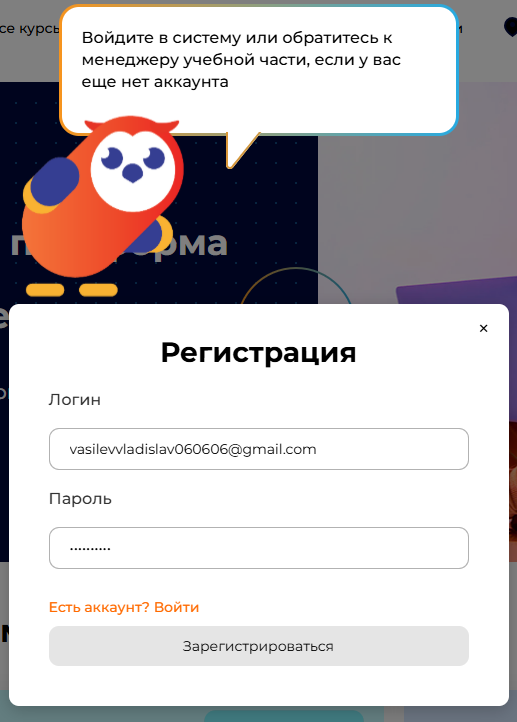


Рисунок 2.3 – Компонент Auth.js(Регистрация)

Для уведомлений и обратной связи реализованы модальные окна, которые отображают важные сообщения, такие как подтверждение успешной отправки данных. Все элементы интерфейса выполнены в минималистичном стиле, что обеспечивает визуальную легкость и акцентирует внимание на главной информации.

Также было создан компонент Application, предоставляет пользователю форму для оставления заявки на участие в выбранном мероприятии(Рисунок 2.4). В данной форме пользователь может указать своё имя, номер телефона, e-mail адрес и сообщение, которое вы хотели бы приложить к заявке

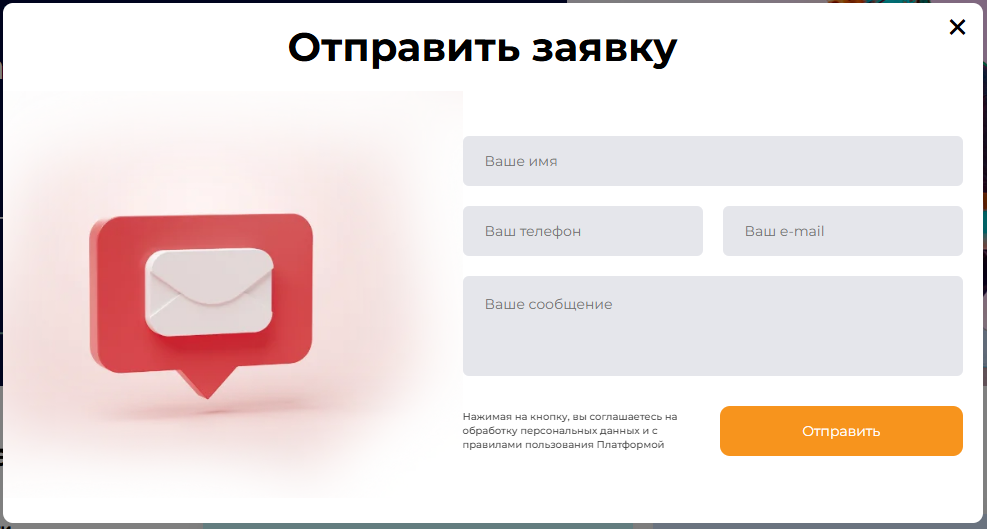


Рисунок 2.4 – Компонент Application

Если же у вас остались вопросы, или вы затрудняетесь в выборе того или иного мероприятия, в конце страницы расположен компонент FeedBack, представляющий форму для обратной связи (Рисунок 2.5).

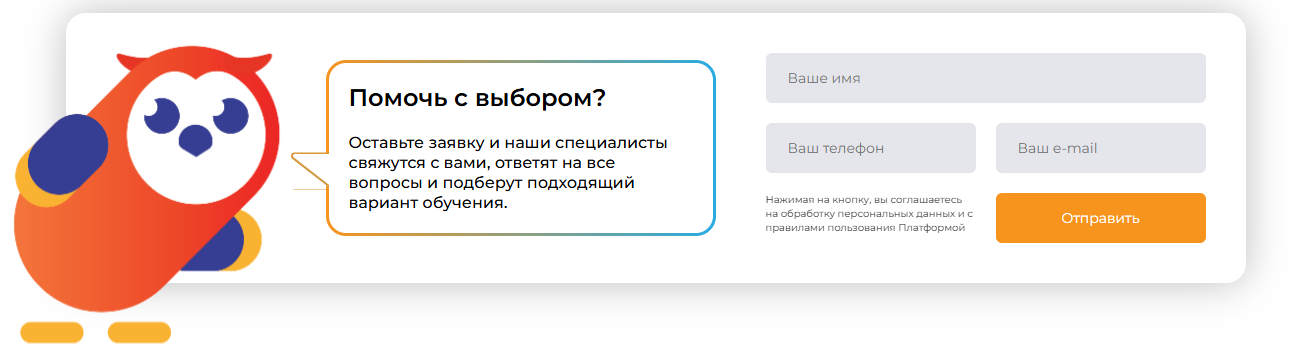


Рисунок 2.5 – Компонент FeedBack

После отправки этой формы, Администратор получает ещё в виде текстовых данных и организует телефонный звонок или оправляет электронное письмо на почту, которая была указана в данной форме.

Также на сайте был применён слайдер изображений (Рисунок 2.6). В качестве слайдов выступают блоки с отзывами пользователей, которые уже воспользовались данным электронным ресурсом.

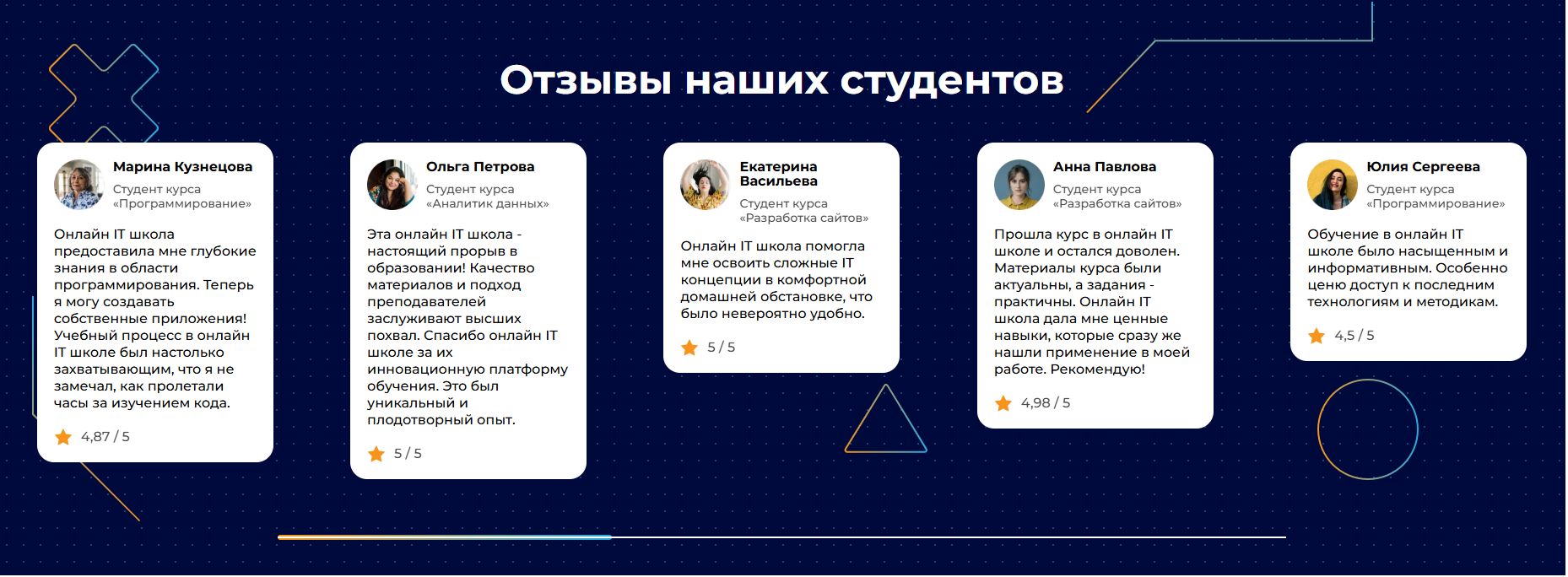


Рисунок 2.6 – Слайдер отзывов

Для реализации слайдера на React с использованием Swiper.js была установлена библиотека Swiper.js через npm и подключены её стили в проект. Настроен базовый функционал, включая указание количества отображаемых слайдов и добавление отступов между ними. Внутри слайдов размещён необходимый контент. Для удобства навигации добавлены стрелки переключения и пагинация в виде точек. Слайдер адаптирован под разные размеры экрана, чтобы на маленьких экранах отображался один слайд, а на больших — несколько. Для повышения производительности настроена ленивая загрузка изображений. Также выполнена кастомизация элементов управления, таких как стрелки и точки, чтобы они соответствовали дизайну проекта.

## **2.4. Выводы**

На этапе проектирования страниц веб-сайта **«Future Academy»** были приняты ключевые решения, направленные на создание современного и удобного интерфейса, адаптированного для различных устройств. Для реализации вёрстки выбран подход с использованием Flexbox и CSS Grid, что обеспечивает гибкость в компоновке элементов и удобство адаптации под экраны любого размера. Минималистичная цветовая схема, состоящая из тёмно-синего, оранжевого и белого цветов, создаёт профессиональный и современный вид, а шрифт Montserrat обеспечивает высокую читаемость текста. При разработке пользовательских элементов, таких как навигационное меню, формы регистрации, авторизации, и модальные окна, было уделено особое внимание интерактивности и удобству взаимодействия. Все эти решения позволили заложить основу для создания функционального веб-сайта, который соответствует современным требованиям и ожиданиям пользователей.

# **Реализация структуры веб-сайта.**

## **3.1 Компонент App.js**

Разработка структуры веб-сайта «Future Academy» была выполнена с использованием фреймворка React, который позволяет строить приложение на основе компонентов. Такой подход обеспечивает удобство разработки, тестирования и дальнейшего расширения функциональности.

Каждая страница сайта реализована как отдельный React-компонент, что позволяет легко управлять её содержимым и добавлять новый функционал. Основной компонент App.js (Листинг 3.1) отвечает за маршрутизацию и объединяет все страницы, используя библиотеку React Router. Это позволяет организовать переходы между страницами, такими как Главная, Курсы, Регистрация, Авторизация и Личный кабинет, без перезагрузки страницы.

import Header from './components/Header/Header.js';

import { BrowserRouter as Router, Route, Routes, Navigate } from 'react-router-dom';

import { YMaps } from 'react-yandex-map';

import Footer from './components/Footer/Footer.js';

import About from './components/About/About.js';

import AuthPage from './components/AuthPage/AuthPage.js';

import Contacts from './components/Contacts/Contacts.js';

import Home from './components/Home/Home.js';

import './App.css';

import { useState } from 'react';

function App() {

    const [userName, setUserName] = useState(null);

    const handleLoginSuccess = (name) => {

        setUserName(name);

    }

    const handleLogout = () => {

        setUserName(null);

    }

    return (

        <YMaps>

            <Router>

                <div className='box'>

                    <Header userName={userName} onLogout={handleLogout} />

                    <Routes>

                        <Route path="/" element={<Home />} />

                        <Route path="/home" element={<Home/>} />

                        <Route path="/auth" element={<AuthPage onLoginSuccess={handleLoginSuccess} />} />

                        <Route path='/contacts' element={<Contacts />} />

                        <Route path='/about' element={<About/>}/>

                    </Routes>

                    <Footer />

                </div>

            </Router>

        </YMaps>

    );

}

export default App;

Листинг 3.1 ­­– Компонент App.js

Также в каждый компонент мы импортируем файл со стилями, которые отвечают за стилизацию конкретного компонента:

import './Header.css';

Листинг 3.2 – Импорт стилей в компонент Header.js

## **3.2 Диаграмма использования**

Построение диаграммы использований является важной деталью в процессе создания веб-приложения, так как оно позволяет визуализировать ключевые функции системы и взаимодействие пользователей с ними. Диаграмма использования этого курсового проекта представлена на Рисунке 3.1.

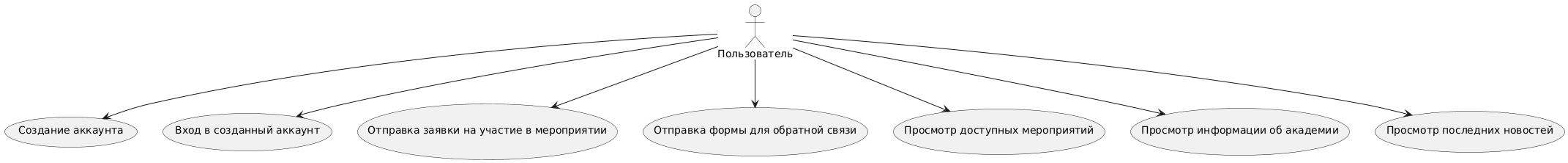


Рисунок 2.6 – Слайдер отзывов

Это помогает разработчикам четко определить, какие сценарии работы необходимо реализовать, установить приоритеты функций и убедиться, что система соответствует требованиям заказчика. Диаграмма также служит основой для дальнейшего проектирования и разработки, обеспечивая ясность и структурированный подход к созданию приложения.

**Выводы**

В результате реализации структуры веб-сайта **«Future Academy»** был создан удобный и функциональный интерфейс, который обеспечивает пользователям доступ ко всем основным страницам платформы, включая Главную, О нас, Регистрацию, Авторизацию и Контакты. Использование компонентного подхода в React позволило эффективно разделить функциональность на логические блоки, что значительно упрощает процесс разработки и дальнейшего масштабирования проекта.

# **Тестирование веб-сайта**

4.1 Адаптивный дизайн веб-сайта

Для достижения адаптивности, помимо использования flexbox-контейнеров, использовались медиа-запросы.

Если сайт адаптивный, то он должен корректно отображаться на экранах различных устройств: мобильные устройства (рисунок 4.1), планшетные устройства (рисунок 4.2), компьютер (рисунок 4.3).

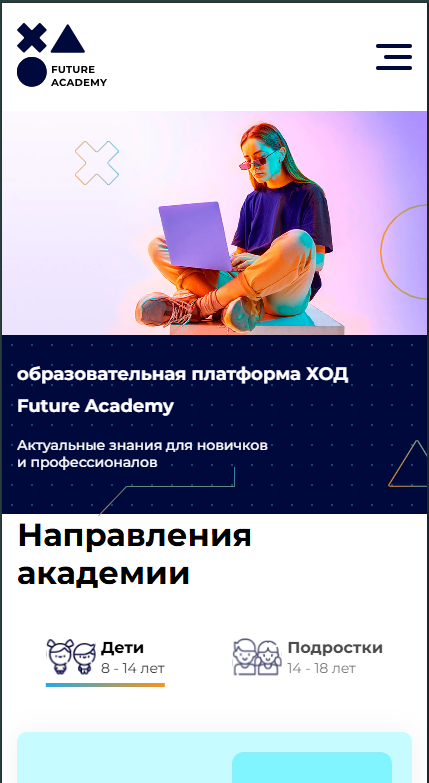


Рисунок 4.1 – Вид сайта для мобильных устройств

Данный пример показывает нам вид веб-сайта, который открыли на мобильном устройстве.

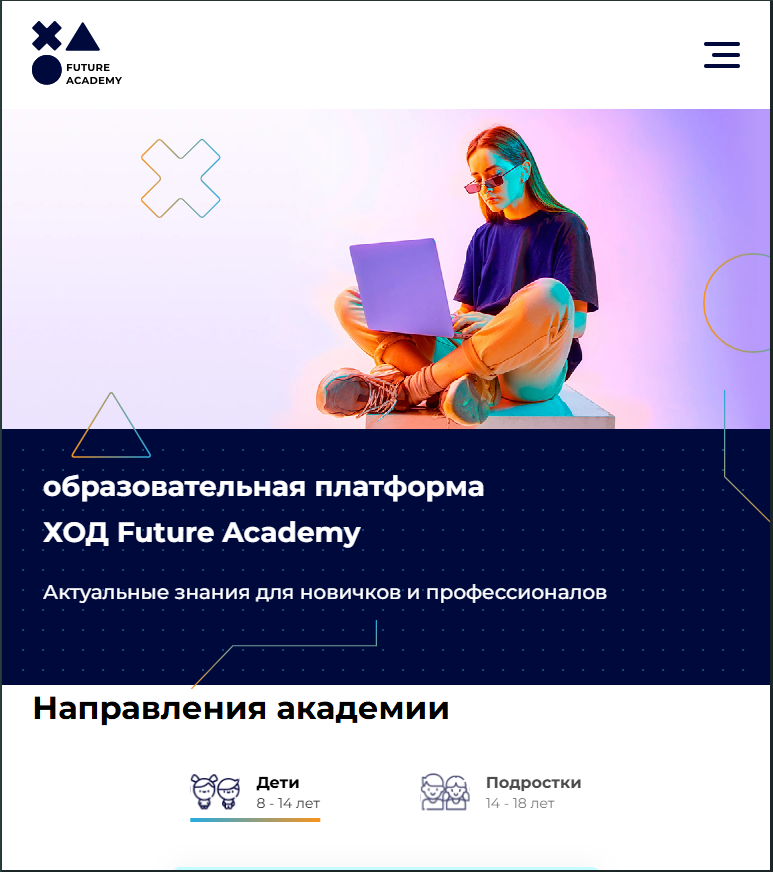


Рисунок 4.2 – Вид сайта с планшетных устройств

Данный пример показывает нам вид веб-сайта, который открыли на планшетном устройстве.

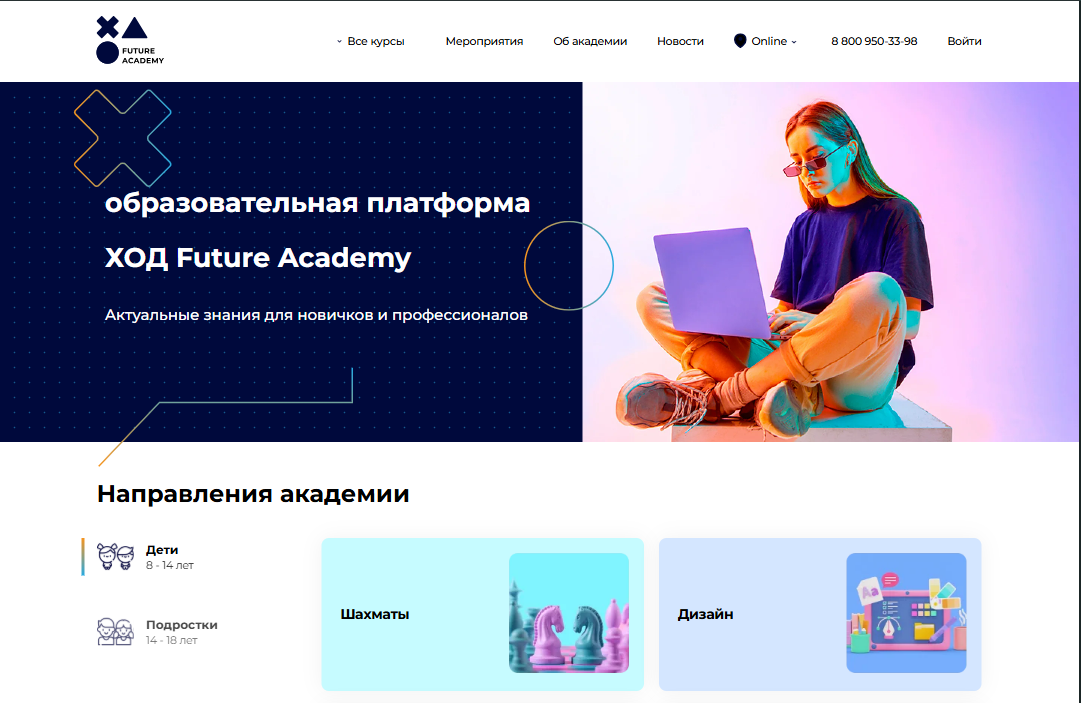


Рисунок 4.3 – Вид сайта с компьютера

Данный пример показывает нам вид веб-сайта, который открыли на компьютере, либо ноутбуке.

Медиа-запросы в основном использовались для сокрытия элементов, которые перекрывают другие элементы и особо не являются обязательными для отображения на странице.

Адаптивный дизайн веб-сайта позволяет ему корректно отображаться и работать на различных устройствах и экранах, таких как компьютеры, планшеты и смартфоны. Вот несколько причин, почему адаптивность важна для сайта:

1. Пользовательский опыт (UX): Адаптивный дизайн обеспечивает удобство использования сайта для всех пользователей, независимо от того, какое устройство они используют. Это создает позитивное впечатление и повышает вероятность того, что пользователь останется на сайте.
2. SEO-оптимизация: Поисковые системы, такие как Google, предпочитают сайты с адаптивным дизайном, поскольку такие сайты обеспечивают более удобный доступ к контенту для пользователей, что положительно влияет на показатели SEO.
3. Доступность: Адаптивный дизайн делает сайт доступным для всех категорий пользователей, включая людей с ограниченными возможностями, так как он учитывает различные способы взаимодействия с сайтом.
4. Повышение конверсии: Хорошо спроектированный адаптивный сайт может увеличить конверсию, поскольку улучшает удобство использования и повышает вероятность выполнения целевых действий, таких как покупка товара или заполнение формы.
5. Экономия времени и ресурсов: Вместо создания отдельных версий сайта для разных устройств, адаптивный дизайн позволяет оптимизировать процесс разработки и обслуживания, что экономит время и ресурсы.

# **Заключение**

В процессе создания веб-сайта для образовательной платформы «Future Academy» была выполнена значительная работа, направленная на разработку современного и функционального цифрового продукта, отвечающего высоким требованиям пользователей. Основной целью являлось создание платформы, которая предоставляет удобный доступ к образовательным материалам, возможность регистрации, авторизации, управления личными данными и отслеживания прогресса. В ходе выполнения работы была спроектирована структура сайта, включающая ключевые страницы, такие как Главная, Курсы, Регистрация, Авторизация и Личный кабинет.

Для реализации интерфейса использовались современные технологии, включая React, HTML5 и CSS3, что позволило достичь высокого уровня интерактивности и удобства пользовательского опыта. Дополнительное внимание было уделено обеспечению адаптивности веб-сайта, что гарантировало корректное отображение и функциональность на различных устройствах, таких как персональные компьютеры, планшеты и смартфоны. Особое внимание уделялось тестированию функциональности, кросс-браузерной совместимости и безопасности, что сделало сайт устойчивым к возможным сбоям и надежным для пользователей.

Использование REST API обеспечило эффективное взаимодействие между клиентской и серверной частью приложения, а современные подходы к разработке способствовали созданию надежного и производительного решения. В результате проведенной работы был создан веб-сайт, который полностью соответствует задачам проекта, предоставляя пользователям не только доступ к образовательным материалам, но и возможность эффективно взаимодействовать с платформой, развивать свои навыки и получать новые знания.

Таким образом, итоговая реализация подтверждает, что образовательные платформы являются важным инструментом в условиях цифровой трансформации общества, а грамотный подход к разработке позволяет создавать решения, которые способствуют развитию непрерывного образования и переквалификации. Веб-сайт «Future Academy» является примером успешного применения современных технологий для удовлетворения образовательных потребностей широкого круга пользователей.