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 1 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.А.Вовна

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 11](#_35nkun2)

[3 Последовательность разработки содержания и структуры веб-сайта с использованием компонентов 12](#_1ksv4uv)

[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 17](#_2xcytpi)

[4.3 Тестирование ПП 18](#_1ci93xb)

[4.4 Вывод 19](#_3whwml4)

[Заключение 20](#_2bn6wsx)

[Список использованных литературных источников 21](#_qsh70q)

[5 Приложения 22](#_3as4poj)

**Введение**

Веб-приложения находят широкое применение в различных сферах благодаря своей способности предоставлять обновляемый контент по запросу пользователя. Такие веб-страницы, содержание которых формируется в зависимости от запросов, называются динамическими.

Сегодня существует высокий спрос на веб-приложения, позволяющие пользователям удобно совершать покупки, например, в интернет-магазинах. Особенно актуальны решения, предлагающие продажу автозапчастей с использованием различных устройств — от смартфонов до персональных компьютеров. 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 разработка кроссплатформенного и кроссбраузерного веб-приложения для продажи автозапчастей. 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 для одежды. В ходе анализа большинства подобных сайтов было выявлено, что контент, как правило, располагается в центральной части страницы, в то время как в верхней части находится меню.

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

Одним из таких сайтов является Mark Formelle.by — известный интернет-магазин одежды в Беларуси.

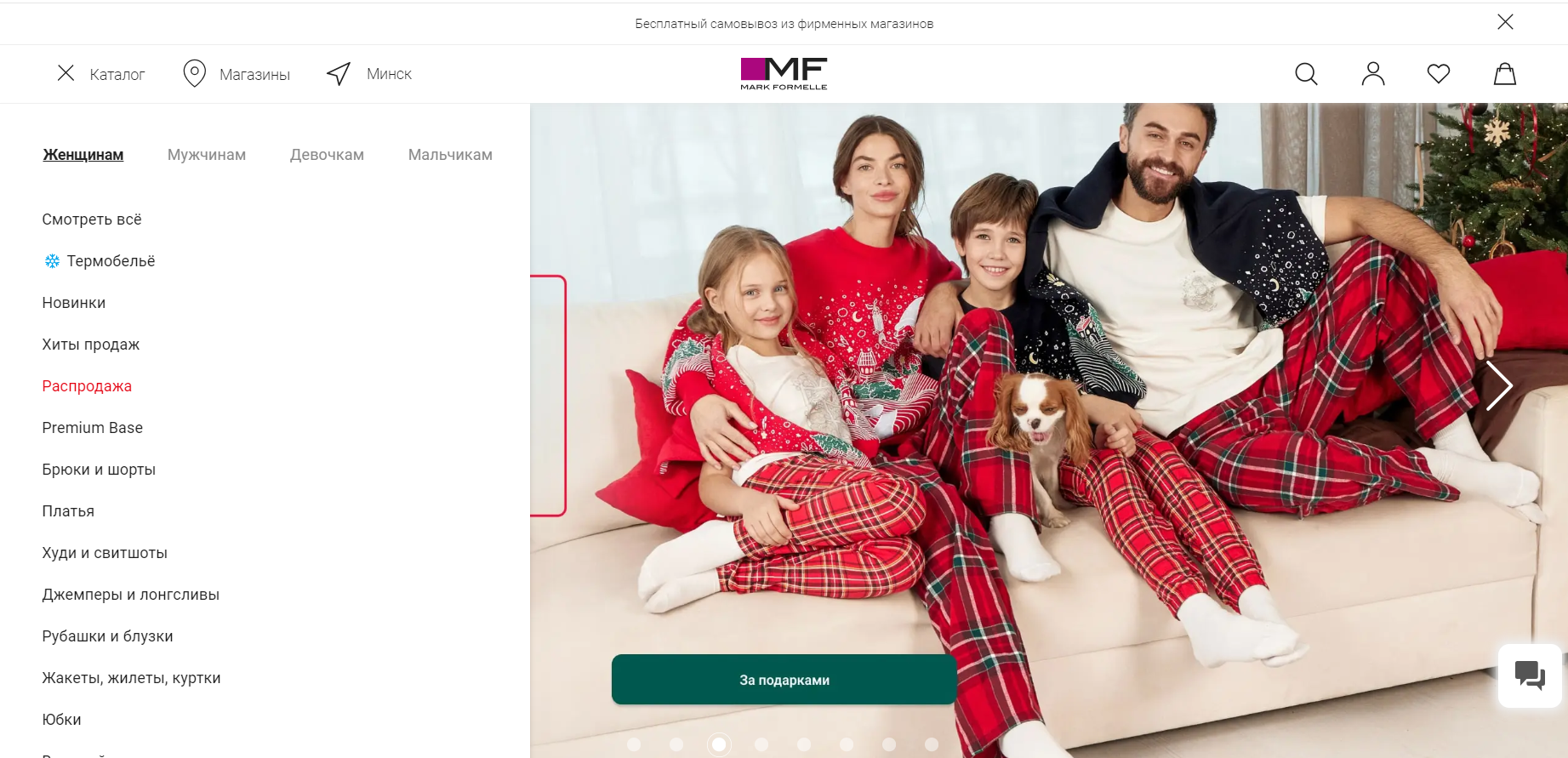


Рисунок 1.1 – главная страница Mark Formelle, и ее выдвигающееся меню

Слайдер, расположенный на главной странице сайта, служит эффективным инструментом привлечения внимания пользователей и побуждает их к дальнейшему изучению каталога. Удобная структура поиска товаров, реализованная на сайте, способствует более эффективному поиску необходимой продукции.

Примером подобного подхода может служить популярный сайт ZARA.by, предлагающий широкий ассортимент одежды. Каталог интернет-магазина позволяет пользователям просматривать товары, фильтровать их по различным параметрам и осуществлять поиск по названию.

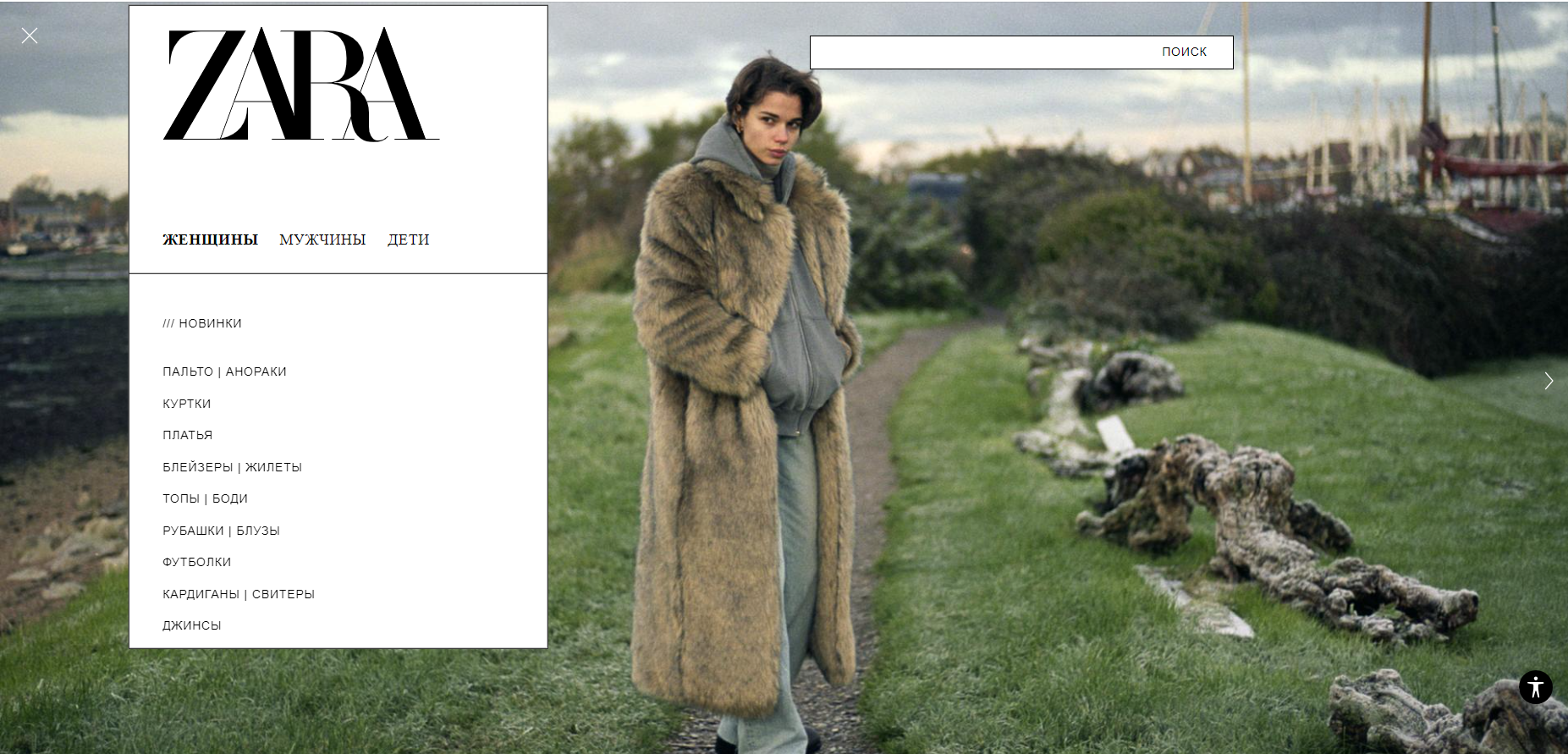


Рисунок 1.2 – каталог сайта ZARA

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

**1.2 Формирование требований к ПП**

Проведя анализ доступных примеров, можно сделать вывод, что типичный интернет-магазин одежды включает в себя привлекательную 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 одежду 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.

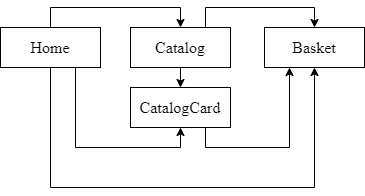


Рисунок 2.1 – структура приложения

На стартовой странице сайта, расположенной в разделе Home, располагается ключевой элемент, предназначенный для привлечения внимания. Также здесь можно увидеть крупный заголовок, который привлекает внимание, 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.

Также здесь присутствует элемент CatalogCard. Если выбрать товар в этом элементе, можно перейти на страницу, где представлен элемент Basket.

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

**2.2 Выбор стилевого оформления**

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

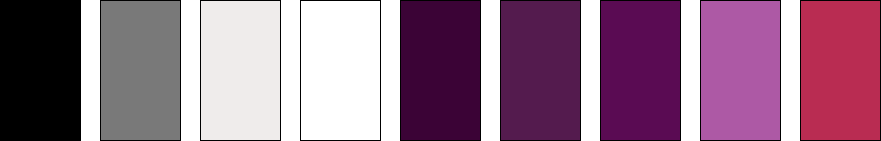


Рисунок 2.2 – Цветовая палитра сайта

Акцент выбора цветов делался на универсальную тематику: фиолетовые тона хорошо акцентируют важные элементы, а универсальности добавляют оттенки черного, отлично подходящие в этом случае. Ключевой цвет для оформления текста - черный(#000000) используется в оформлении заголовка и в выделениях в текстах. #5A0B53 используется для выделения ключевых элементов и акцента на сайте. Также он присутствует в заднем фоне иконки сайта. #3B0336 используется для цветов кнопки по всем компонентам сайта. #FFFFFF используется для заднего фона страниц сайта, и в некоторых текстах сайта. #B92C52 в основном используется для текстов и декоративных элементов.

**2.3 Выбор шрифтового оформления**

В данном проекте используется один единственный шрифт Poppins. 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 вышеупомянутым шрифтом.



Рисунок 2.3 – логотип интернет-магазина Your Choice

Решение сделать логотипа сайта таковым было обусловлено его стилистическим решением: иметь логотипа рядом с выдвигающимся меню было не совсем корректно с точки зрения дизайна, поэтому логотип сайта остался таковым какой он есть.

**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 всей сессии посещения сайта, у пользователя будет 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 ознакомиться с списком товаров сайта, состоящим из миниатюр. 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.

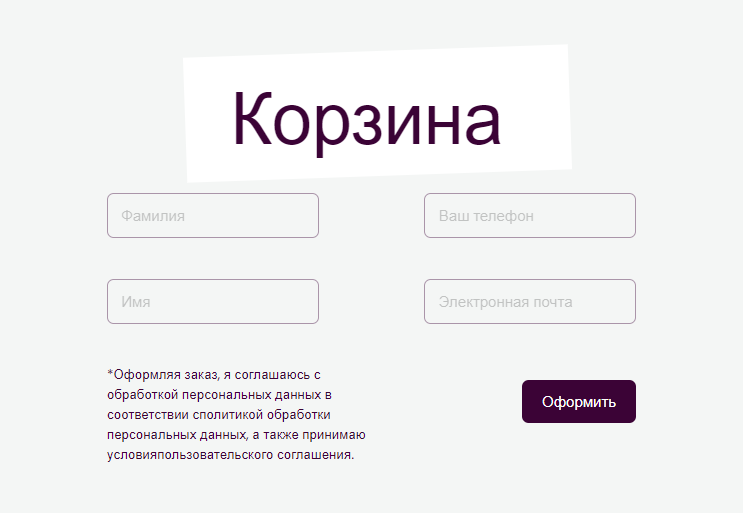


Рисунок 2.6 – форма оформления заказа

Для компонента Basket также необходимо разработать компонент, который будет давать возможность оформить заказ, с указанием данных о покупателе. 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 Basket из страницы корзины приложения.

import './basket.css';

import promoImg from './../../img/images/basket-img.jpg';

import { useState, useEffect } from 'react';

import mask from './../../libs/phoneMask/mask';

const Basket = () => {

const [formData, setFormData] = useState({

lastName: '',

firstName: '',

phone: '',

email: ''

});

const [successMessage, setSuccessMessage] = useState('');

const handleChange = (e) => {

const { name, value } = e.target;

setFormData({ ...formData, [name]: value });

};

const validateEmail = (email) => {

const re = /^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$/;

return re.test(String(email).toLowerCase());

};

const validatePhone = (phone) => {

const cleanedPhone = phone.replace(/\D/g, '');

return cleanedPhone.length >= 10 && cleanedPhone.length <= 15;

};

const handleSubmit = (e) => {

e.preventDefault();

const { lastName, firstName, phone, email } = formData;

if (!lastName || !firstName || !validatePhone(phone) || !validateEmail(email)) {

alert('Пожалуйста, заполните все поля корректно.');

return;

}

setSuccessMessage('Ваш заказ был успешно оформлен!');

setTimeout(() => setSuccessMessage(''), 5000);

};

useEffect(() => {

mask('input[data-tel-input]');

const phoneInput = document.querySelector('input[data-tel-input]');

const handleInput = (e) => {

setFormData((prevFormData) => ({ ...prevFormData, phone: e.target.value }));

};

phoneInput.addEventListener('input', handleInput);

return () => {

phoneInput.removeEventListener('input', handleInput);

};

}, []);

...

Листинг 3.1 – Фрагмент компонента Basket

Для создания компонента `Basket` потребовалось использование таких React-хуков как `useState` и `useEffect`. Этот компонент представляет собой форму корзины, где пользователи могут вводить свои данные для оформления заказа. В целом, данный функциональный компонент обеспечивает пользователю возможность ввода и проверки данных перед отправкой формы. Компонент использует маску ввода для поля телефона, чтобы обеспечить корректный формат ввода номера.

React Hooks:

- `useState` для управления состоянием формы.

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

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

**3.2 Использование таблиц стилей CSS**

.header {

padding: 46px 0 56px;

}

.header\_\_row {

display: flex;

column-gap: 30px;

align-items: center;

justify-content: space-between;

}

.header\_\_logo {

display: flex;

align-items: center;

column-gap: 13px;

}

.header\_\_logo span {

font-weight: 900;

font-size: 40px;

line-height: 1.5;

letter-spacing: 0.03em;

text-transform: uppercase;

}

.header\_\_nav {

font-weight: 500;

font-size: 22px;

line-height: 1.5;

letter-spacing: 0.01em;

}

Листинг 3.3 – фрагмент Header.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**

При заходе в интернет-магазин посетителя встречает главная страница магазина, которая содержит в себе большой слайдер с тематическими картинками, предлагающими перейти к каталогу сайта.

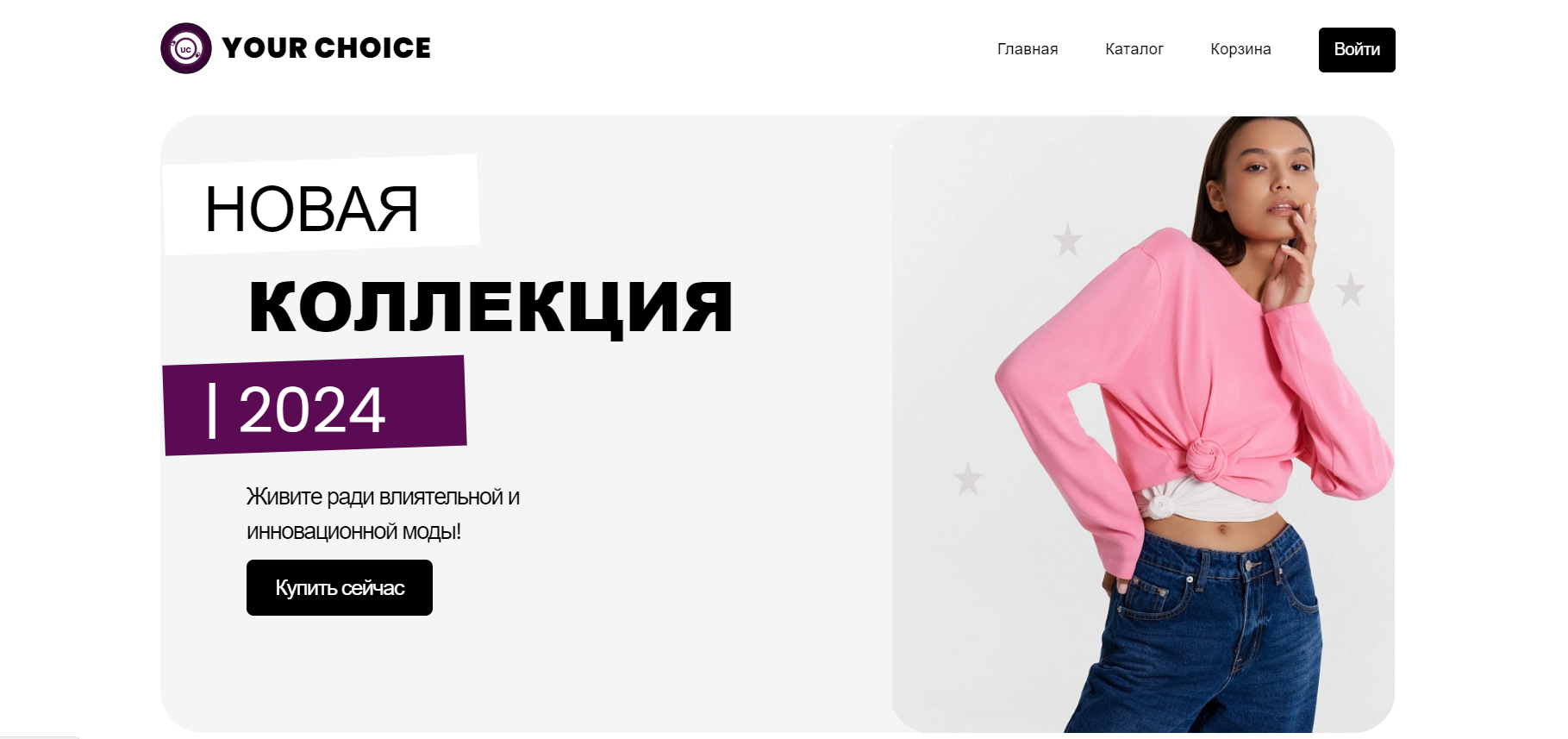


Рисунок 4.1 – Главная страница интернет-магазина

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

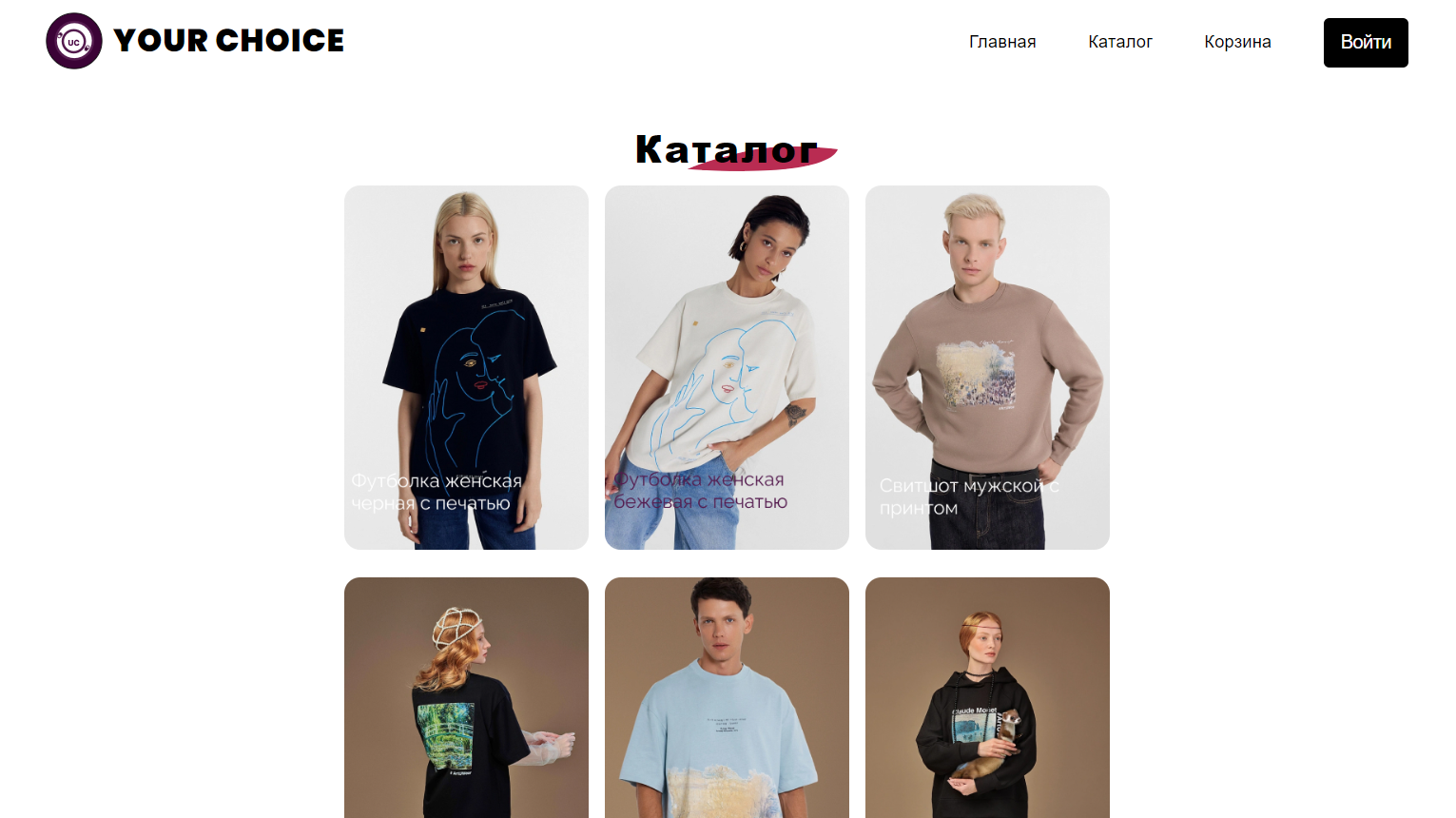


Рисунок 4.2 – Каталог интернет-магазина

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 ввести свои личные данные, для подтверждения товара. Подразумевается, что способ оплаты указан у пользователя в личном кабинете. Форма проверяет каждое поле на валидность и выводит различные сообщения, зависящие от введенных данных.

**4.2 Диаграмма вариантов использования веб-приложения**

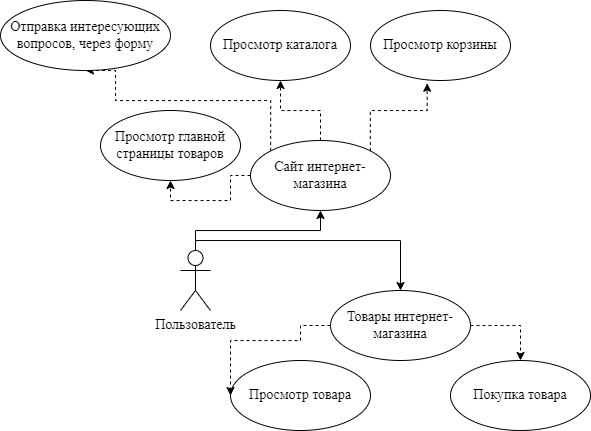


Рисунок 4.4 – Диаграмма вариантов использования интернет-магазина

Диаграмма вариантов использования наглядно демонстрирует взаимодействие пользователя с функциональными возможностями приложения. На ней представлен ключевой 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 автозапчастей 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 сбоев, ошибок в верстке и с одинаково читаемым контентом.

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

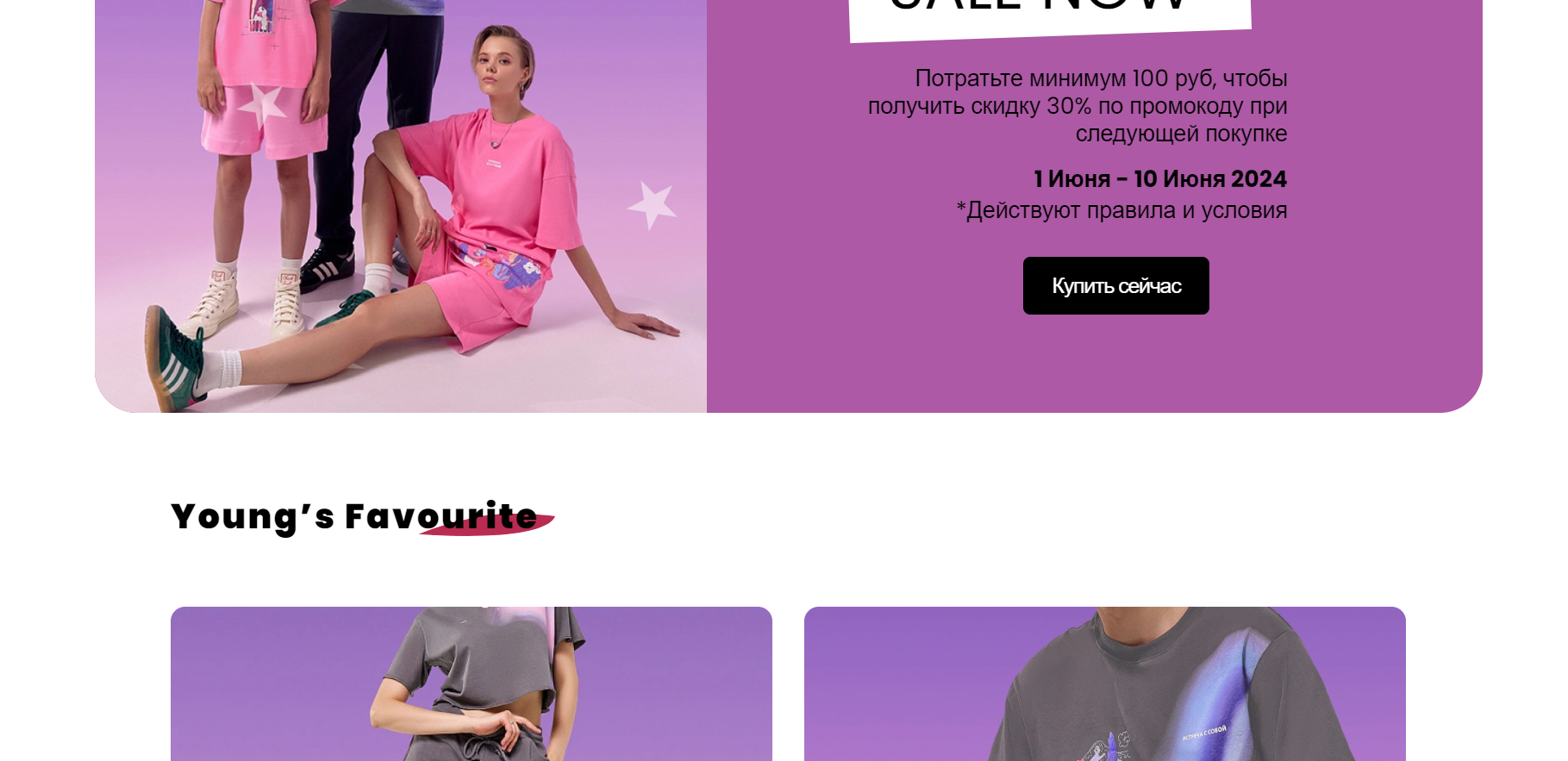


Рисунок 4.5 – Работа в браузере YandexBrowser

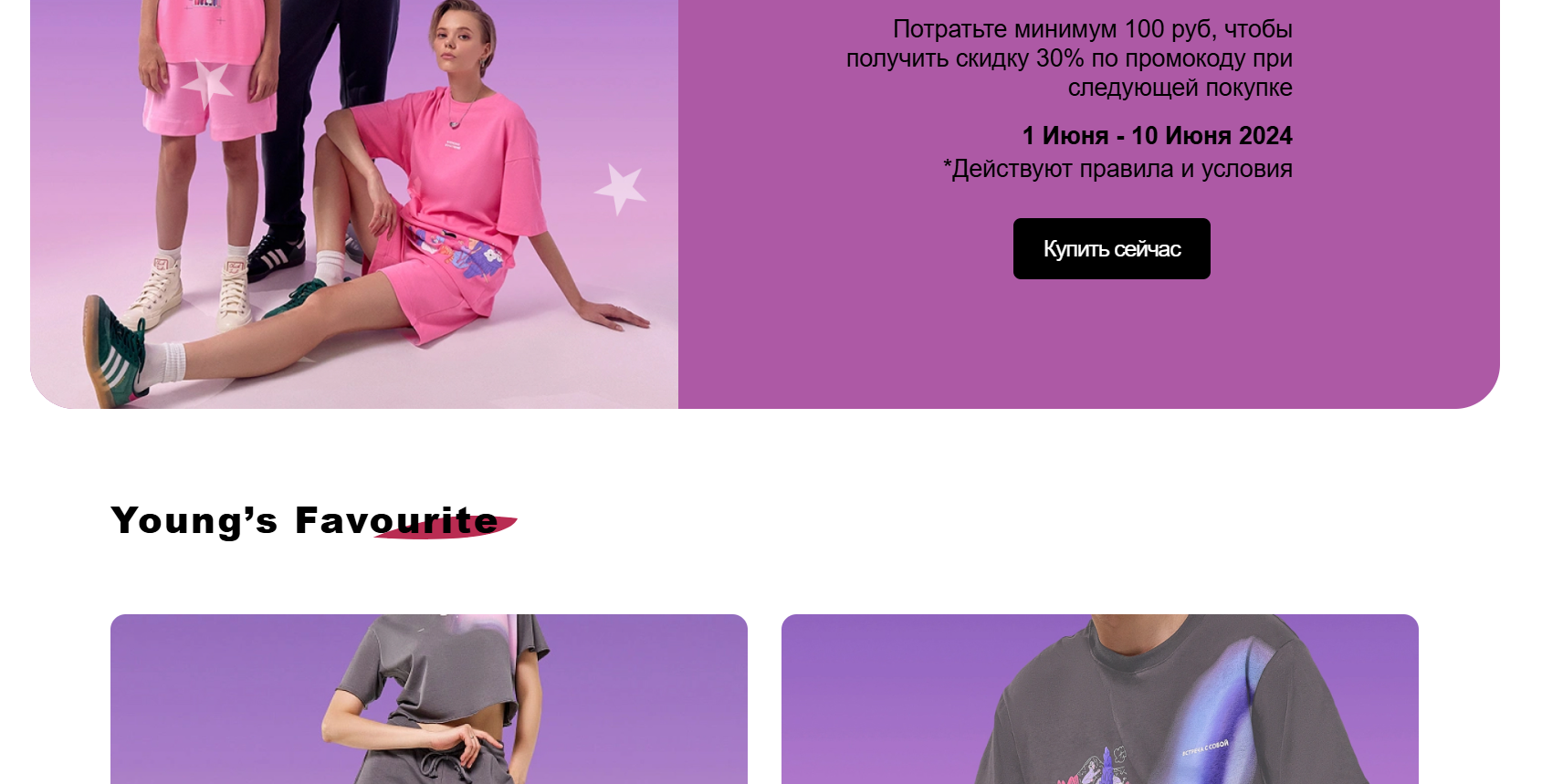


Рисунок 4.6 – Работа в браузере MicroSoft Edge

**4.4 Вывод**

В этом разделе подробно представлено руководство по использованию веб-приложения, которое помогает пользователям лучше понять функционал и возможности интернет-магазина автозапчастей. 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, Mozilla Firefox. 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 одежды».

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 был выбран современный набор 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 уникальный стиль 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 автозапчастей, 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/client';

import App from './App';

import './styles/reset.css';

import './styles/common.css';

ReactDOM.createRoot(document.getElementById('root')).render(

<App />

);

Листинг index.js

import React from 'react';

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

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

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

import Catalog from './components/catalog\_page/Catalog';

import Community from './components/community/Community';

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

import Basket from './components/basket/Basket';

function App() {

return (

<Router>

<div className="App">

<Header />

<Routes>

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

<Route path="/catalog" element={<Catalog />} />

<Route path="/basket" element={<Basket />} />

</Routes>

<Community />

<Footer />

</div>

</Router>

);

}

export default App;

Листинг App.js

import React from 'react';

import Promo from '../promo/Promo';

import Brands from '../brands/Brands';

import Arrivals from '../arrivals/Arrivals';

import Banner from '../banner/Banner';

import YoungsFavourite from '../youngs\_favourite/YoungsFavourite';

import Vouchers from '../vouchers/Vouchers';

function Home() {

return (

<div>

<Promo />

<Brands />

<Arrivals />

<Banner />

<YoungsFavourite />

<Vouchers />

</div>

);

}

export default Home;

Листинг Home.jsx

import logoImg from './../../img/icons/logo.svg'

import './header.css'

import { Link } from 'react-router-dom'

import { useState } from 'react'

function Header() {

const [message, setMessage] = useState('');

const handleLoginClick = () => {

setMessage('Вы успешно вошли на сайт!');

setTimeout(() => setMessage(''), 2500);

};

return (

<header className="header">

<div className="container">

<div className="header\_\_row">

<div className="header\_\_logo">

<img src={logoImg} alt="Logo" />

<span>YOUR CHOICE</span>

</div>

<nav className="header\_\_nav">

<ul>

<li><Link to='/'>Главная</Link></li>

<li><Link to='/catalog'>Каталог</Link></li>

<li><Link to='/basket'>Корзина</Link></li>

<li><a href="#!" className="header\_\_nav-btn" onClick={handleLoginClick}>Войти</a></li>

</ul>

</nav>

</div>

{message && <div className="login-message">{message}</div>}

</div>

</header>

);

}

export default Header;

Листинг Header.jsx

import './promo.css'

import promoImg from './../../img/images/header-img.jpg';

const Promo = () => {

return (

<section className="promo">

<div className="container">

<div className="promo\_\_content">

<div className="promo\_\_text">

<div className="promo\_\_title">

<span class="highlight">

<span>НОВАЯ</span>

</span>

КОЛЛЕКЦИЯ

<span class="highlight highlight--yellow">

<span>| 2024</span>

</span>

</div>

<div className="promo\_\_desc">

Живите ради влиятельной и инновационной моды!

</div>

<div className="promo\_\_btn-wrapper">

<a href="#!" className="banner\_\_btn">

Купить сейчас

</a>

</div>

</div>

<div className="promo\_\_img">

<img src={promoImg} alt="Promo" />

</div>

</div>

</div>

</section>

);

}

export default Promo;

Листинг Promo.jsx

import './brands.css'

import hm from './../../img/brands/HM.png'

import obey from './../../img/brands/Obey.png'

import shopify from './../../img/brands/Shopify.png'

import lacoste from './../../img/brands/Lacoste.png'

import levis from './../../img/brands/Levis.png'

import amazon from './../../img/brands/Amazon.png'

const Brands = () => {

return (<section className="brands">

<div className="container">

<ul className="brands\_\_list">

<li><a href="#!"><img src={hm} alt="" /></a></li>

<li><a href="#!"><img src={obey} alt="" /></a></li>

<li><a href="#!"><img src={shopify} alt="" /></a></li>

<li><a href="#!"><img src={lacoste} alt="" /></a></li>

<li><a href="#!"><img src={levis} alt="" /></a></li>

<li><a href="#!"><img src={amazon} alt="" /></a></li>

</ul>

</div>

</section>);

}

export default Brands;

Листинг Brands.jsx

import React from 'react';

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

import Card from '../card/Card';

import './arrivals.css';

import cat01Img from './../../img/categories/cat-01.jpg';

import cat02Img from './../../img/categories/cat-02.jpg';

import cat03Img from './../../img/categories/cat-03.jpg';

const Arrivals = () => {

return (

<section className="arrivals">

<div className="container">

<div className="arrivals\_\_header">

<h2 className="title-2">NEW ARRIVALS</h2>

</div>

<div className="arrivals\_\_cards">

<Link to="/basket">

<Card title="Толстовки & Свитшоты" img={cat01Img} />

</Link>

<Link to="/basket">

<Card title="Сумки & Обувь" img={cat02Img} />

</Link>

<Link to="/basket">

<Card title="Футболки & Топы" img={cat03Img} />

</Link>

</div>

</div>

</section>

);

};

export default Arrivals;

Листинг Arrivals.jsx

import './banner.css';

import bannerImg from './../../img/images/promo-img.jpg';

const Banner = () => {

return (

<section className="banner">

<div className="banner\_\_content">

<div className="banner\_\_img">

<img src={bannerImg} alt="Banner" />

</div>

<div className="banner\_\_text">

<div className="banner\_\_title-wrapper">

<div className="highlight">

<span>PAYDAY</span>

</div>

<div className="highlight highlight--white">

<span>SALE NOW</span>

</div>

</div>

<div className="banner\_\_text-container">

<div className="banner\_\_desc">

Потратьте минимум 100 руб, чтобы получить скидку 30% по промокоду при следующей покупке

</div>

<div className="banner\_\_desc banner\_\_desc--date">

1 Июня - 10 Июня 2024

</div>

<div className="banner\_\_desc">

\*Действуют правила и условия

</div>

</div>

<div className="banner\_\_btn-wrapper">

<a href="#!" className="banner\_\_btn">

Купить сейчас

</a>

</div>

</div>

</div>

</section>

);

};

export default Banner;

Листинг Banner.jsx

import Card from '../card/Card';

import './youngsfavourite.css';

import cat04Img from './../../img/categories/cat-04.jpg';

import cat05Img from './../../img/categories/cat-05.jpg';

const YoungsFavourite = () => {

return (

<section className="arrivals">

<div className="container">

<div className="arrivals\_\_header">

<h2 className="title-2">Young’s Favourite</h2>

</div>

<div className="arrivals\_\_cards">

<Card title="Trending on instagram" img={cat04Img} />

<Card title="All Under $40" img={cat05Img} />

</div>

</div>

</section>

);

};

export default YoungsFavourite;

Листинг YoungsFavorite.jsx

import React from 'react';

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

import Card from '../card/Card';

import './youngsfavourite.css';

import cat04Img from './../../img/categories/cat-04.jpg';

import cat05Img from './../../img/categories/cat-05.jpg';

const YoungsFavourite = () => {

return (

<section className="arrivals">

<div className="container">

<div className="arrivals\_\_header">

<h2 className="title-2">Young’s Favourite</h2>

</div>

<div className="arrivals\_\_cards">

<Link to="/catalog">

<Card title="Trending on instagram" img={cat04Img} />

</Link>

<Link to="/catalog">

<Card title="All Under $40" img={cat05Img} />

</Link>

</div>

</div>

</section>

);

};

export default YoungsFavourite;

Листинг Vouchers.jsx

import React from 'react';

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

import CatalogCard from '../catalog\_card/CatalogCard';

import './catalog.css';

import Img01\_1 from './../../img/catalog/01\_1.jpg';

import Img02\_1 from './../../img/catalog/02\_1.jpg';

import Img03\_1 from './../../img/catalog/03\_1.jpg';

import Img01\_2 from './../../img/catalog/01\_2.jpg';

import Img02\_2 from './../../img/catalog/02\_2.jpg';

import Img03\_2 from './../../img/catalog/03\_2.jpg';

import Img01\_3 from './../../img/catalog/01\_3.jpg';

import Img02\_3 from './../../img/catalog/02\_3.jpg';

import Img03\_3 from './../../img/catalog/03\_3.jpg';

import Img01\_4 from './../../img/catalog/01\_4.jpg';

import Img02\_4 from './../../img/catalog/02\_4.jpg';

import Img03\_4 from './../../img/catalog/03\_4.jpg';

const Catalog = () => {

return (

<section className="catalog">

<div className="container">

<div className="catalog\_\_header">

<h2 className="title-2">Каталог</h2>

</div>

<div className="catalog\_\_cards">

<Link to="/basket">

<CatalogCard img={Img01\_1} />

</Link>

<Link to="/basket">

<CatalogCard img={Img02\_1} />

</Link>

<Link to="/basket">

<CatalogCard img={Img03\_1} />

</Link>

</div>

<div className="catalog\_\_cards">

<Link to="/basket">

<CatalogCard img={Img01\_2} />

</Link>

<Link to="/basket">

<CatalogCard img={Img02\_2} />

</Link>

<Link to="/basket">

<CatalogCard img={Img03\_2} />

</Link>

</div>

<div className="catalog\_\_cards">

<Link to="/basket">

<CatalogCard img={Img01\_3} />

</Link>

<Link to="/basket">

<CatalogCard img={Img02\_3} />

</Link>

<Link to="/basket">

<CatalogCard img={Img03\_3} />

</Link>

</div>

<div className="catalog\_\_cards">

<Link to="/basket">

<CatalogCard img={Img01\_4} />

</Link>

<Link to="/basket">

<CatalogCard img={Img02\_4} />

</Link>

<Link to="/basket">

<CatalogCard img={Img03\_4} />

</Link>

</div>

</div>

</section>

);

};

export default Catalog;

Листинг Catalog.jsx

import styles from '../card/Card.module.css'

const CatalogCard = ({ title, img }) => {

return (

<div className={styles.card}>

<a href="#!" className={styles.card\_\_link}>

</a>

<img

className={styles.card\_\_img}

src={img}

alt="Category ..."

/>

<div className={styles.card\_\_body}>

<div className={styles.card\_\_text}>

<div className={styles.card\_\_title}>

{title}

</div>

</div>

</div>

</div>

);

}

export default CatalogCard;

Листинг CatalogCard.jsx

import './basket.css';

import promoImg from './../../img/images/basket-img.jpg';

import { useState, useEffect } from 'react';

import mask from './../../libs/phoneMask/mask';

const Basket = () => {

const [formData, setFormData] = useState({

lastName: '',

firstName: '',

phone: '',

email: ''

});

const [successMessage, setSuccessMessage] = useState('');

const handleChange = (e) => {

const { name, value } = e.target;

setFormData({ ...formData, [name]: value });

};

const validateEmail = (email) => {

const re = /^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$/;

return re.test(String(email).toLowerCase());

};

const validatePhone = (phone) => {

const cleanedPhone = phone.replace(/\D/g, '');

return cleanedPhone.length >= 10 && cleanedPhone.length <= 15;

};

const handleSubmit = (e) => {

e.preventDefault();

const { lastName, firstName, phone, email } = formData;

if (!lastName || !firstName || !validatePhone(phone) || !validateEmail(email)) {

alert('Пожалуйста, заполните все поля корректно.');

return;

}

setSuccessMessage('Ваш заказ был успешно оформлен!');

setTimeout(() => setSuccessMessage(''), 5000);

};

useEffect(() => {

mask('input[data-tel-input]');

const phoneInput = document.querySelector('input[data-tel-input]');

const handleInput = (e) => {

setFormData((prevFormData) => ({ ...prevFormData, phone: e.target.value }));

};

phoneInput.addEventListener('input', handleInput);

return () => {

phoneInput.removeEventListener('input', handleInput);

};

}, []);

return (

<section className="basket">

<div className="container">

{successMessage && <div className="popup-message">{successMessage}</div>}

<div className="basket\_\_content">

<div className="basket\_\_form">

<div className="basket\_\_form-title">

<h2 className="title2">

<span className="highlight">

<span>Корзина</span>

</span>

</h2>

</div>

<form className="basket\_\_form-form" onSubmit={handleSubmit}>

<div className="form\_\_row">

<input

type="text"

className="form\_\_input"

placeholder="Фамилия"

name="lastName"

value={formData.lastName}

onChange={handleChange}

autoComplete="off"

required

/>

<input

type="text"

className="form\_\_input"

placeholder="Ваш телефон"

name="phone"

data-tel-input

value={formData.phone}

onChange={handleChange}

autoComplete="off"

required

/>

</div>

<div className="form\_\_row">

<input

type="text"

className="form\_\_input"

placeholder="Имя"

name="firstName"

value={formData.firstName}

onChange={handleChange}

autoComplete="off"

required

/>

<input

type="email"

className="form\_\_input"

placeholder="Электронная почта"

name="email"

value={formData.email}

onChange={handleChange}

autoComplete="off"

required

/>

</div>

<div className="form\_\_privacy-wrapper">

<p className="form\_\_privacy">

\*Оформляя заказ, я соглашаюсь с обработкой персональных данных в соответствии с

<a href="https://markformelle.by/internet-magazin/politika-o-zashchite-personalnykh-dannykh/" target="\_blank" rel="noopener noreferrer">

политикой обработки персональных данных

</a>, а также принимаю условия

<a href="https://markformelle.by/internet-magazin/public-offert/" target="\_blank" rel="noopener noreferrer">

пользовательского соглашения

</a>.

</p>

<button type="submit" className="form\_\_btn">Оформить</button>

</div>

</form>

</div>

<div className="basket\_\_img">

<img src={promoImg} alt="Promo" />

</div>

</div>

</div>

</section>

);

};

export default Basket;

Листинг Basket.jsx

const maskList = [

{ "code": "+247 ####" },

{ "code": "+290 ####" },

{ "code": "+290 ####" },

{ "code": "+683 ####" },

{ "code": "+690 ####" },

{ "code": "+500 #####" },

{ "code": "+676 #####" },

{ "code": "+677 #####" },

{ "code": "+678 #####" },

{ "code": "+688 2####" },

{ "code": "+49 ### ###" },

{ "code": "+682 ## ###" },

{ "code": "+686 ## ###" },

{ "code": "+688 90####" },

{ "code": "+95 ### ###" },

{ "code": "+298 ### ###" },

{ "code": "+376 ### ###" },

{ "code": "+387 ## ####" },

{ "code": "+508 ## ####" },

{ "code": "+597 ### ###" },

{ "code": "+672 1## ###" },

{ "code": "+672 3## ###" },

{ "code": "+681 ## ####" },

{ "code": "+685 ## ####" },

{ "code": "+687 ## ####" },

{ "code": "+850 ### ###" },

{ "code": "+230 ### ####" },

{ "code": "+239 ## #####" },

{ "code": "+245 # ######" },

{ "code": "+246 ### ####" },

{ "code": "+263 # ######" },

{ "code": "+269 ## #####" },

{ "code": "+297 ### ####" },

{ "code": "+299 ## ## ##" },

{ "code": "+354 ### ####" },

{ "code": "+372 ### ####" },

{ "code": "+387 ## #####" },

{ "code": "+49 ### ## ##" },

{ "code": "+501 ### ####" },

{ "code": "+507 ### ####" },

{ "code": "+592 ### ####" },

{ "code": "+597 ### ####" },

{ "code": "+599 ### ####" },

{ "code": "+599 ### ####" },

{ "code": "+599 ### ####" },

{ "code": "+60 # ### ###" },

{ "code": "+62 ## ### ##" },

{ "code": "+65 #### ####" },

{ "code": "+670 ### ####" },

{ "code": "+673 ### ####" },

{ "code": "+674 ### ####" },

{ "code": "+677 ### ####" },

{ "code": "+678 ## #####" },

{ "code": "+679 ## #####" },

{ "code": "+680 ### ####" },

{ "code": "+689 ## ## ##" },

{ "code": "+691 ### ####" },

{ "code": "+692 ### ####" },

{ "code": "+95 # ### ###" },

{ "code": "+960 ### ####" },

{ "code": "+220 ### ## ##" },

{ "code": "+232 ## ######" },

{ "code": "+234 ## ### ##" },

{ "code": "+237 #### ####" },

{ "code": "+238 ### ## ##" },

{ "code": "+248 # ### ###" },

{ "code": "+252 # ### ###" },

{ "code": "+252 # ### ###" },

{ "code": "+265 1 ### ###" },

{ "code": "+291 # ### ###" },

{ "code": "+350 ### #####" },

{ "code": "+356 #### ####" },

{ "code": "+372 #### ####" },

{ "code": "+373 #### ####" },

{ "code": "+47 ### ## ###" },

{ "code": "+49 ### ## ###" },

{ "code": "+504 #### ####" },

{ "code": "+505 #### ####" },

{ "code": "+506 #### ####" },

{ "code": "+52 ## ## ####" },

{ "code": "+53 # ### ####" },

{ "code": "+599 9### ####" },

{ "code": "+60 ## ### ###" },

{ "code": "+62 ## ### ###" },

{ "code": "+64 ## ### ###" },

{ "code": "+66 ## ### ###" },

{ "code": "+670 77# #####" },

{ "code": "+670 78# #####" },

{ "code": "+850 #### ####" },

{ "code": "+852 #### ####" },

{ "code": "+853 #### ####" },

{ "code": "+886 #### ####" },

{ "code": "+95 ## ### ###" },

{ "code": "+961 # ### ###" },

{ "code": "+965 #### ####" },

{ "code": "+967 # ### ###" },

{ "code": "+973 #### ####" },

{ "code": "+974 #### ####" },

{ "code": "+975 # ### ###" },

{ "code": "+1 ### ### ####" },

{ "code": "+1 242 ### ####" },

{ "code": "+1 246 ### ####" },

{ "code": "+1 264 ### ####" },

{ "code": "+1 268 ### ####" },

{ "code": "+1 284 ### ####" },

{ "code": "+1 340 ### ####" },

{ "code": "+1 345 ### ####" },

{ "code": "+1 441 ### ####" },

{ "code": "+1 473 ### ####" },

{ "code": "+1 649 ### ####" },

{ "code": "+1 664 ### ####" },

{ "code": "+1 670 ### ####" },

{ "code": "+1 671 ### ####" },

{ "code": "+1 684 ### ####" },

{ "code": "+1 721 ### ####" },

{ "code": "+1 758 ### ####" },

{ "code": "+1 767 ### ####" },

{ "code": "+1 784 ### ####" },

{ "code": "+1 809 ### ####" },

{ "code": "+1 829 ### ####" },

{ "code": "+1 849 ### ####" },

{ "code": "+1 868 ### ####" },

{ "code": "+1 869 ### ####" },

{ "code": "+1 876 ### ####" },

{ "code": "+216 ## ### ###" },

{ "code": "+218 ## ### ###" },

{ "code": "+222 ## ## ####" },

{ "code": "+223 ## ## ####" },

{ "code": "+224 ## ### ###" },

{ "code": "+225 ## ### ###" },

{ "code": "+226 ## ## ####" },

{ "code": "+227 ## ## ####" },

{ "code": "+228 ## ### ###" },

{ "code": "+229 ## ## ####" },

{ "code": "+231 ## ### ###" },

{ "code": "+234 ## ### ###" },

{ "code": "+236 ## ## ####" },

{ "code": "+241 # ## ## ##" },

{ "code": "+252 ## ### ###" },

{ "code": "+254 ### ######" },

{ "code": "+257 ## ## ####" },

{ "code": "+258 ## ### ###" },

{ "code": "+262 ##### ####" },

{ "code": "+262 ##### ####" },

{ "code": "+266 # ### ####" },

{ "code": "+267 ## ### ###" },

{ "code": "+268 ## ## ####" },

{ "code": "+27 ## ### ####" },

{ "code": "+31 ## ### ####" },

{ "code": "+32 ### ### ###" },

{ "code": "+33 ### ### ###" },

{ "code": "+34 ### ### ###" },

{ "code": "+357 ## ### ###" },

{ "code": "+36 ### ### ###" },

{ "code": "+370 ### ## ###" },

{ "code": "+371 ## ### ###" },

{ "code": "+374 ## ### ###" },

{ "code": "+377 ## ### ###" },

{ "code": "+382 ## ### ###" },

{ "code": "+385 ## ### ###" },

{ "code": "+386 ## ### ###" },

{ "code": "+389 ## ### ###" },

{ "code": "+39 6 698 #####" },

{ "code": "+40 ## ### ####" },

{ "code": "+41 ## ### ####" },

{ "code": "+45 ## ## ## ##" },

{ "code": "+46 ## ### ####" },

{ "code": "+48 ### ### ###" },

{ "code": "+49 ### ## ####" },

{ "code": "+502 # ### ####" },

{ "code": "+503 ## ## ####" },

{ "code": "+509 ## ## ####" },

{ "code": "+51 ### ### ###" },

{ "code": "+56 # #### ####" },

{ "code": "+591 # ### ####" },

{ "code": "+593 # ### ####" },

{ "code": "+594 ##### ####" },

{ "code": "+60 ## ### ####" },

{ "code": "+60 ### ### ###" },

{ "code": "+61 # #### ####" },

{ "code": "+62 ## ### ####" },

{ "code": "+62 8## ### ###" },

{ "code": "+64 ### ### ###" },

{ "code": "+66 ## ### ####" },

{ "code": "+675 ### ## ###" },

{ "code": "+81 ### ### ###" },

{ "code": "+82 ## ### ####" },

{ "code": "+84 ## #### ###" },

{ "code": "+850 ## ### ###" },

{ "code": "+855 ## ### ###" },

{ "code": "+856 ## ### ###" },

{ "code": "+880 ## ### ###" },

{ "code": "+93 ## ### ####" },

{ "code": "+94 ## ### ####" },

{ "code": "+961 ## ### ###" },

{ "code": "+966 # ### ####" },

{ "code": "+967 ## ### ###" },

{ "code": "+968 ## ### ###" },

{ "code": "+971 # ### ####" },

{ "code": "+972 # ### ####" },

{ "code": "+975 17 ### ###" },

{ "code": "+976 ## ## ####" },

{ "code": "+977 ## ### ###" },

{ "code": "+993 # ### ####" },

{ "code": "+20 ### ### ####" },

{ "code": "+211 ## ### ####" },

{ "code": "+212 ## #### ###" },

{ "code": "+213 ## ### ####" },

{ "code": "+218 21 ### ####" },

{ "code": "+221 ## ### ####" },

{ "code": "+233 ### ### ###" },

{ "code": "+235 ## ## ## ##" },

{ "code": "+240 ## ### ####" },

{ "code": "+242 ## ### ####" },

{ "code": "+243 ### ### ###" },

{ "code": "+244 ### ### ###" },

{ "code": "+249 ## ### ####" },

{ "code": "+250 ### ### ###" },

{ "code": "+251 ## ### ####" },

{ "code": "+253 ## ## ## ##" },

{ "code": "+255 ## ### ####" },

{ "code": "+256 ### ### ###" },

{ "code": "+260 ## ### ####" },

{ "code": "+261 ## ## #####" },

{ "code": "+264 ## ### ####" },

{ "code": "+265 # #### ####" },

{ "code": "+30 ### ### ####" },

{ "code": "+351 ## ### ####" },

{ "code": "+352 ### ### ###" },

{ "code": "+353 ### ### ###" },

{ "code": "+355 ### ### ###" },

{ "code": "+359 ### ### ###" },

{ "code": "+377 ### ### ###" },

{ "code": "+378 #### ######" },

{ "code": "+381 ## ### ####" },

{ "code": "+39 ### #### ###" },

{ "code": "+420 ### ### ###" },

{ "code": "+421 ### ### ###" },

{ "code": "+43 ### ### ####" },

{ "code": "+44 ## #### ####" },

{ "code": "+49 ### ### ####" },

{ "code": "+52 ### ### ####" },

{ "code": "+54 ### ### ####" },

{ "code": "+55 ## #### ####" },

{ "code": "+55 ## 7### ####" },

{ "code": "+57 ### ### ####" },

{ "code": "+58 ### ### ####" },

{ "code": "+590 ### ### ###" },

{ "code": "+593 ## ### ####" },

{ "code": "+595 ### ### ###" },

{ "code": "+598 # ### ## ##" },

{ "code": "+62 8## ### ####" },

{ "code": "+63 ### ### ####" },

{ "code": "+64 ### ### ####" },

{ "code": "+7 ### ### ## ##" },

{ "code": "+7 6## ### ## ##" },

{ "code": "+7 7## ### ## ##" },

{ "code": "+81 ## #### ####" },

{ "code": "+84 ### #### ###" },

{ "code": "+86 ### #### ###" },

{ "code": "+886 # #### ####" },

{ "code": "+90 ### ### ####" },

{ "code": "+91 #### ### ###" },

{ "code": "+92 ### ### ####" },

{ "code": "+962 # #### ####" },

{ "code": "+963 ## #### ###" },

{ "code": "+966 5 #### ####" },

{ "code": "+967 ### ### ###" },

{ "code": "+970 ## ### ####" },

{ "code": "+971 5# ### ####" },

{ "code": "+972 5# ### ####" },

{ "code": "+98 ### ### ####" },

{ "code": "+992 ## ### ####" },

{ "code": "+995 ### ### ###" },

{ "code": "+996 ### ### ###" },

{ "code": "+998 ## ### ####" },

{ "code": "+234 ### ### ####" },

{ "code": "+234 ### ### ####" },

{ "code": "+375 ## ### ## ##" },

{ "code": "+380 ## ### ## ##" },

{ "code": "+423 ### ### ####" },

{ "code": "+49 #### ### ####" },

{ "code": "+55 ## 9#### ####" },

{ "code": "+596 ### ## ## ##" },

{ "code": "+850 ### #### ###" },

{ "code": "+850 191 ### ####" },

{ "code": "+856 20## ### ###" },

{ "code": "+86 ### #### ####" },

{ "code": "+964 ### ### ####" },

{ "code": "+994 ## ### ## ##" },

{ "code": "+358 ### ### ## ##" },

{ "code": "+62 8## ### ## ###" },

{ "code": "+86 ## ##### #####" },

{ "code": "+850 #### #############" }

];

export default maskList;

Листинг mask\_list.js

import maskList from './mask\_list';

const mask = (selector) => {

function setMask(event) {

let matrix = '+###############';

const input = event.target;

if (Array.isArray(maskList)) {

maskList.forEach(item => {

let code = item.code.replace(/[\s#]/g, ''),

phone = input.value.replace(/[\s#-)(]/g, '');

if (phone.startsWith(code)) {

matrix = item.code;

}

});

} else {

console.error('maskList is not an array');

}

let i = 0,

val = input.value.replace(/\D/g, '');

input.value = matrix.replace(/(?!\+)./g, function (a) {

return /[#\d]/.test(a) && i < val.length ? val.charAt(i++) : i >= val.length ? '' : a;

});

}

let inputs = document.querySelectorAll(selector);

inputs.forEach(input => {

input.addEventListener('input', setMask);

input.addEventListener('focus', setMask);

input.addEventListener('blur', setMask);

});

};

export default mask;

Листинг mask.js