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

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Скриптовые языки программирования»

Тема «Web-сайт «Интернет-магазин одежды»»

**Исполнитель**

студент 2 курса 3 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А.Шманай

подпись, дата

**Руководитель**

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

должность, учен. степень, ученое звание подпись, дата

Допущен к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

дата, подпись

Курсовой проект защищен с оценкой

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

подпись дата инициалы и фамилия

Минск 2024

**Содержание**

[Введение 3](#_1fob9te)

[1 Обзор технических методов и программных средств разработки сайта 4](#_3znysh7)

[1.1 Обоснование необходимости создания программного продукта 4](#_2et92p0)

[1.2 Формирование требований к ПП 5](#_tyjcwt)

[1.3 Выбор средств реализации программного продукта 6](#_3dy6vkm)

[1.4 Вывод 6](#_1t3h5sf)

[2 Последовательность разработки содержания и структуры веб-сайта с использованием компонентов 7](#_4d34og8)

[2.1 Структура ПП 7](#_2s8eyo1)

[2.2 Выбор стилевого оформления 7](#_17dp8vu)

[2.3 Выбор шрифтового оформления 8](#_3rdcrjn)

[2.4 Разработка логотипа 8](#_26in1rg)

[2.5 Разработка пользовательских элементов 9](#_lnxbz9)

[2.6 Вывод 1](#_35nkun2)0

[3 Последовательность разработки содержания и структуры веб-сайта с использованием компонентов](#_1ksv4uv) 11

[3.1 Внедрение в проект каталога 12](#_44sinio)

3.2 Использование таблиц стилей CSS [12](#_2jxsxqh)

[3.3 Вывод 13](#_3j2qqm3)

[4 Описание хода работы программного средства 14](#_1y810tw)

[4.1 Руководство пользователя 14](#_4i7ojhp)

[4.2 Диаграмма вариантов использования веб-приложения 1](#_2xcytpi)6

[4.3 Тестирование ПП 1](#_1ci93xb)7

[4.4 Вывод 1](#_3whwml4)8

[Заключение](#_2bn6wsx) 19

[Список использованных литературных источников 2](#_qsh70q)0

[5 Приложения 2](#_3as4poj)1

**Введение**

Веб-приложения наблюдается широкий спрос на веб-приложения, которые позволяют пользователям удобно взаимодействовать с обновляемым контентом. Такие веб-страницы, содержание которых формируется в зависимости от запросов, называются динамическими.

Особенно актуальны решения, предназначенные для онлайн-магазинов, где покупатели смогут легко приобретать модные товары и обувь, чтобы всегда быть в мире моды. Для таких приложений крайне важно обеспечить кроссбраузерность и кроссплатформенность, чтобы контент корректно отображался независимо от используемого устройства.

Учитывая постоянный спрос на качественные интернет-магазины, целью данного курсового проекта является создание кроссплатформенного и кроссбраузерного веб-приложения для создания и продажи кастомной обуви. В процессе реализации будут применяться современные JavaScript-библиотеки, такие как React и другие инструменты для создания удобного и функционального интерфейса.

Основные задачи проекта включают:

* анализ существующих решений;
* определение особенностей разрабатываемого веб-приложения;
* выбор подходящих технологий и инструментов;
* проектирование интерфейса и разделение его на компоненты;
* разработка пользовательских элементов интерфейса;
* создание фирменного логотипа;
* разработка структуры приложения и подключение таблиц стилей;
* интеграция Redux Toolkit для управления состоянием;
* подготовка руководства пользователя;
* подведение итогов по каждому этапу работы.

**1 Обзор технических методов и программных средств разработки сайта**

**1.1 Обследование объекта и обоснование необходимости создания программного продукта**

В сети Интернет существует множество торговых площадок, предлагающих товары на любой вкус, будь то обувь или техника. В ходе анализа большинства подобных сайтов было выявлено, что контент, как правило, располагается в центральной части страницы, в то время как в верхней или в боковой части будет находиться меню.

Кроме того, многие сайты оснащены динамичным слайдером на главной странице, что делает их более привлекательными для пользователей.

Одним из таких сайтов является VseMayki.ru — известный интернет-магазин одежды, обуви и аксессуаров на территории Беларуси.

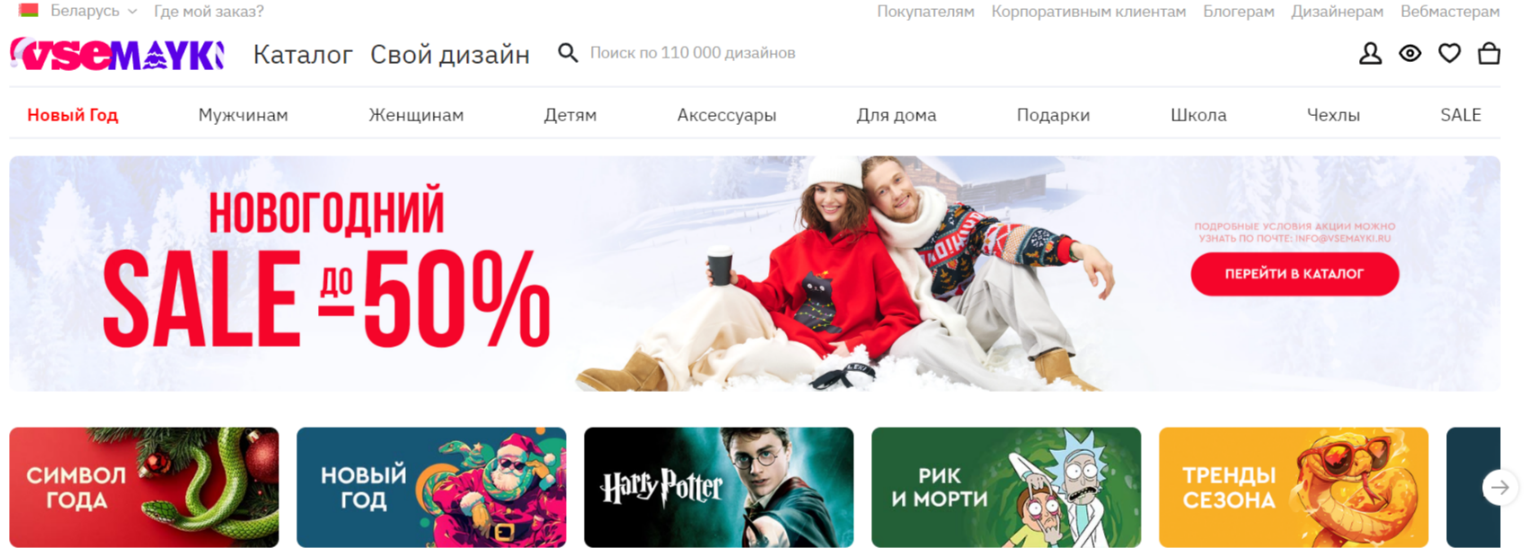


Рисунок 1.1 – главная страница VseMayki.ru, и ее категории

Слайдер, который размещен на главной странице сайта, является эффективным инструментом привлечения внимания пользователей и пробуждает в них дальнейшее желание к изучению каталога. Удобная структура поиска товаров, реализованная на сайте, способствует более эффективному поиску необходимой продукции, что является огромным плюсом.

Примером подобного подхода может послужить популярный сайт Megatop.by, предлагающий огромный выбор обуви на любое время года. Каталог интернет-магазина позволяет пользователям просматривать товары, фильтровать их по различным параметрам и осуществлять поиск по названию.



Рисунок 1.2 – каталог сайта Megatop.by

Каталог этого сайта может послужить хорошим примером для проекта, так как он довольно удобный в использовании, на странице присутствует нумерация, есть список фильтров, и поиск по названию.

**1.2 Формирование требований к ПП**

Проведя анализ доступных примеров, можно сделать вывод, что типичный интернет-магазин обуви должен включать в себя привлекательную, и в то же время максимально удобную главную страницу с приветствием посетителя, каталог с возможностью тонкой настройки фильтров, страницу товара, корзину для оформления заказа и информацию об интернет-магазине в конце страницы.

На главной странице должно быть размещено краткое приветствие для пользователя и предложение перейти в каталог. Необходимо реализовать каталог и корзину с формой для отправки заказа и её валидацией.

Основная функция сайта заключается в предоставлении возможности приобрести понравившийся товар через интернет-сайт для обычного пользователя.

Основные задачи курсового проекта включают в себя разработку дизайна, разделение составляющих сайта на React-компоненты и создание этих компонентов, а также разработку функционала интернет-магазина.

**1.3 Выбор средств реализации программного продукта**

Для реализации программного продукта применяется бесплатный редактор кода Visual Studio Code. При создании веб-приложения используются библиотека React и другие библиотеки, от которых зависят данные библиотеки. Веб-приложение разрабатывается преимущественно на языке JavaScript с использованием JSX — расширения этого языка, позволяющего использовать HTML5 в JavaScript-коде. В качестве стилевого оформления проекта применяются каскадные таблицы стилей CSS3. Для запуска и сборки приложения используются Node.js и его пакетный менеджер npm.

**1.4 Вывод**

В рамках данного раздела были проанализированы аналогичные решения, выявлена их сущность и специфика восприятия целевой аудиторией. Полученные знания были использованы при разработке последующего продукта, обладающего высоким качеством и отвечающего требованиям пользователей. Были определены цели и задачи для создания веб-приложения.

**2 Последовательность разработки содержания и структуры веб-сайта с использованием компонентов**

**2.1 Структура ПП**

В связи с тем, что разрабатываемая страница является React-приложением, то она состоит из множества компонентов. Каждый компонент включает в себя частичку функционала программного продукта. На рисунке 2.1 представлена структура приложения, соответствующая ее компонентам.

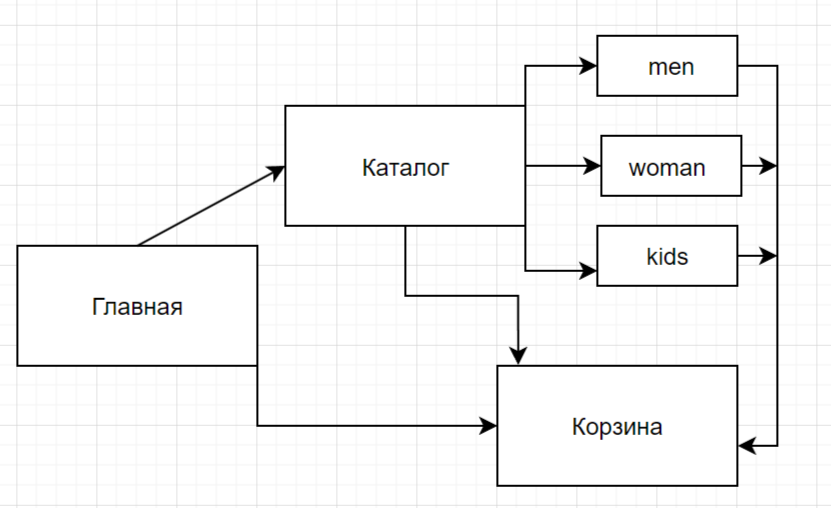


Рисунок 2.1 – структура приложения

На стартовой странице сайта, расположенной в разделе **Главная**, располагается ключевой элемент, предназначенный для привлечения внимания большого количества пользователей. Также здесь можно увидеть крупный заголовок и элемент товара, доступный для изучения в каталоге магазина.

Отсюда пользователь может перейти либо в каталог, либо в корзину. В каталоге интернет-магазина представлены товары, которые поделены на 3 категории. Разделение на несколько видов намного удобнее в том плане, что не придется разбираться где мужская, а где женская пара обуви.

Если выбрать товар в этом элементе, можно перейти на страницу, где представлен элемент **Корзина**.

На странице **Корзина** расположен элемент, представляющий собой форму для оформления заказа. Здесь можно произвести оплату заказа и вернуться в раздел Каталог.

**2.2 Выбор стилевого оформления**

При определении с дизайном и цветовой палитрой сайта были выбраны пастельные тона, хорошо сочетающиеся между собой. Используется лаконичный шрифт, для улучшенной привлекательности сайта. На рисунке 2.2 изображена используемая цветовая палитра сайта.

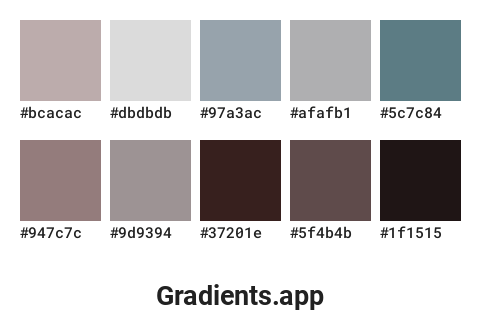


Рисунок 2.2 – Цветовая палитра сайта

Ключевой цвет для оформления текста – бежевый (#947с7с) используется в оформлении фона. #1f1515 используется для выделения ключевых элементов и акцента на сайте. Также он присутствует в заднем фоне иконки сайта. #37201e используется для цветов кнопки по всем компонентам сайта. #5f4b4b используется для рамочек фона текстовых списков, и в некоторых текстах сайта.

**2.3 Выбор шрифтового оформления**

В данном проекте используется один единственный шрифт Impact . Он присутствует во всех надписях сайта, и придает своими очертаниями сайту лаконичность, тем самым делая его привлекательным. Однако в качестве запасного шрифта будет использован шрифт Arial, в случае если по каким-то причинам основной шрифт не будет перегружен. Шрифты поддерживают кириллицу и не конфликтуют друг с другом относительно стилевых решений.

**2.4 Разработка логотипа**

Логотип сайта представляет с собой красиво обрисованное название сайта с кольцами, что означает хорошее качество.



Рисунок 2.3 – логотип интернет-магазина Foskys Boots

Решение сделать логотипа сайта таковым было обусловлено его стилистическим решением: он очень хорошо демонстрирует аккуратность и хорошую выдержку фирмы, что означает хорошая продукция.

**2.5 Разработка пользовательских элементов**

В странице будут представлены такие элементы пользовательского интерфейса как шапка сайта с горизонтальным меню, миниатюра товара в каталоге, форма оформления заказа в корзине.

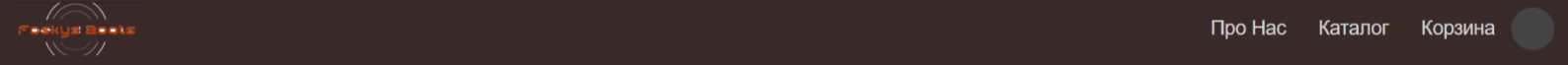


Рисунок 2.4 – шапка сайта и горизонтальное меню

На протяжении всего времени посещения сайта, у пользователя будет отличная возможность воспользоваться меню, состоящим из кнопок, ведущих на различные страницы сайта.

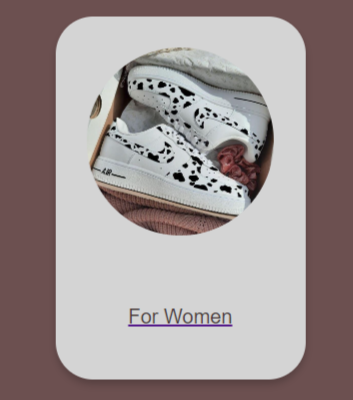


Рисунок 2.5 – миниатюра товара в каталоге

Во время просмотра каталога сайта у пользователя будет возможность познакомиться с списком товаров. Для этого необходимо создать миниатюру, содержащую краткие сведения о товаре (название, итд.), а также изображение самого товара, с возможностью добавить этот самый товар в корзину.

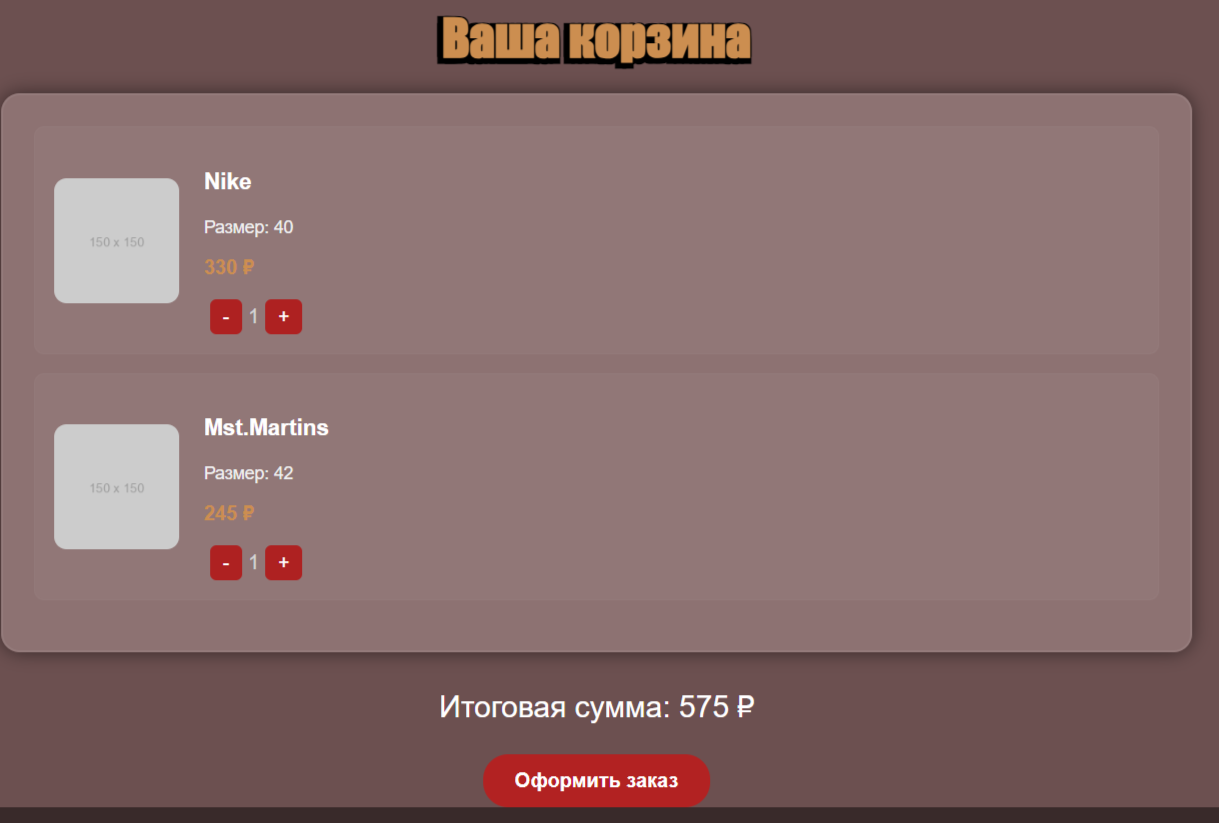


Рисунок 2.6 – форма оформления заказа

Для компонента **Корзина** необходимо разработать компонент, который будет давать возможность оформить заказ. На рисунке 2.6 приведена реализация такой формы.

**2.6 Вывод**

На данном этапе была определена структура компонентов, стилевое оформление, в том числе логотип, шрифты, цветовая палитра. Также были определены ключевые элементы пользовательского интерфейса.

**3 Последовательность разработки содержания и структуры веб-сайта с использованием компонентов**

**3.1 Внедрение в проект каталога**

Приложение разрабатывалось в бесплатном текстовом редакторе Microsoft Visual Studio Code. Так как приложение является React-приложением, то оно разбито на компоненты.

В качестве демонстрации одного из таких компонентов был взят компонент **Корзина** из страницы корзины приложения.

import React, { useState } from "react";

import { Link } from "react-router-dom";

import "./Cart.css";

import logo from "./images/logot.png";

const shoesData = [

  { id: 1, name: "Nike", category: "Кроссовки", size: 40, price: 330, quantity: 1, image: "https://via.placeholder.com/150" },

  { id: 2, name: "Mst.Martins", category: "Ботинки", size: 42, price: 245, quantity: 1, image: "https://via.placeholder.com/150" },

];

const Cart = () => {

  const [items, setItems] = useState(shoesData);

  const [dropdownVisible, setDropdownVisible] = useState(false);

  const increaseQuantity = (id) => {

    setItems(items.map((item) => (item.id === id ? { ...item, quantity: item.quantity + 1 } : item)));

  };

  const decreaseQuantity = (id) => {

    setItems(

      items

        .map((item) => item.id === id ? { ...item, quantity: item.quantity > 1 ? item.quantity - 1 : 0 } : item)

        .filter((item) => item.quantity > 0)

    );

  };

  const totalPrice = items.reduce((acc, item) => acc + item.price \* item.quantity, 0);

  const handleCheckout = () => { alert("Ваш заказ успешно оформлен! 🎉 Спасибо за покупку!");

  };

  …

Листинг 3.1 – Фрагмент компонента Корзина

Для создания компонента `Корзина` потребовалось использование таких React-хуков как `useState` и `useEffect`. Этот компонент представляет собой форму корзины, где пользователи могут вводить свои пожелания в покупке и оформлять заказы. В целом, данный функциональный компонент обеспечивает пользователю возможность ввода и проверки данных перед отправкой формы.

React Hooks:

- `useState` для управления состоянием формы.

- `useEffect` для добавления и удаления обработчика событий на поле ввода телефона.

- Валидация ввода: Функции `validateEmail` для проверки корректности введенных данных.

**3.2 Использование таблиц стилей CSS**

.header {

    display: flex;

    justify-content: space-between;

    align-items: center;

    background-color: rgba(57, 42, 42, 1);

    padding: 10px 20px;

}

.logo {

    width: 120px;

    cursor: pointer;

}

.nav {

    display: flex;

    align-items: center;

}

.nav-link {

    margin: 0 15px;

    text-decoration: none;

    color: #d4d4d4;

    cursor: pointer;

    transition: transform 0.2s, border-bottom 0.2s;

}

.nav-link:hover {

    transform: scale(1.1);

    border-bottom: 2px solid #b22222;

}

.account-icon {

    width: 40px;

    height: 40px;

    background-color: #444;

    border-radius: 50%;

    cursor: pointer;

}

Листинг 3.3 – фрагмент AboutUs.css

В приложении был использован стандартный CSS3 для написания стилей компонентам приложения. В листинге приведена реализация стиля для навигационного меню в шапке сайта, с указанием атрибутов, придающих ему плавность и красоту.

**3.3 Вывод**

В данном разделе была продемонстрирована часть стека технологий, использованного при разработке программного продукта. Были продемонстрированы листинги кода приложения.

**4 Описание хода работы программного средства**

**4.1 Руководство пользователя**

Когда пользователь заходит в интернет-магазин, его встречает главная страница магазина, которая содержит в себе приветствие с зашедшим человеком на сайт, а также большой слайдер с тематическими картинками, предлагающими перейти к каталогу сайта.



Рисунок 4.1 – Главная страница интернет-магазина

На странице каталога пользователь может просмотреть каталог товаров, найти товар по его имени, отсортировать по цене. Также пользователь может добавить нужный ему товар в корзину прямо из каталога путем нажатия соответствующей кнопки.

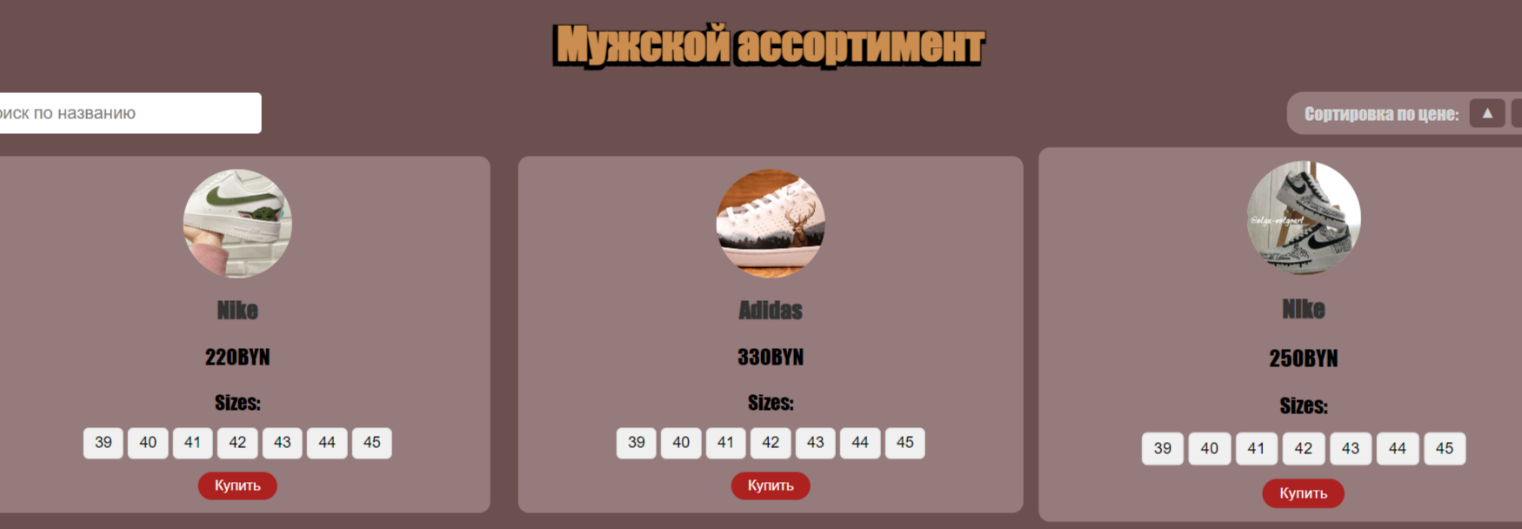


Рисунок 4.2 – Каталог интернет-магазина

При нажатии на миниатюру товара, пользователь попадает на страницу корзины, где представляется возможность оформить заказ.

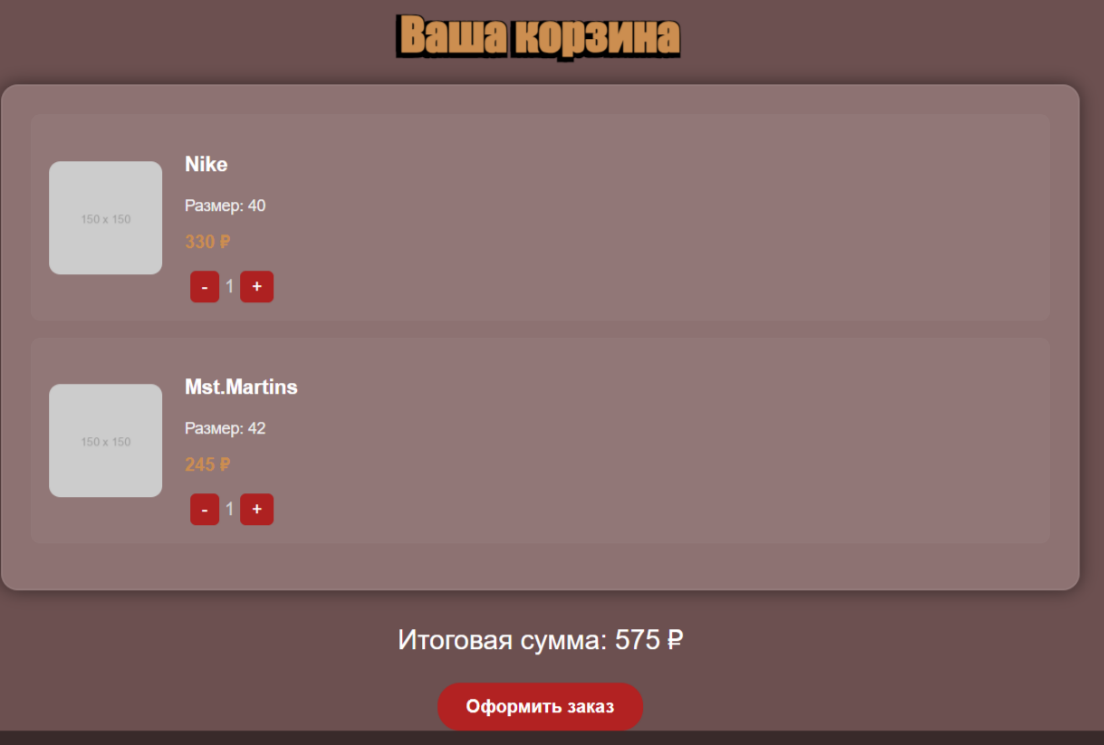


Рисунок 4.3 – Страница корзины

На странице корзины, пользователю предлагают выбрать количество пар обуви и автоматический подсчёт итоговой суммы. Подразумевается, что способ оплаты указан у пользователя в личном кабинете.

**4.2 Диаграмма вариантов использования веб-приложения**



Рисунок 4.4 – Диаграмма вариантов использования интернет-магазина

Диаграмма вариантов использования наглядно демонстрирует взаимодействие пользователя с функциональными возможностями приложения. На ней представлен главный актер: Пользователь. Он представляет собой посетителя сайта, который обладает своим набором действий. К основным действиям Пользователя относятся:

* Просмотр главной страницы: начальная точка взаимодействия с сайтом, где пользователь может ознакомиться с основной информацией и перейти к интересующим разделам.
* Просмотр формы для интересующих вопросов, демонстрирует дальнейшие обновления платформы, под разных пользователей.
* Просмотр каталога: предоставляет доступ к ассортименту.
* Просмотр страницы каталога: позволяет изучить подробное описание конкретной интересующей пользователя одежды.
* Добавление товара в корзину: при выборе подходящего товара пользователь может добавить его в корзину для последующего оформления заказа.
* Покупка: завершение процесса, при котором пользователь оформляет заказ на выбранные товары.

Диаграмма подчёркивает простоту и интуитивность использования интернет-магазина. Все действия связаны логической последовательностью, что делает процесс выбора и покупки обуви удобным для пользователя. В результате система предоставляет функциональность, необходимую для комфортного взаимодействия, независимо от уровня технической подготовки клиента.

**4.3 Тестирование ПП**

Кроссбраузерность – это свойство веб-страницы корректно отображаться и функционировать во всех популярных браузерах, без ошибок в верстке и с одинаково читаемым контентом.

Для проверки реализованной кроссбраузерности веб-приложения были протестированы два браузера, работающие на различных движках рендеринга: GoogleChrome и Microsoft Edge. На рисунках 4.5 и 4.6 продемонстрированы примеры корректного отображения и функционирования сайта в этих браузерах.



Рисунок 4.5 – Работа в браузере GoogleChrome



Рисунок 4.6 – Работа в браузере MicroSoft Edge

**4.4 Вывод**

В этом разделе подробно представлено руководство по использованию веб-приложения, которое помогает пользователям лучше понимать функционал и все возможности интернет-магазина кастомной обуви. Были рассмотрены основные шаги и действия, необходимые для комфортного взаимодействия с сайтом, что позволяет пользователям легко освоить работу с приложением, независимо от их уровня технической подготовки.

В рамках этого этапа разработки проведено всестороннее тестирование приложения на совместимость с различными браузерами. Для проверки были использованы наиболее популярные современные браузеры, такие как Google Chrome, MicroSoft Edge. Результаты тестирования подтвердили стабильную и корректную работу сайта, что полностью соответствует требованиям кроссбраузерности. Это гарантирует, что все пользователи, смогут без затруднений взаимодействовать с веб-приложением.

Для лучшего понимания возможностей платформы была представлена диаграмма вариантов использования, визуализирующая действия, доступные пользователям. Диаграмма отражает функционал пользователя. Пользователи могут просматривать каталог автозапчастей, информацию о товарах, добавлять товары в корзину, управлять содержимым корзины и оформлять покупки. Этот графический инструмент наглядно демонстрирует, как пользователи взаимодействуют с системой, подчёркивая различия в доступных функциях в зависимости от их статуса.

**Заключение**

Целью данной курсовой работы было создание одностраничного веб-приложения на тему «Интернет-магазин кастомной обуви».

В рамках проекта был проведён анализ существующих решений, что позволило выявить ключевые тренды в дизайне и функционале современных веб-приложений. На этапе формирования технического задания были детально продуманы основные страницы сайта и их содержание.

Для разработки приложения были выбраны современные наборы инструментов, обеспечивающий эффективную реализацию поставленных задач. В ходе проектирования была разработана структура приложения, а также представлены примеры реализации некоторых функций. Особое внимание уделялось созданию пользовательского интерфейса и фирменного логотипа, что придало веб-сайту уникальность и повысило его узнаваемость.

Пользователям было предоставлено руководство по использованию интернет-магазина, чтобы сделать процесс работы с сайтом максимально удобным. На этапе тестирования было подтверждено, что приложение корректно работает во всех популярных браузерах, что гарантирует его кроссбраузерность.

В процессе разработки использовались передовые технологии, такие как Redux Toolkit, что позволило выйти за рамки базовых требований курса и внедрить современные подходы к веб-разработке.

В результате был создан кроссбраузерный интернет-магазин кастомной обуви, включающий уникальные элементы интерфейса и высокую функциональность. Приложение демонстрирует надёжность, и интегрирует современные технологии для обеспечения удобства пользователей.

**Список использованных литературных источников**

1. Сайт Mark Formrlle.by [Электронный ресурс]. Режим доступа: <https://detali.by/>– Дата доступа: 10.09.2024
2. Сайт Zara.by [Электронный ресурс]. Режим доступа: [https://autoostrov.by/](https://autoostrov.by/%20) – Дата доступа: 10.09.2024
3. Документация HTML5 [Электронный ресурс]. Режим доступа: [https://metanit.com/web/html5/](https://metanit.com/web/html5/%20) – Дата доступа: 6.10.2024
4. Документация CSS3[Электронный ресурс]. Режим доступа: [https://htmlbook.ru/css](https://htmlbook.ru/css%20) – Дата доступа: 10.10.2024
5. Документация React [Электронный ресурс]. Режим доступа: [https://ru.legacy.reactjs.org/docs/getting-started.html](https://ru.legacy.reactjs.org/docs/getting-started.html%20) – Дата доступа: 15.11.2024
6. Документация Redux [Электронный ресурс]. Режим доступа: [https://ru.react-redux.js.org/introduction/getting-started/](https://ru.react-redux.js.org/introduction/getting-started/%20) – Дата доступа: 20.11.2024

**5 Приложения**

import React from "react";

import ReactDOM from "react-dom";

import { BrowserRouter, Routes, Route } from "react-router-dom";

import App from "./App";

import AboutUs from "./AboutUs";

import MensShoes from "./Catalog";

import Cart from "./Cart";

import Woman from "./Woman";

import Kids from "./Kids";

import Auth from "./Auth";

ReactDOM.render(

  <BrowserRouter>

    <Routes>

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

      <Route path="/about" element={<AboutUs />} />

      <Route path="/mens-shoes" element={<MensShoes />} />

      <Route path="/cart" element={<Cart />} />

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

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

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

    </Routes>

  </BrowserRouter>,

  document.getElementById("root")

);

Листинг index.js

import React, { useState } from "react";

import { Link } from "react-router-dom";

import "./AboutUs.css";

import logo from "./images/logot.png";

import foto1 from "./images/company.jpg";

import foto2 from "./images/sasha.jpg";

import foto3 from "./images/Motisha.jpg";

import foto4 from "./images/nadya.jpg";

import foto5 from "./images/kosya.jpg";

import foto6 from "./images/vika.jpg";

import foto7 from "./images/ivan.jpg";

const AboutUs = () => {

  const [dropdownVisible, setDropdownVisible] = useState(false);

  return (

    <div className="about-us">

      <header className="header">

        <Link to="/">

          <img src={logo} alt="Foskys Boots Logo" className="logo" />

        </Link>

        <nav className="nav">

          <Link to="/" className="nav-link">Главная</Link>

            <div className="nav-link" onClick={() => setDropdownVisible(!dropdownVisible)}>

            Каталог

            <div className={`dropdown-menu ${dropdownVisible ? 'visible' : ''}`}>

          <Link to="/mens-shoes">Мужская обувь</Link>

          <Link to="/woman">Женская обувь</Link>

          <Link to="/kids">Детская обувь</Link>

            </div>

          </div>

          <a href="/cart" className="nav-link">Корзина</a>

          <div className="account-icon" onClick={() => window.location.href = '/auth'}></div>

        </nav>

      </header>

      <…>

      <footer className="footer">

        <div className="contact">

          <p>ФИО: Шманай Виктория</p>

          <p>Телефон: +375298652225</p>

        </div>

        <div className="socials">

          <a href="https://vk.com" target="\_blank" rel="noopener noreferrer">

            VK

          </a>

          <a

            href="https://instagram.com"

            target="\_blank"

            rel="noopener noreferrer"

          >

            Instagram

          </a>

          <a

            href="https://facebook.com"

            target="\_blank" rel="noopener noreferrer"

          >

            Facebook

          </a>

        </div>

      </footer>

    </div>

  );

};

export default AboutUs;

Листинг AboutUs.js

import React, { useState } from "react";

import { Link } from "react-router-dom";

import "./App.css";

import logo from "./images/logot.png";

import slide1 from "./images/assor.jpg";

import slide2 from "./images/1w.jpg";

import slide3 from "./images/4w.jpg";

import slide5 from "./images/men.jpg";

import slide6 from "./images/woman.jpg";

import slide7 from "./images/kids.jpg";

import slide8 from "./images/company.jpg";

function App() {

  const slides = [slide1, slide2, slide3];

  const [currentSlide, setCurrentSlide] = useState(0);

  const [dropdownVisible, setDropdownVisible] = useState(false);

  const nextSlide = () => setCurrentSlide((currentSlide + 1) % slides.length);

  const prevSlide = () =>

    setCurrentSlide((currentSlide - 1 + slides.length) % slides.length);

  return (

    <div className="app">

      <header className="header">

        <img

          src={logo}

          alt="Foskys Boots Logo"

          className="logo"

          onClick={() => window.location.reload()}

        />

        <nav className="nav">

          <Link to="/about" className="nav-link">Про Нас</Link>

                              <div className="nav-link" onClick={() => setDropdownVisible(!dropdownVisible)}>

                                  Каталог

                                  <div className={`dropdown-menu ${dropdownVisible ? 'visible' : ''}`}>

                                      <Link to="/mens-shoes">Мужская обувь</Link>

                                      <Link to="/woman">Женская обувь</Link>

                                      <Link to="/kids">Детская обувь</Link>

                                  </div>

          </div>

          <Link to="/cart" className="nav-link">Корзина</Link>

          <div className="account-icon" onClick={() => window.location.href = '/auth'}></div>

        </nav>

      </header>

      <…>

      <footer className="footer">

        <div className="contact">

          <p>ФИО: Шманай Виктория</p>

          <p>Телефон: +375298652225</p>

        </div>

        <div className="socials">

          <a href="https://vk.com" target="\_blank" rel="noopener noreferrer">

            VK

          </a>

          <a

            href="https://instagram.com"

            target="\_blank"

            rel="noopener noreferrer"

          >

            Instagram

          </a>

          <a

            href="https://facebook.com"

            target="\_blank" rel="noopener noreferrer"

          >

            Facebook

          </a>

        </div>

      </footer>

    </div>

  );

}

export default App;

Листинг App.js

import React, { useState } from "react";

import { useNavigate } from "react-router-dom";

import "./Auth.css";

const Auth = () => {

  const [isRegister, setIsRegister] = useState(true);

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const [error, setError] = useState("");

  const [profilePhoto, setProfilePhoto] = useState(null);

  const [preview, setPreview] = useState(null);

  const navigate = useNavigate();

<…>

  return (

    <div className="auth-container">

      <h1>{isRegister ? "Регистрация" : "Вход"}</h1>

      <div className="auth-switch">

        <button

          onClick={() => setIsRegister(true)}

          className={isRegister ? "active" : ""}>

          Регистрация

        </button>

        <button

          onClick={() => setIsRegister(false)}

          className={!isRegister ? "active" : ""}

        >

          Вход

        </button>

      </div>

      <form onSubmit={handleSubmit} className="auth-form">

        <label>

          Email:

          <input

            type="email"

            placeholder="Введите вашу почту"

            value={email}

            onChange={(e) => setEmail(e.target.value)}

          />

        </label>

        <label>

          Пароль:

          <input

            type="password"

            placeholder="Введите ваш пароль"

            value={password}

            onChange={(e) => setPassword(e.target.value)}

          />

        </label>

        {isRegister && (

          <label className="photo-upload">

            Фото профиля:

            <input

              type="file"

              accept="image/\*"

              onChange={handlePhotoChange}

            />

            {preview && (

              <img

                src={preview}

                alt="Предпросмотр"

                className="profile-preview"

              />

            )}

          </label>

        )}

        {error && <p className="error-message">{error}</p>}

        <button type="submit" className="auth-button">

          {isRegister ? "Зарегистрироваться" : "Войти"}

        </button>

      </form>

    </div>

  );

};

export default Auth;

Листинг Auth.js

import React, { useState } from "react";

import { Link } from "react-router-dom";

import "./Cart.css";

import logo from "./images/logot.png";

const shoesData = [

  { id: 1, name: "Nike", category: "Кроссовки", size: 40, price: 330, quantity: 1, image: "https://via.placeholder.com/150" },

  { id: 2, name: "Mst.Martins", category: "Ботинки", size: 42, price: 245, quantity: 1, image: "https://via.placeholder.com/150" },];

const Cart = () => {

  const [items, setItems] = useState(shoesData);

  const [dropdownVisible, setDropdownVisible] = useState(false);

  const increaseQuantity = (id) => {

    setItems(items.map((item) => (item.id === id ? { ...item, quantity: item.quantity + 1 } : item)));

  };

  <…>

      <div className="cart-summary">

          <p>Итоговая сумма: {totalPrice.toLocaleString()} ₽</p>

          {items.length > 0 && (

            <button className="btn-checkout" onClick={handleCheckout}>

              Оформить заказ

            </button>

          )}

        </div>

      </main>

      <footer className="footer">

        <div className="contact">

          <p>ФИО: Шманай Виктория</p>

          <p>Телефон: +375298652225</p>

        </div>

        <div className="socials">

          <a href="https://vk.com" target="\_blank" rel="noopener noreferrer">VK</a>

          <a href="https://instagram.com" target="\_blank" rel="noopener noreferrer">Instagram</a>

          <a href="https://facebook.com" target="\_blank" rel="noopener noreferrer">Facebook</a>

        </div>

      </footer>

    </div>

  );

};

export default Cart;

Листинг Cart.js

import React from "react";

import "../cart.css";

const CartShoesPage = ({ cartItems, setCartItems }) => {

  const handleCheckout = () => {

    if (cartItems.length > 0) {

      setCartItems([]);

      alert("Ваш заказ оформлен! Спасибо за покупку.");

    } else {

      alert("Ваша корзина пуста. Добавьте товары перед оформлением заказа.");

    }

  };

<…>

      <button className="checkout-button" onClick={handleCheckout}>

        Оформить заказ

      </button>

    </div>

  );

};

export default CartShoesPage;

Листинг CartShoesPage.js

import React, { useState } from "react";

import { Link } from "react-router-dom";

import "./Catalog.css";

import logo from "./images/logot.png";

import foto1 from "./images/1m.jpg";

import foto2 from "./images/2m.jpg";

import foto3 from "./images/3m.jpg";

import foto4 from "./images/4m.jpg";

import foto5 from "./images/5m.jpg";

import foto6 from "./images/6m.jpg";

import foto7 from "./images/7m.jpg";

import foto8 from "./images/8m.jpg";

import foto9 from "./images/9m.jpg";

<…>

const Catalog = () => {

  const [dropdownVisible, setDropdownVisible] = useState(false);

  const [searchTerm, setSearchTerm] = useState('');

  const [sortedProducts, setSortedProducts] = useState(productData);

  const handleSearch = (event) => {

    setSearchTerm(event.target.value);

    const filteredProducts = productData.filter(product =>

      product.name.toLowerCase().includes(event.target.value.toLowerCase())

    );

    setSortedProducts(filteredProducts);

  };

  const handleSort = (order) => {

    const sorted = [...sortedProducts].sort((a, b) => {

      const priceA = parseInt(a.price.replace('$', ''));

      const priceB = parseInt(b.price.replace('$', ''));

      return order === 'asc' ? priceA - priceB : priceB - priceA;

    });

    setSortedProducts(sorted);

  };

  return (

    <div className="catalog-page">

      <header className="header">

        <img

          src={logo}

          alt="Foskys Boots Logo"

          className="logo"

          onClick={() => window.location.href = '/'}

        />

        <nav className="nav">

          <Link to="/" className="nav-link">Главная</Link>

          <Link to="/about" className="nav-link">Про Нас</Link>

                              <div className="nav-link" onClick={() => setDropdownVisible(!dropdownVisible)}>

                                  Каталог

                                  <div className={`dropdown-menu ${dropdownVisible ? 'visible' : ''}`}>

                                      <Link to="/mens-shoes">Мужская обувь</Link>

                                      <Link to="/woman">Женская обувь</Link>

                                      <Link to="/kids">Детская обувь</Link>

                                  </div>

                    </div>

          <Link to="/cart" className="nav-link">Корзина</Link>

          <div className="account-icon" onClick={() => window.location.href = '/auth'}></div>

        </nav>

      </header>

      <main className="catalog-content">

        <h1 className="welcome">Мужской ассортимент</h1>

        <div className="search-sort">

          <input

            type="text"

            placeholder="Поиск по названию"

            value={searchTerm}

            onChange={handleSearch}

            className="search-input"

          />

          <div className="sort-box">

            <span>Сортировка по цене:</span>

            <button className="sort-button" onClick={() => handleSort('asc')}>

            ▲

            </button>

            <button className="sort-button" onClick={() => handleSort('desc')}>

            ▼

            </button>

          </div>

        </div>

        <div className="products-grid">

          {sortedProducts.map((product, index) => (

            <div className="product-card" key={index}>

              <div className="photo-placeholder">

                {product.image}

              </div>

              <h2 className="product-name">{product.name}</h2>

              <p className="price-color">{product.price}</p>

              <p className="sizesText">Sizes:</p>

              <div className="product-sizes">

                {product.sizes.map((size) => (

                  <span key={size}>{size}</span>

                ))}

              </div>

              <button className="btn-buy">Купить</button>

            </div>

          ))}

        </div>

      </main>

      <footer className="footer">

        <div className="contact">

          <p>ФИО: Шманай Виктория</p>

          <p>Телефон: +375298652225</p>

        </div>

        <div className="socials">

          <a href="https://vk.com" target="\_blank" rel="noopener noreferrer">VK</a>

          <a href="https://instagram.com" target="\_blank" rel="noopener noreferrer">Instagram</a>

          <a href="https://facebook.com" target="\_blank" rel="noopener noreferrer">Facebook</a>

        </div>

      </footer>

    </div>

  );

};

export default Catalog;

Листинг Catalog.js

import React, { useState, useEffect } from "react";

import { Link } from "react-router-dom";

import "./Kids.css";

import logo from "./images/logot.png";

import foto1 from "./images/1k.jpg";

import foto2 from "./images/2k.jpg";

import foto3 from "./images/3k.jpg";

import foto4 from "./images/4k.jpg";

import foto5 from "./images/5k.jpg";

import foto6 from "./images/6k.jpg";

import foto7 from "./images/7k.jpg";

import foto8 from "./images/8k.jpg";

import foto9 from "./images/9k.jpg";

<…>

const Kids = () => {

  const [dropdownVisible, setDropdownVisible] = useState(false);

  const [searchTerm, setSearchTerm] = useState('');

  const [sortedProducts, setSortedProducts] = useState(productData);

  useEffect(() => {

    if (window.location.pathname === "/woman") {

      scrollToTop();

    }

  }, []);

  const handleSearch = (event) => {

    setSearchTerm(event.target.value);

    const filteredProducts = productData.filter(product =>

      product.name.toLowerCase().includes(event.target.value.toLowerCase())

    );

    setSortedProducts(filteredProducts);

  };

  <…>

      <footer className="footer">

        <div className="contact">

          <p>ФИО: Шманай Виктория</p>

          <p>Телефон: +375298652225</p>

        </div>

        <div className="socials">

          <a href="https://vk.com" target="\_blank" rel="noopener noreferrer">VK</a>

          <a href="https://instagram.com" target="\_blank" rel="noopener noreferrer">Instagram</a>

          <a href="https://facebook.com" target="\_blank" rel="noopener noreferrer">Facebook</a>

        </div>

      </footer>

    </div>

  );

};

export default Kids;

Листинг Kids.js

import React, { useState } from "react";

import "./App.css";

const Product = ({ product, onAddToCart }) => {

  const [selectedSize, setSelectedSize] = useState(null);

  const handleAddToCart = () => {

    if (!selectedSize) {

      alert("Пожалуйста, выберите размер перед покупкой!");

      return;

    }

    onAddToCart({

      id: product.id,

      name: product.name,

      image: product.image,

      size: selectedSize,

      price: product.price,

      quantity: 1,

    });

  };

  return (

    <div className="product-card">

      <img src={product.image} alt={product.name} className="product-image" />

      <h3>{product.name}</h3>

      <p>${product.price}</p>

      <div className="size-selector">

        <p>Выберите размер:</p>

        {product.sizes.map((size) => (

          <button

            key={size}

            className={`size-btn ${selectedSize === size ? "active" : ""}`}

            onClick={() => setSelectedSize(size)}

          >

            {size}

          </button>

        ))}

      </div>

      <button className="add-to-cart-btn" onClick={handleAddToCart}>

        Купить

      </button>

    </div>

  );

};

export default Product;

Листинг Product.js

import React, { useState, useEffect } from "react";

import { Link } from "react-router-dom";

import "./Woman.css";

import logo from "./images/logot.png";

import foto1 from "./images/1w.jpg";

import foto2 from "./images/2w.jpg";

import foto3 from "./images/3w.jpg";

import foto4 from "./images/4w.jpg";

import foto5 from "./images/5w.jpg";

import foto6 from "./images/6w.jpg";

import foto7 from "./images/7w.jpg";

import foto8 from "./images/8w.jpg";

import foto9 from "./images/9w.jpg";

<…>

const Woman = () => {

  const [dropdownVisible, setDropdownVisible] = useState(false);

  const [searchTerm, setSearchTerm] = useState('');

  const [sortedProducts, setSortedProducts] = useState(productData);

  useEffect(() => {

    // Check if URL contains "/woman" to scroll to the top

    if (window.location.pathname === "/woman") {

      scrollToTop();

    }

  }, []);

  const handleSearch = (event) => {

    setSearchTerm(event.target.value);

    const filteredProducts = productData.filter(product =>

      product.name.toLowerCase().includes(event.target.value.toLowerCase())

    );

    setSortedProducts(filteredProducts);

  };

  const handleSort = (order) => {

    const sorted = [...sortedProducts].sort((a, b) => {

      const priceA = parseInt(a.price.replace('BYN', ''));

      const priceB = parseInt(b.price.replace('BYN', ''));

      return order === 'asc' ? priceA - priceB : priceB - priceA;

    });

    setSortedProducts(sorted);

  };

  const scrollToTop = () => {

    window.scrollTo({

      top: 0,

      behavior: 'smooth'

    });

  };

  <…>

      <main className="catalog-content">

        <h1 className="welcome">Женский ассортимент</h1>

        <div className="search-sort">

          <input

            type="text"

            placeholder="Поиск по названию"

            value={searchTerm}

            onChange={handleSearch}

            className="search-input"

          />

          <div className="sort-box">

            <span>Сортировка по цене:</span>

            <button className="sort-button" onClick={() => handleSort('asc')}>

            ▲

            </button>

            <button className="sort-button" onClick={() => handleSort('desc')}>

            ▼

            </button>

          </div>

        </div>

<…>

      <footer className="footer">

        <div className="contact">

          <p>ФИО: Шманай Виктория</p>

          <p>Телефон: +375298652225</p>

        </div>

        <div className="socials">

          <a href="https://vk.com" target="\_blank" rel="noopener noreferrer">VK</a>

          <a href="https://instagram.com" target="\_blank" rel="noopener noreferrer">Instagram</a>

          <a href="https://facebook.com" target="\_blank" rel="noopener noreferrer">Facebook</a>

        </div>

      </footer>

    </div>

  );

};

export default Woman;

Листинг Woman.js