### Введение

Цель проекта - Создание эффективного онлайн-магазина для продажи гироскутеров, который обеспечит простой и удобный интерфейс для покупателей.

Веб приложение должно содержать следующий функционал:

- Авторизация и регистрация: возможность регистрации новых пользователей и авторизации уже зарегистрированных пользователей для просмотра истории заказов.

- Оформление заказа: форма заказа, которая позволяет покупателям указать адрес доставки, способ оплаты и другие дополнительные параметры заказа.

- Обратная связь: возможность отправки запросов и вопросов клиентской службе, отзывов и рейтингов товаров.

- Каталог товаров: каталог товаров, включающий информацию о каждом продукте, такую как наименование, фотографии, описание.

### Основная часть

#### **Технологическая задача**

Разработка системы управления содержимым: позволяет администраторам сайта легко добавлять и редактировать информацию на сайте, такую как описание товаров, новости и другой контент.

Разработка функциональности: разработка всех функциональных возможностей, необходимых для полноценного функционирования интернет-магазина, таких как корзина покупок, авторизация и регистрация.

Разработка дизайна: создание дизайна сайта, который соответствует бренду и обеспечивает простой и интуитивно понятный интерфейс для покупателей.

Поддержка и развитие: поддержка интернет-магазина и его дальнейшее развитие, включая управление заказами, внесение изменений в функциональность и дизайн сайта.

#### **Стек технологий**

Язык: JavaScript

JavaScript является одним из самых популярных языков программирования для создания сайтов и веб-приложений.

Преимущества использования JavaScript:

* Интерактивность JavaScript:
* Улучшенный пользовательский опыт:
* Валидация форм:
* Асинхронное программирование:
* Расширяемость:

Основная библиотека: React

React - это библиотека JavaScript, которая используется для создания пользовательского интерфейса (UI) веб-приложений. Одной из основных особенностей React является его компонентный подход к созданию пользовательского интерфейса.

При создании сайта имеет несколько преимуществ:

* Компонентный подход:
* Оптимизация производительности:
* Управление состоянием:
* Большое сообщество разработчиков:
* Легкость в использовании:

Основная база данных Firebase

Firebase - это платформа для разработки мобильных и веб-приложений, которая была приобретена компанией Google в 2014 году.

Была использована для создания сайта по нескольким причинам:

* Реальное время: Firebase Realtime Database обеспечивает мгновенную синхронизацию данных между клиентом и сервером.
* Аутентификация: Firebase Authentication предоставляет простой и надежный способ аутентификации пользователей.
* Хостинг: Firebase Hosting обеспечивает быстрый и надежный хостинг веб-приложений на серверах Google.
* Простота использования: Firebase предоставляет простой и интуитивно понятный интерфейс управления базой данных и другими сервисами.

**Технологическая задача**

В рамках проекта было несколько технологических задач, которые решались для достижения общей цели создания функционального и успешно работающего веб-сайта.

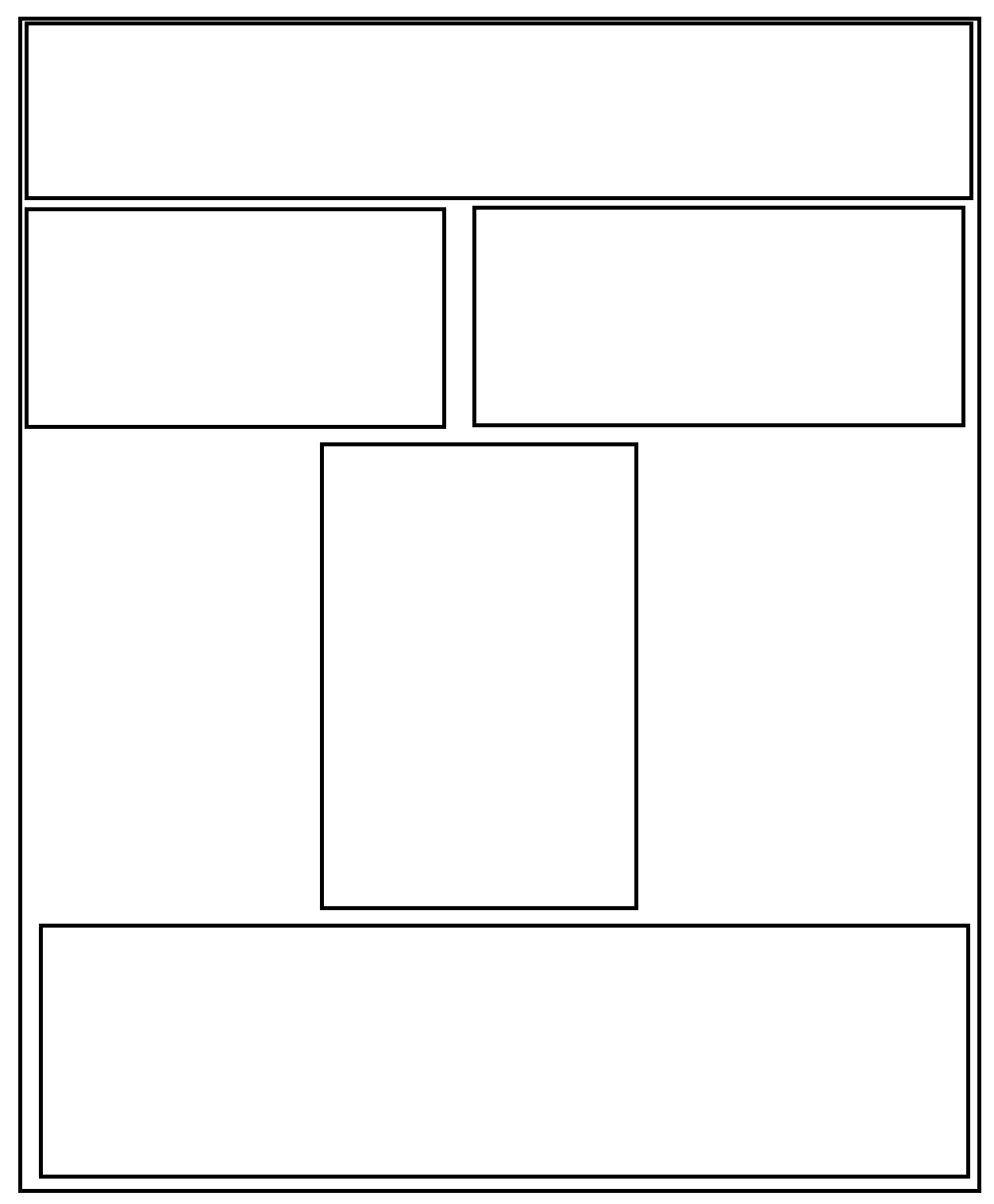
Некоторые из этих задач включают:

1. Разработка и дизайн пользовательского интерфейса: Это включало создание понятного дизайна сайта с удобной навигацией, категориями товаров, фильтрами и другими важными элементами пользовательского интерфейса.
2. Разработка функционала регистрации и авторизации: Это включало создание механизмов, позволяющих пользователям создавать учетные записи, входить в систему, восстанавливать пароли и обеспечивать безопасность данных пользователей.
3. Интеграция с базой данных: Использование базы данных (на примере Firebase) позволило сохранять и управлять информацией о пользователях, товарах, заказах и других сущностях, необходимых для функционирования интернет-магазина.

**Описание архитектуры:**

Архитектура проекта интернет-магазина и базы данных выглядит следующим образом:

* Компоненты React:
  + Login: Компонент для авторизации пользователей.
  + Catalog: Компонент для отображения каталога продуктов и фильтрации.
  + Header: Компонент для отображения заголовка и основной навигации сайта.
* Firebase:
  + Аутентификация: Используется для регистрации и авторизации пользователей.
  + Firestore: База данных, где хранятся данные о продуктах и пользовательских аккаунтах.
  + Storage: Хранилище для загрузки и хранения изображений продуктов.



Изображения логотип партнеров, или брендов

Слайды с изображениями или информацией о продуктах или услугах

Главная страница

Заголовок

Логотип и название приложения

Контактная информация, такая как телефон

Кнопка для открытия модального окна с формой заказа

Контактная информация, адрес и карта с указанием расположения

Модальное окно для заполнения формы заказа

Форма для регистрации нового пользователя

Форма для входа в систему

Дополнительный текстовый контент, например, новости или статьи

Форма для подписки на рассылку новостей

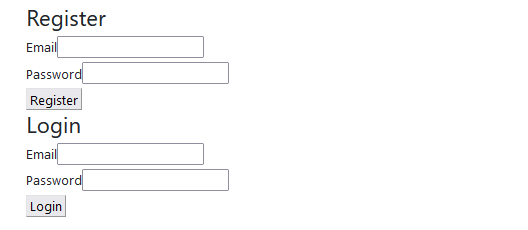
Список продуктов или услуг с возможностью фильтрации и сортировки

Информация о компании или о проекте

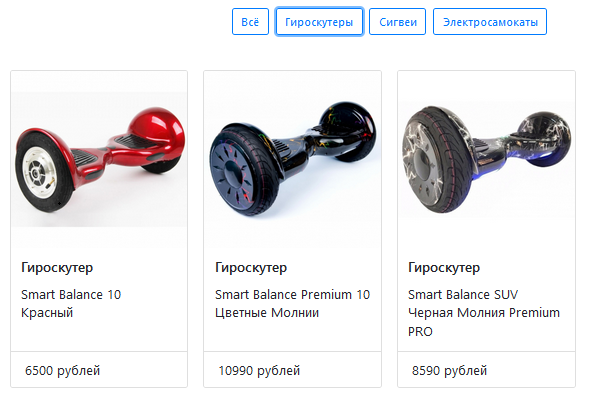
Информация о консультантах или команде проекта

**Инструкция для пользователя:**

* Регистрация и авторизация: Пользователь может зарегистрироваться на сайте, используя форму регистрации. Затем он может авторизоваться, вводя свой email и пароль.



* Просмотр каталога: Пользователь может просматривать список доступных гироскутеров в каталоге. Можно использовать фильтры для отображения продуктов определенного типа.



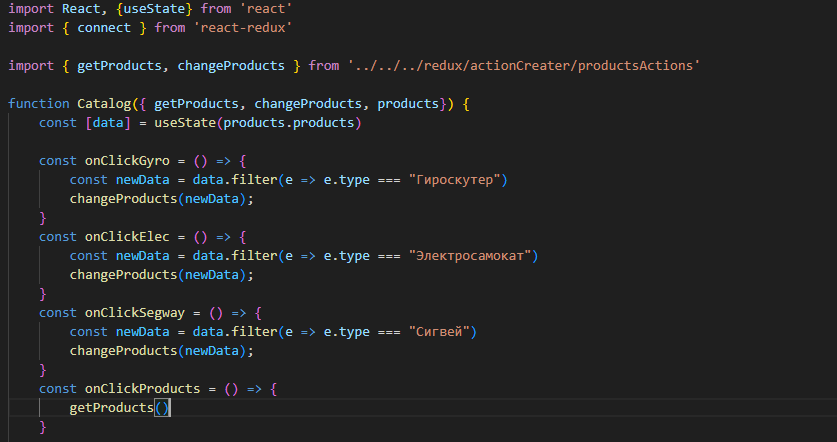


Данный код представляет компонент Header, который отображает верхнюю часть интерфейса сайта, включающую логотип, контактный номер телефона и кнопку для совершения заказа.

Компонент Header возвращает JSX-разметку, которая состоит из div с классом "bg-primary" и nav с классом "container navbar navbar-light justify-content-between". Внутри nav определены следующие элементы:

1. <a> с классом "navbar-brand" - ссылка на главную страницу сайта. Внутри ссылки располагается изображение логотипа, которое импортируется из файла logo-white.png.
2. <a> с классом "phone" - ссылка на номер телефона, который отображается с изображением телефона. Изображение телефона импортируется из файла phone.png.
3. <button> - кнопка для совершения заказа. Кнопка имеет классы "btn btn-outline-light btn-lg" и использует атрибуты data-toggle и data-target для вызова модального окна при нажатии. Текст на кнопке - "Сделать заказ".

Возвращаемый компонент Header предоставляет основные элементы верхней части интерфейса сайта, включая логотип, контактный номер телефона и кнопку для совершения заказа. Этот компонент может быть использован в других компонентах или страницах сайта для создания единообразного и функционального заголовка.

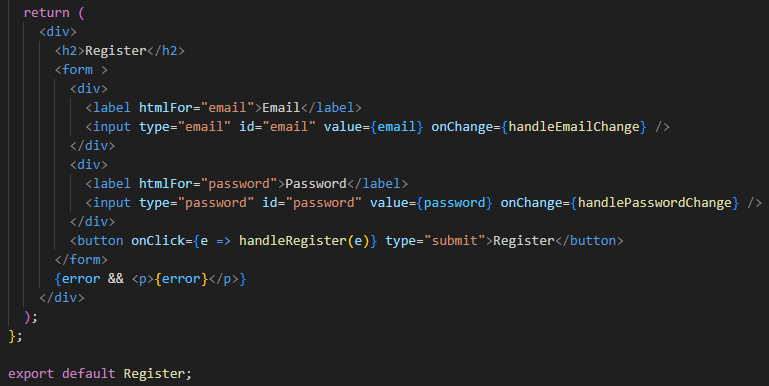
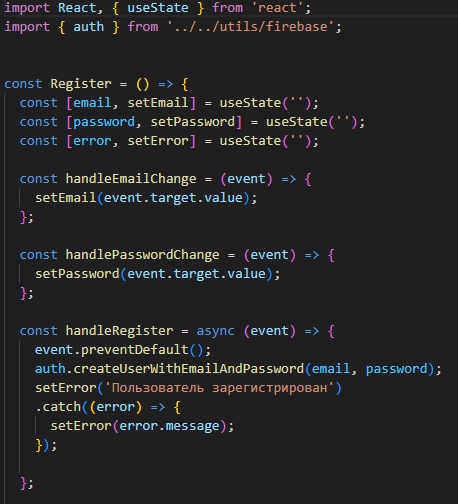


Данный код представляет компонент Catalog, который отображает список продуктов в интернет-магазине и позволяет пользователю фильтровать продукты по категориям.

Компонент Catalog получает список продуктов из хранилища Redux через свойство products. Затем он использует хук useState для создания локального состояния data, которое инициализируется списком продуктов.

В компоненте определены несколько функций-обработчиков событий, таких как onClickGyro, onClickElec, onClickSegway и onClickProducts, которые выполняют фильтрацию списка продуктов по соответствующим категориям. Фильтрованный список передается функции changeProducts для обновления состояния в хранилище Redux.

Общий функционал данного кода заключается в отображении списка продуктов и возможности фильтровать их по категориям. Он обеспечивает пользовательский интерфейс для просмотра и выбора продуктов в интернет-магазине.



Данный код позволяет реализовать функциональность входа в систему (авторизации) пользователей в приложении. Он обеспечивает следующие возможности:

1. Ввод данных пользователя: Пользователь может ввести свой электронный адрес (email) и пароль через соответствующие поля ввода.
2. Обработка входа в систему: При нажатии на кнопку "Login", функция onLogin вызывается для обработки входа пользователя. Внутри этой функции происходит вызов метода signInWithEmailAndPassword(email, password) объекта auth, который предоставляется Firebase. Этот метод выполняет аутентификацию пользователя на основе введенных данных.
3. Обработка успешного входа: Если вход пользователя успешен, то в объекте userCredential содержится информация о пользователе, которая может быть использована для дальнейшей обработки или отображения. В данном случае, код выводит информацию о пользователе в консоль и устанавливает сообщение "Пользователь авторизован" в состояние error.
4. Обработка ошибок входа: Если при выполнении входа возникла ошибка, то в блоке catch перехватывается и обрабатывается ошибка. Код выводит код ошибки (errorCode) и сообщение об ошибке (errorMessage) в консоль и устанавливает сообщение об ошибке в состояние error. Это сообщение затем отображается на странице, чтобы пользователь мог увидеть информацию об ошибке.



Данный код позволяет инициализировать и настроить Firebase в проекте, он открывает доступ к различным сервисам и функциональностям, предоставляемым Firebase. В конкретном случае, код позволяет использовать следующие возможности:

1. Аутентификация пользователей: Firebase предоставляет механизмы аутентификации, которые позволяют пользователю регистрироваться, входить в систему и управлять своим профилем.
2. Работа с базой данных: Firebase предоставляет базу данных Firestore, которая является гибкой и масштабируемой NoSQL базой данных.
3. Хранение файлов: Firebase также предлагает облачное хранилище для загрузки, хранения и обработки файлов.

В моем проекте данный код интегрирован к интернет магазину и пользователи могут регистрироваться и авторизоваться

**Вывод**

В ходе процесса создания проекта были достигнуты некоторые значимые результаты. Удалось успешно реализовать сам сайт и привязать сайт к базе данных Firebase. Это позволило реализовать взаимодействие с базой данных для хранения и извлечения данных, а также обновления и удаления записей, реализован функционал регистрации и авторизации пользователей с использованием базы данных. Поскольку теперь пользователи могут создавать учетные записи, входить в систему и получать доступ к персонализированной информации.

С помощью Firebase SDK для JavaScript и соответствующего API, сайт получил возможность установить соединение с Firebase Realtime Database. Это позволило клиентской части взаимодействовать с базой данных непосредственно из браузера пользователя.

Однако, не удалось реализовать функционал, позволяющий пользователям добавлять собственные гироскутеры на продажу. Не удалось реализовать функциональность подписки на рассылку новостей. Для рассылку новостей обычно требует интеграции с почтовым сервисом или использования сторонних платформ для управления подписчиками и отправки сообщений.

По моим личным выводам разработка проекта интернет-магазина был познавательным. В процессе работы над проектом я столкнулся с различными технологическими задачами и решениями, что позволило мне расширить свои навыки. Одним из ключевых достижений было успешное привязывание сайта к базе данных Firebase. Это позволило создать функциональный интернет-магазин, где клиенты могли просматривать и покупать гироскутеры, администраторы могли управлять продуктами

В процессе работы я также использовал React для создания динамического и отзывчивого пользовательского интерфейса. React позволил мне эффективно организовать компоненты, управлять состоянием приложения и обновлять интерфейс в реальном времени, что было особенно полезно при работе с базой данных Firebase.

Однако, в ходе разработки, не удалось реализовать некоторые дополнительные функциональности которые были указаны выше по причине новой для себя технологией и недостаток времени для более подробного изучения что в итоге приводила к ошибкам.

**Ссылки**

В ходе работы над проектом я использовал следующие ресурсы и документацию:

* Официальная документация React: <https://reactjs.org/>
* Официальная документация Firebase: <https://firebase.google.com/docs>
* "Серия уроков на YouTube" по созданию сайта на React: <https://www.youtube.com/watch?v=jT8wO8aKea0&list=PLVfMKQXDAhGWamOcqqIsMAbsP6pl0vUtW>
* Официальная документация Firebase о работе с базой данных в вебе: <https://firebase.google.com/docs/database/web/start?hl=ru>
* Статья на тему подключения базы данных Firebase Realtime к веб-сайту: <https://themewp.inform.click/kak-podkljuchit-bazu-dannyh-firebase-realtime-k-forme-vashego-veb-sajta/>
* Видеоурок на YouTube о создании интернет-магазина с использованием React и Firebase: <https://www.youtube.com/watch?v=uf_2rDv3jDs>
* Видеоурок на YouTube о создании фильтра продуктов с помощью React и Firebase: https://www.youtube.com/watch?v=ZPW\_JabgPgo