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 «Печатный центр»

**Исполнитель**

студент 2 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. В. Дребезов

подпись, дата

**Руководитель**

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Жиляк

(должность, уч. звание) (подпись, дата)

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

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

дата, подпись

Курсовой проект защищен с оценкой

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

подпись дата инициалы и фамилия

Минск 2022

**Содержание**

[Введение 4](#_Toc122376412)

[1 Постановка задач и обзор аналогичных решений 5](#_Toc122376413)

[1.1 Актуальность темы 5](#_Toc122376414)

[1.2 Обзор аналогичных решений 5](#_Toc122376415)

[1.3 Постановка задачи 9](#_Toc122376416)

[1.4 Выводы по разделу 9](#_Toc122376417)

[2 Проектирование веб-сайта 1](#_Toc122376418)0

[2.1 Выбор средств реализации 1](#_Toc122376419)0

[2.2 Проектирование страниц 1](#_Toc122376422)1

[2.3 Выбор стилевого оформления 1](#_Toc122376423)2

[2.4 Разработка пользовательских элементов 1](#_Toc122376424)3

[2.5 Выводы по разделу 1](#_Toc122376425)5

[3 Реализация структуры веб-сайты 1](#_Toc122376426)6

[3.1 Структура JSX-документа 1](#_Toc122376427)6

[3.2 Структура React-redux 2](#_Toc122376428)1

[3.3 Добавление таблиц стилей Sass 2](#_Toc122376431)1

[3.4 Выводы по разделу 2](#_Toc122376432)5

[4 Тестирование веб-сайта 2](#_Toc122376433)6

[4.1 Адаптивный дизайн веб-сайта 2](#_Toc122376434)6

[4.2 Кроссбраузерность веб-сайта 2](#_Toc122376435)7

[4.3 Тестирование компонентов 2](#_Toc122376436)9

[4.4 Выводы по разделу 2](#_Toc122376437)9

[5 Руководство пользователю 3](#_Toc122376438)0

[Заключение 3](#_Toc122376439)1

[Список использованных источников 3](#_Toc122376440)2

[Приложение А Листинг JSX-документа 3](#_Toc122376442)3

[Приложение Б Листинг (SASS) 3](#_Toc122376443)4

# 

# Введение

В настоящее время количество новых сайтов увеличивается с неимоверной скоростью. Именно поэтому, основной задачей, ставящейся при разработке сайта – сделать его максимально привлекательным для потенциальных пользователей и придать ему индивидуальность. На заре развития интернета можно было привлечь достаточно большое количество посетителей на сайт, не прикладывая к его разработке больших усилий и без особых материальных затрат. Сейчас же, в условиях возросшей конкуренции, функциональность сайта, его стиль и запоминаемость играют очень большую роль. Ведь независимо от того создаётся сайт для представления какой-либо фирмы или просто для заработка на рекламе, основной его функцией является привлечение как можно большей аудитории. Поэтому для достижения успеха важен правильный подход к выбору стратегии и исполнителей.

Профессиональные студии веб – дизайна предлагают создание сайтов различного предназначения, это может быть корпоративныый сайт, сайт – визитка, интернет – магазин, сайты для ведения бизнеса как онлайн, так и в реальном мире.

Наше время называют «информационным веком». Это название возникло потому, что самым важным, ценным и необходимым ресурсом является информация. Обладание информацией и умение своевременно, быстро и четко донести необходимую информацию до клиентов – ключ к успеху в реалиях современного бизнеса. В настоящий момент именно Интернет, способен оперативно и массово передавать текст, звук, изображения и даже видео-ролики, т.е. практически любую информацию. C точки зрения пользователя, Интернет – это огромный информационный ресурс, в котором можно найти все, что угодно: от прогноза погоды до личных предпочтений голливудских кинозвезд. Аудитория сети Интернет растет с каждым годом. Интернет – самое быстроразвивающееся средство передачи информации за все историю человечества. Пользователи сети интернет являются целью особого интереса для рекламодателей, потому что факт остается фактом – среди пользователей сети куча потенциальных клиентов. Актуальность темы сайта печатного центра в современном мире обусловлена тем, не только тем, что печатная продукция остается одной из самых продоваемых, но тем, что сегодня все больше людей предпочитают получать информацию и оформлять заказы через интернет, а не в традиционных центрах. Это связано с тем, что на сайте доступна вся актуальная информация и заказчику нет нужды лишний раз звонить в справочный цент или идти туда самому, что бы узнать информацию или оформить заказ.

Цель курсовой работы: разработать сайт печатного центра с использованием React JS, а также с применением Sass/CSS3.

Задачи курсовой работы:

* проанализировать и описать основные принципы работы React и Redux;
* разработать приложение, используя React и Redux;
* провести тестирование приложения;
* провести анализ и оценку качества приложения.

# 1 Постановка задач и обзор аналогичных решений

## Актуальность темы

Актуальность темы заключается в том, что печатная продукция остаётся одним из самых продаваемых товаров в мире. Как для покупателей важно, чтобы было удобно и быстро сделать заказ с нужными параметрами, так и для продавцов важно, чтобы было удобно и быстро обрабатывать заказы и получать корректную информацию о заказе клиента.

Сегодня на рынке существует множество печатных центров, которые используют различные технологии и платформы для создания своих сайтов. Многие из них представляют собой лишь сайты-визитки, но существуют и более продвинутые решения, использующие самый популярный фреймворк – React JS.

## 1.2 Обзор аналогичных решений

Веб-сайт – это набор взаимосвязанных веб-страниц, которые представляют собой информацию, предназначенную для просмотра в браузере. Веб-сайт представляет собой совокупность веб-страниц, которые могут быть связаны между собой посредством гипертекстовых ссылок.

Для создания быстрого и динамического веб-сайта необходимо использовать современные технологии, которые позволяют создавать веб-сайт с минимальными затратами времени и средств. Веб-сайт должен быть удобным для пользователя, а также должен быть адаптирован под мобильные устройства. Тема веб-сайта должна быть востребованной, а также должна быть актуальной. Поэтому было принято решение разработать сайт печатного центра, который будет представлять собой веб-сайт, на котором можно будет оформить заказ на доступную продукцию. Прежде чем приступить к разработке веб-сайта, необходимо проанализировать аналоги, чтобы понять, какие технологии используются для создания веб-сайтов, а также какие есть недостатки и преимущества веб-сайтов аналогов. И самое главное найти в каждом аналоге свои уникальные особенности, которые будут отличать этот сайт от других. Для этого будет рассмотрены 3 аналога.

Аналог №1 – сайт Карандаш [1].

Карандаш является одним из самых крупных печатных центров в Беларуси. Основным преимуществом сайта является его удобство и простота использования. На главной странице сайта можно увидеть разделы, которые помогут пользователю найти нужный товар. На рисунке 1.1 представлено оформление сайта. Также все товары разделены по категориям, что позволяет пользователю найти нужный товар быстрее. Веб-сайт очень простой и удобный для пользователя, а также имеет красивый дизайн. Шрифт читабельный. Сайт динамический, интерактивный. Также есть анимация при наведении на картинки. Сайт использует три основных цвета (белый, серый, нежно коричневый), что положительно сказывается на общем восприятии сайта. На этом сайте есть уникальная функция, которая позволяет пользователю сравнить товары, которые он хочет купить. Ещё можно отметить, что на сайте есть режим для слабовидящих, который позволяет пользователю увеличить шрифт и контрастность. Также можно добавить товары в избранное, чтобы не потерять их, причем без регистрации.

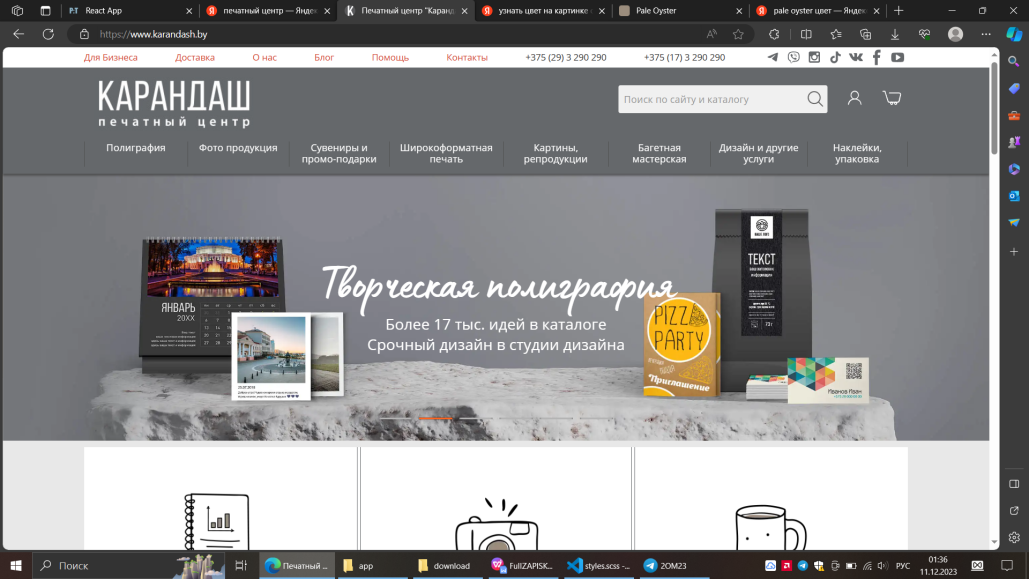


Рисунок 1.1 – Главная страница сайта Карандаш

Веб-сайт имеет адаптивный дизайн, который позволяет использовать сайт на мобильных устройствах, горизонтальное меню становится выпадающим, текст остаётся читабельным. Дизайн представлен на рисунке 1.2.

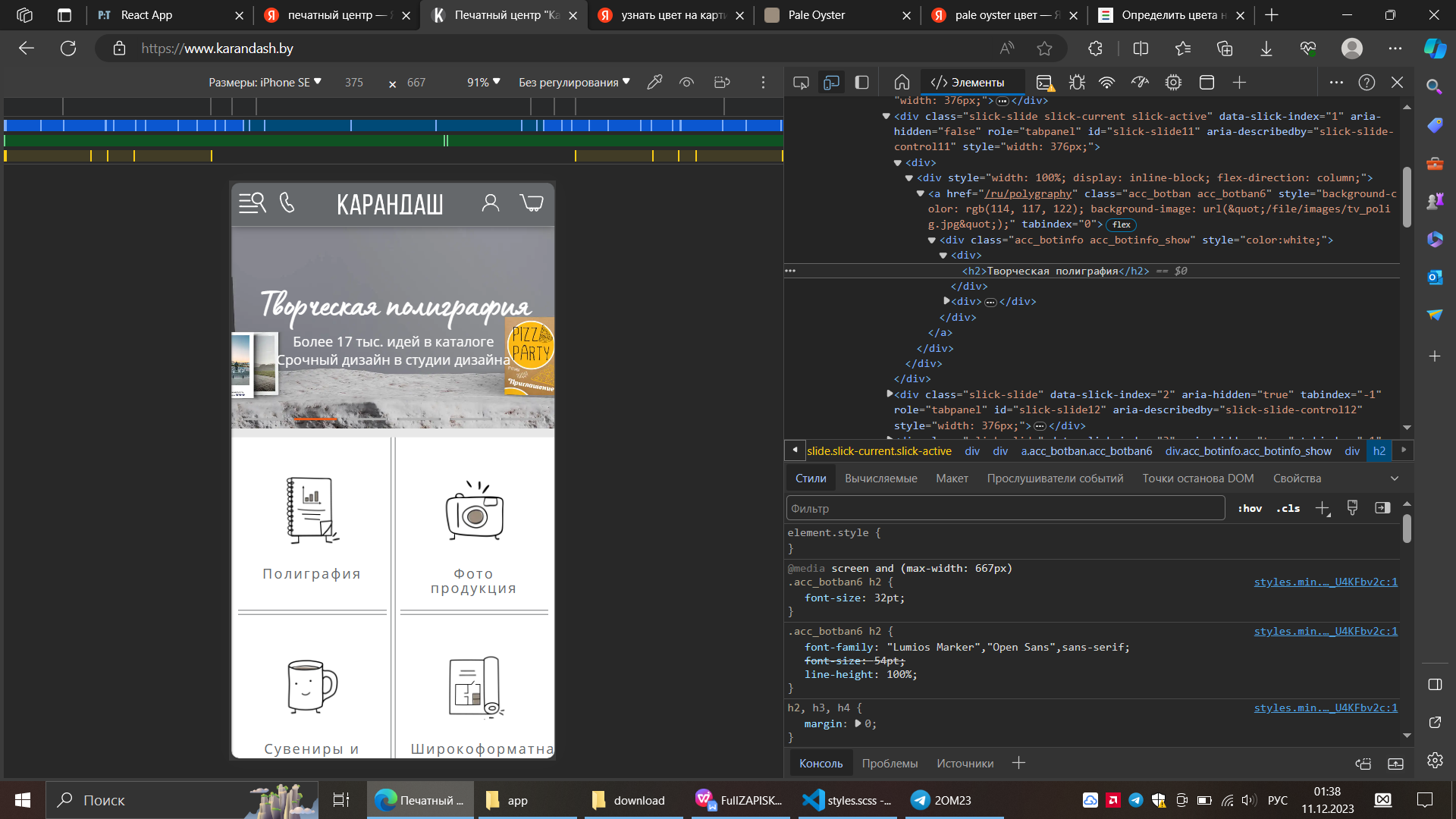


Рисунок 1.2 – Мобильная версия сайта Карандаш

Из минусов данного веб-сайта можно выделить то, что он долго загружается из-за большого количества товаров, которые находятся на сайте. Данный веб-сайт был разработан с помощью HTML, CSS, JavaScript, JQuery, React, MySQL.

Аналог №2 – сайт Печатник [2].

Следующим аналогом будет сайт Печатник, который является крупным печатным центром в Беларуси. На сайте можно оформить заказ на разную продукцию от фотографий до книг в переплёте. На сайте есть разделы, в которых можно выбрать нужный товар. Веб-сайт очень удобен для пользователя, так как он имеет простой и понятный интерфейс. Также пользователь может быстро найти нужный товар, так как на сайте есть поиск по названию товара и по категориям. Все товары имеют подробное описание. Дизайн простой и использует черные и белые цвета с небольшими вкраплениями розового, голубого и желтого. Дизайн представлен на рисунке 1.3. Особенностью сайта является то, что но позволяет добавлять товары в избранное, а не сразу оформлять заказ.

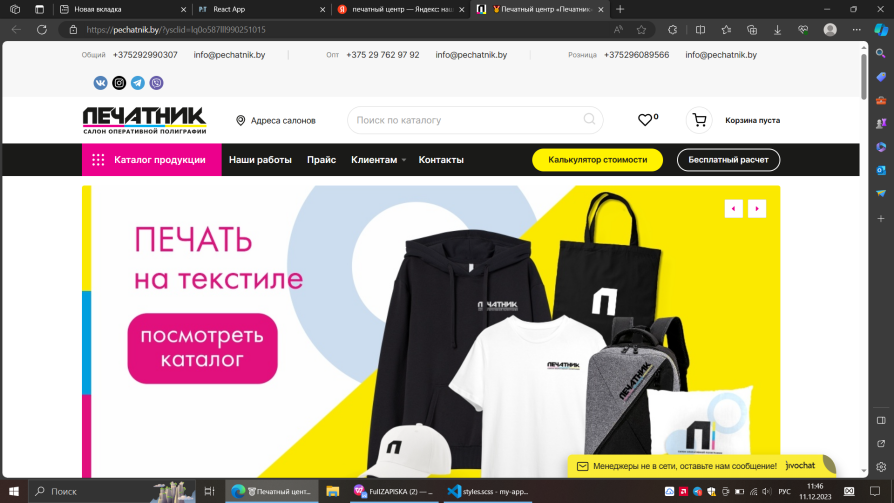


Рисунок 1.3 – Главная страница сайта Печатник

Также сайт имеет адаптивный дизайн, который позволяет просматривать сайт на мобильных устройствах. У данного сайта нет никаких проблем с адаптивностью, на разных устройствах сайт выглядит привлекательно. На рисунке 1.4 представлено оформление на мобильных устройствах.

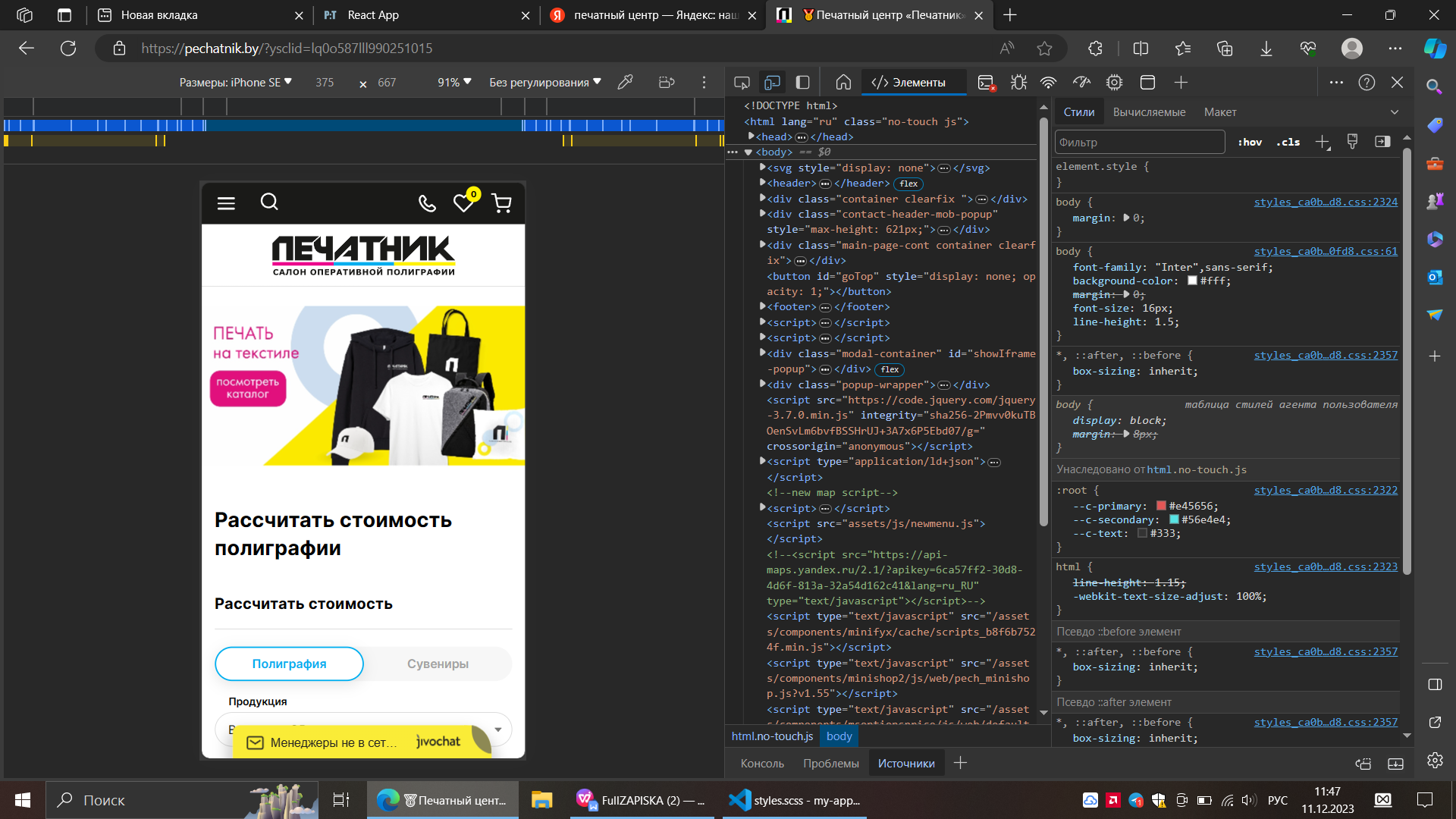


Рисунок 1.4 – Мобильная версия сайта Печатник

Из недостатков можно выделить то, что некоторые блоки сайта плохо подогнаны по размеру страницы из-за чего приходится лишний раз использовать прокрутку. Данный сайт был разработан на таких технологи как HTML, CSS, JavaScript, MySQL.

Аналог №3 – сайт Санпринт [3].

Санпринт - крупный печатный центр Беларуси. На сатйе можно оформить заказ на разную полиграфическую продукцию. Но главным образом сайт наполнен информацией о центре и контактами и не позволяет оформлять заказ сразу на несколько товаров или добавлять их в корзину. Дизайн выполнен в простом стиле с использованием простых форм в серых и белых тонах. Сайт был реализован следующими технологиями HTML, CSS, JavaScript, MySQL. На рисунке 1.5 представлено оформление сайта Aliexpress.

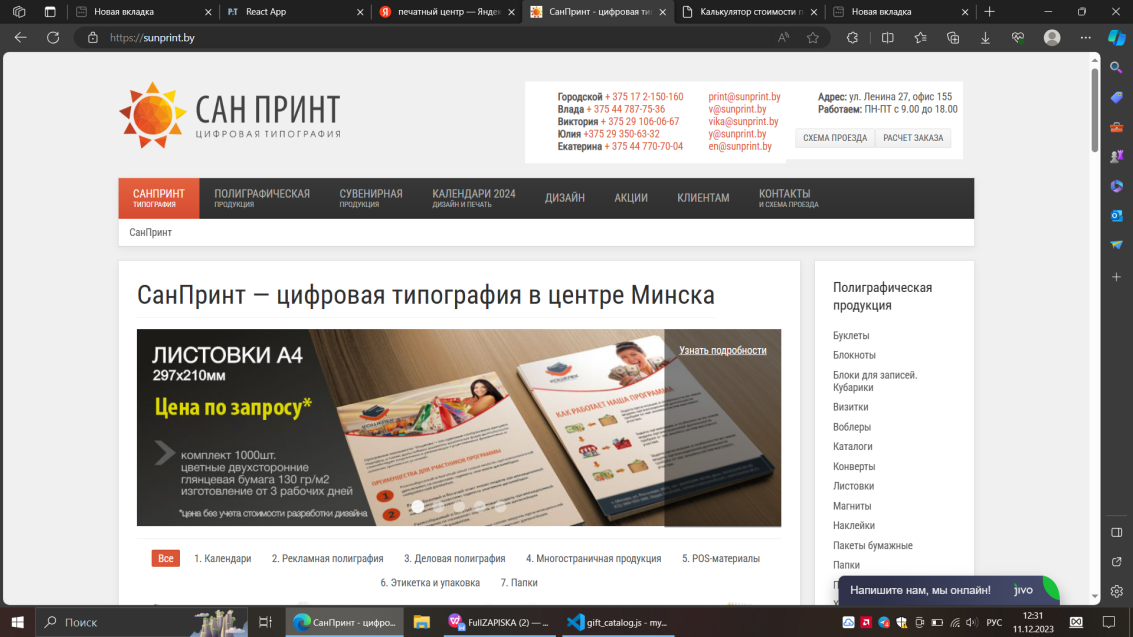


Рисунок 1.5 – Главная страница сайта Санпринт

Также данный сайт кроссплатформенный, а также адаптирован под мобильные устройства. На рисунке 1.6 представлено оформление на мобильных устройствах.

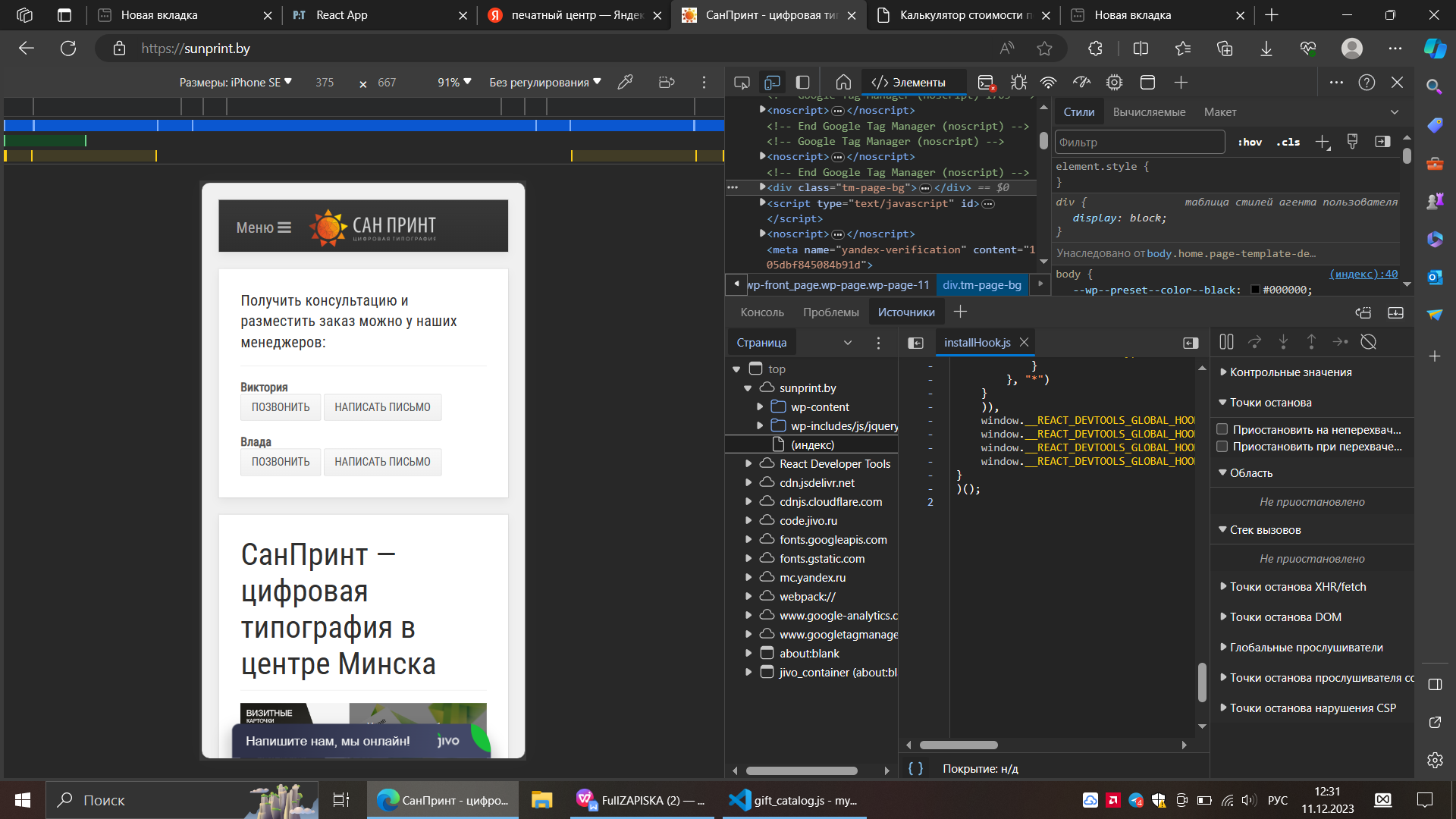


Рисунок 1.6 – Мобильная версия сайта Санпринт

И таким образом, было проанализировано 3 сайта, которые являются аналогами нашего будущего сайта. Основные преимущества и недостатки этих сайтов были выделены, а также были выделены технологии, которые используются для создания сайтов. Можно сделать вывод, что сайт, который будет создан в рамках данной работы, будет иметь следующие преимущества:

* простой и удобный дизайн;
* яркие и не раздражающие глаза цвета;
* кроссплатформенность;
* адаптивность под мобильные устройства;
* удобная навигация по сайту.

## Постановка задачи

Основные задачи для интернет-магазина:

* разработать возможность добавления и удаления товаров из корзины;
* разработать структуру веб-сайта;
* разработать возможность заказа товара;
* разработать возможность сортировки товаров по параметрам
* Разработать возможность выбора параметров для определённого товара

## 1.4 Выводы по разделу

В данном разделе была подчёркнута необходимость создания сайта печатного центра, а также были выделены основные преимущества и недостатки существующих сайтов. Были выделены технологии, которые используются для создания сайтов. Были выделены основные задачи для сайта печатного центра.

# 2 Проектирование веб-сайта

## 2.1 Выбор средств реализации

Главной задачей сайта является предоставление информации о товарах, которые можно заказать. Данный сайт будет предназначен для людей, которые хотят заказать различные товары. Сайт будет адаптирован под мобильные устройства, а также будет кроссплатформенным.

Интернет-магазин будет разработан в текстовом редакторе Visual Studio Code (1.85.0). Страницы сайта могут просматриваться в таких браузерах как Internet Explorer, Microsoft Edge, Safari, Opera, Mozilla Firefox, Google Chrome.

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга.

Для реализации будущего сайта будет использована следующая архитектура:

* Front-end.
* Back-end.

Front-end — это часть сайта, которая видна пользователю. В данном случае это будет внешний вид сайта, его дизайн, а также навигация по сайту. В данном случае будет использованы следующие технологии: React JS (18.2.0), SASS (4.13.0), Redux-Toolkit (1.3.5), React Router (5.1.2).

JavaScript – это язык программирования, который используется для создания динамических веб-страниц. JavaScript позволяет создавать интерактивные веб-страницы, которые могут реагировать на действия пользователя.

React JS — это библиотека JavaScript, предназначенная для разработки пользовательских интерфейсов. В отличие от других библиотек JavaScript React JS использует компонентный подход, который позволяет разрабатывать сложные интерфейсы из простых компонентов. С помощью React разработчики создают веб-приложения, которые изменяют отображение без перезагрузки страницы. Благодаря этому приложения быстро реагируют на действия пользователя, например, заполнение форм, применение фильтров, добавление товаров в корзину и так далее. React применяют для отрисовки компонентов пользовательского интерфейса. Также библиотека может полностью управлять фронтендом. В этом случае React используют с библиотеками для управления состоянием и роутинга, например, Redux и React Router.

SASS — это расширение CSS, которое добавляет в CSS мощные возможности, такие как переменные, вложенные правила, импорт и многое другое. Но самое главное, что SASS позволяет использовать препроцессоры, которые позволяют писать более краткий и лаконичный код.

Redux — это библиотека для управления состоянием приложения.

Redux Toolkit — это набор инструментов для упрощения работы с Redux. Redux Toolkit позволяет создавать более простые и понятные редьюсеры, а также упрощает работу с асинхронными запросами.

React Router — это библиотека для маршрутизации в React-приложениях. React Router позволяет создавать маршруты, которые будут отображаться в зависимости от URL-адреса. Благодаря React Router можно создавать SPA (Single Page Application), которые не перезагружают страницу при переходе по ссылкам.

Back-end - это часть приложения, которая отвечает за обработку данных, которые приходят от клиента и возвращают данные клиенту. Также Back-end отвечает за работу с базой данных, которая хранит данные приложения. Для создания Back-end будет использованы следующие технологии: Node.js (12.16.1).

Node.js — это JavaScript-окружение, которое позволяет запускать JavaScript-код на сервере. Node.js позволяет создавать сервер, который будет обрабатывать запросы от клиента и возвращать данные клиенту.

Для передачи данных между клиентом и сервером будет использоваться формат JSON.

JSON — это текстовый формат обмена данными, который основан на JavaScript. JSON позволяет передавать данные между клиентом и сервером. JSON позволяет передавать данные в виде объектов, массивов, строк, чисел, логических значений и null.

## 2.2 Проектирование страниц

В данном проекте будет использована Flexbox-верстка (Flexbox), так как она позволяет создавать адаптивные веб-страницы, которые будут отображаться корректно на всех устройствах. С помощью этой технологии можно очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров. CSS Flexbox поддерживается всеми используемые на сегодняшний момент современными браузерами (с использованием префиксов: IE10+, Edge12+, Firefox 2+, Chrome 4+, Safari 3.1+, Opera 12.1+, iOS Safari 3.2, Opera mini, Android 2.1+).

Исходя из выбора способа верстки были расположены все основные элементы сайта. На сайте будут представлены следующие страницы:

* + Главная
  + Полиграфическая продукция
  + Сувенирная продукция
  + Страницы товара
  + Контакты
  + Требования к оформлению
  + Форматы в полиграфии
  + Корзина
  + Форма оформления заказа

На всех страницах будет присутствовать хэдер с меню для перемещения по 7 главным страница:

* + Главная
  + Полиграфическая продукция
  + Сувенирная продукция
  + Контакты
  + Требования к оформлению
  + Форматы в полиграфии
  + Корзина

На главной странице будет располагаться логотип компании и краткая информация о сайте. В меню в хэдере кнопки для перехода на “Главную” и “Корзину” будут скрыты.

На “Полиграфическая продукция” будет каталог с товара из категории полиграфической продукции. При нажатии на товар можно попасть на его страницу.

На “Сувенирная продукция” будет каталог с товара из категории сувенирной продукции. Сбоку будет меню с параметрами для фильтрации товаров по подкатегориям и формату При нажатии на товар можно попасть на его страницу.

На “Странице товара” будет доступно меню для выбора параметров заказа, кнопка добавления товара в корзину и таблица с расценками.

Страницы “Требования к оформлению” и “Форматы в полиграфии” будет представлена текст с информацией, которая может пригодится заказчику для корректного оформления заказа.

Страница “Контакты” содержит карточки работников с их именами, номерами телефонов и электронными почтами

Страница “Корзина” содержит список товаров выбранных для оформления. Каждый товар можно удалить отдельно, либо очистить карзину целиком

Страница “Форма оформления заказа” содержит поля для заполнения личной информации и прикрепления файлов для оформления заказа.

## 2.3 Выбор стилевого оформления

Для реализации стилевого оформления сайта было выбрано минималистическое направление. Стиль передаёт серьёзность фирмы, а яркие элементы будут подчерчивать и выделять главные элементы. Акцент на сайте ориентируется на основной контент.

Сайт будет состоять из 3-х основных цветов – светло серого, белого, голубого. Белый будет основным цветом, о серый - выделять основные элементы, такие как меню. Голубой цвет контрастирует с монохромным оформление, чем будет выделять кнопки, но при этом будет передавать строгость и серьезность.

В данном проекте будут использован один основной шрифт: Lato Regular. Шрифт подключался хранится рядом со стилями в папке fonts.

## Шрифт Lato Regular подходит для данного сайта, так как у него нет засечек, за счёт своей формы он будет хорошо читать в любом размере и сочетаться с любым элементом, как в статье, так и на кнопке. Так же он поддерживает utf-8.

## 2.4 Разработка пользовательских элементов

Разработка логотипа осуществлялась в редакторе для векторной графики LibreOffice. Было принято решение создать минималистичный логотип из минимума деталей. Он состоит из повёрнутого на бок обозначения типографского пункта “PT” и названия фирмы “Polygraphy Team”. “Team” выделено тест-цветами, что ещё раз указывает на направление фирмы и показывает дружный настрой. Логотип представлен на Рисунке 2.1.

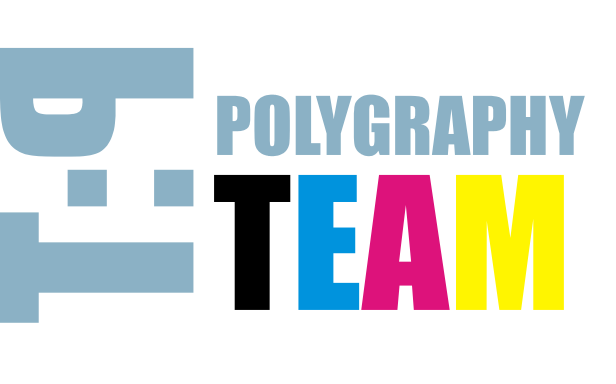


Рисунок 2.1 – Логотип сайта Polygraphy Team

В проекте будет использована иконка сайта, которая будет отображаться во вкладке браузера. Иконка будет иметь формат .svg и будет иметь размер 32x32 пикселя. Иконка будет создана в программе LibreOffice. Иконка представлена на рисунке 2.2.

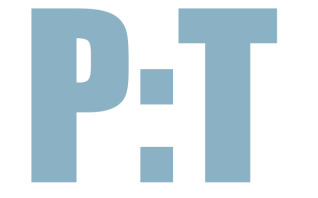


Рисунок 2.2 – Иконка сайта Polygraphy Team

В проекте будут пользовательские элементы, которые будут отвечать за взаимодействие пользователя с сайтом. Это будут следующие элементы: кнопки, модальные окна, табы, пагинация, формы и т.д. Пример кнопки будет на рисунке 2.3.

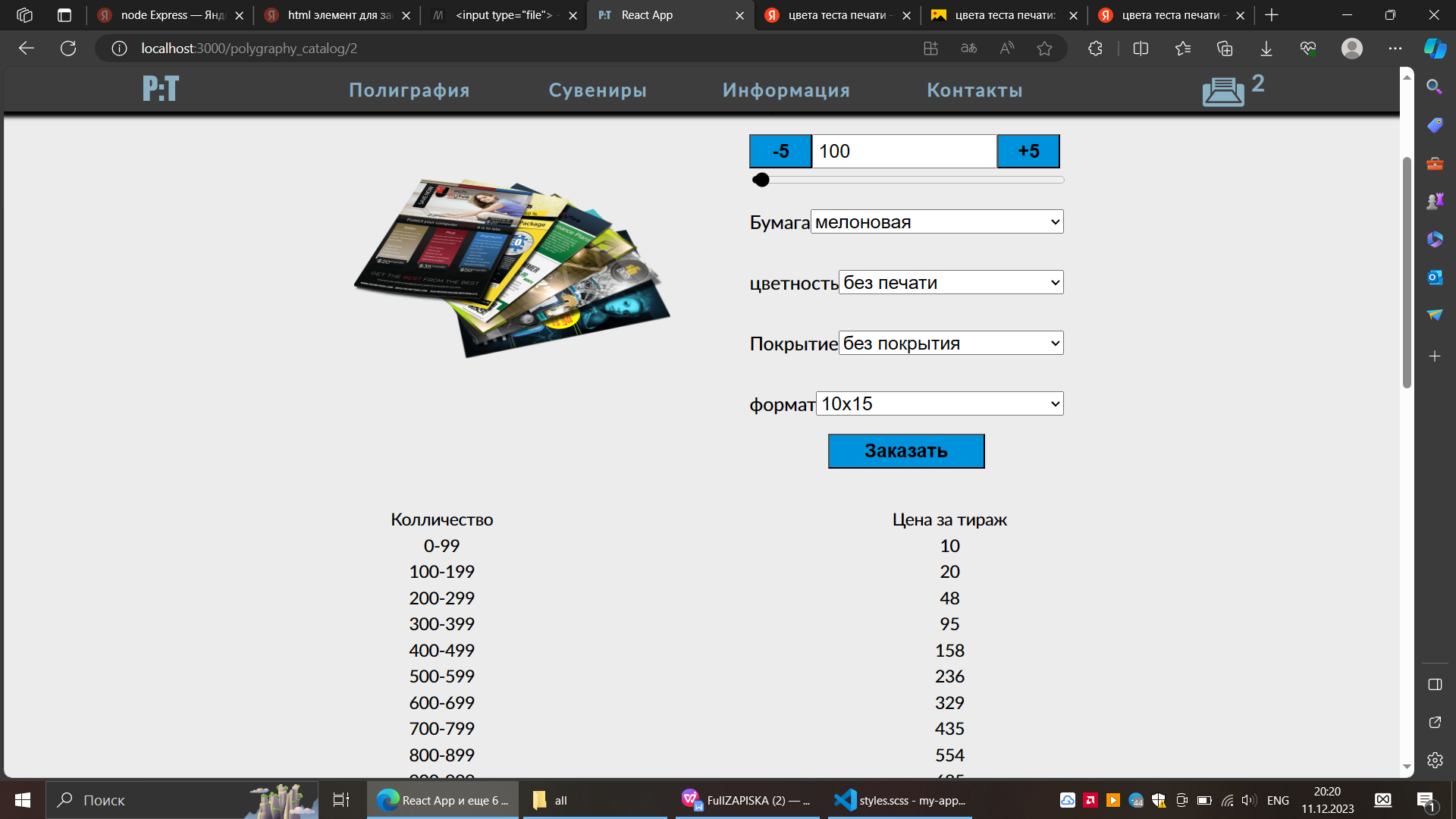


Рисунок 2.3 – Кнопка сайта Polygraphy Team

На сайте будут карточки товара, которые будут состоять из картинки и кнопок «Заказать». Полями для ввода количества тиража, который можно настроить через кнопки или ввести самому, при недопустимом тираже, кнопка «Заказать» отключается. Так же есть поля для выбора характеристик товара. Пример карточки товара будет на рисунке 2.4.

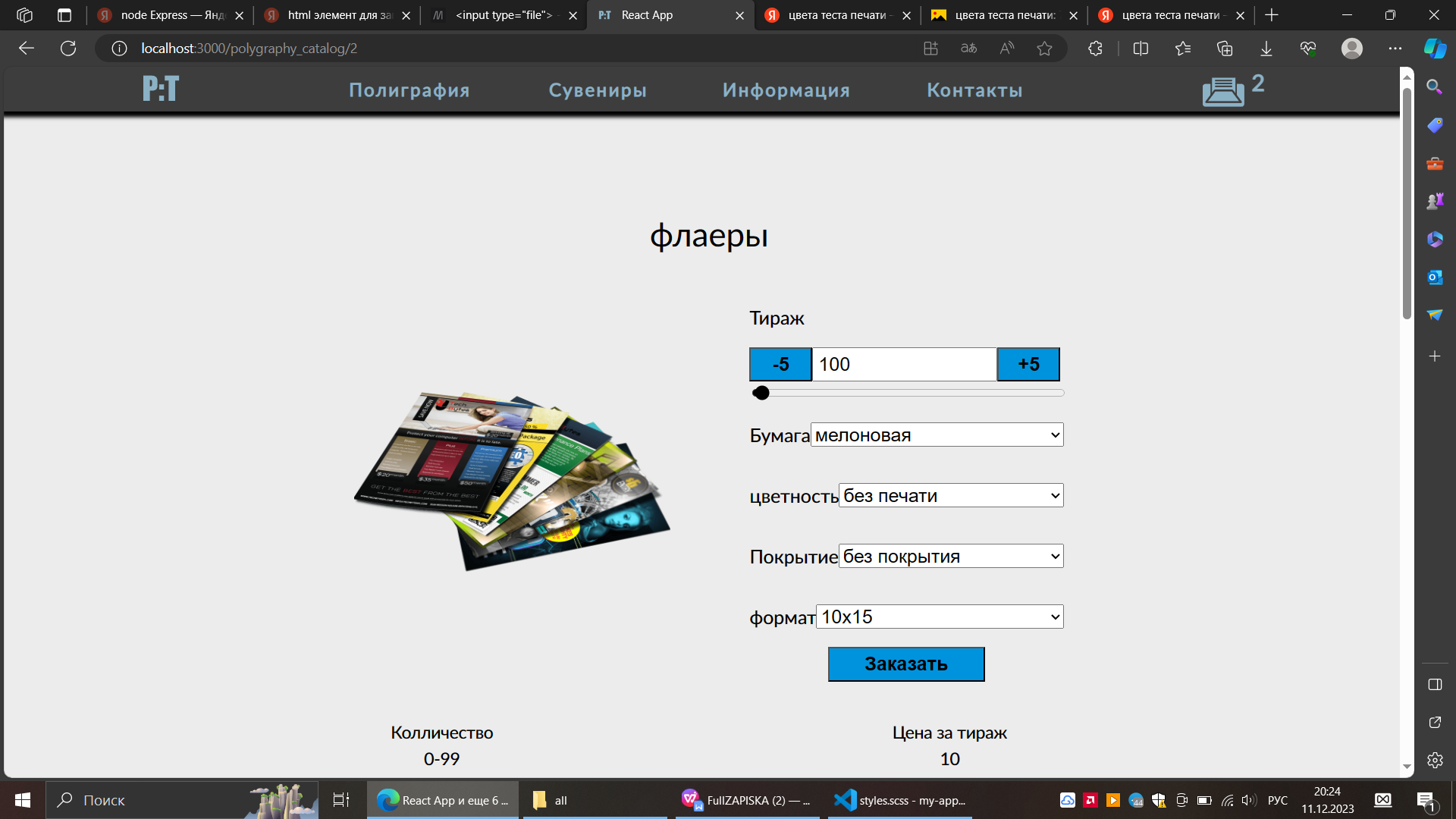


Рисунок 2.4 – Карточка товара

Также на сайте будет блок с фильтром. Пользователь сможет искать продукцию по категории и формату. Пример блока поиска товара будет рисунке 2.5.

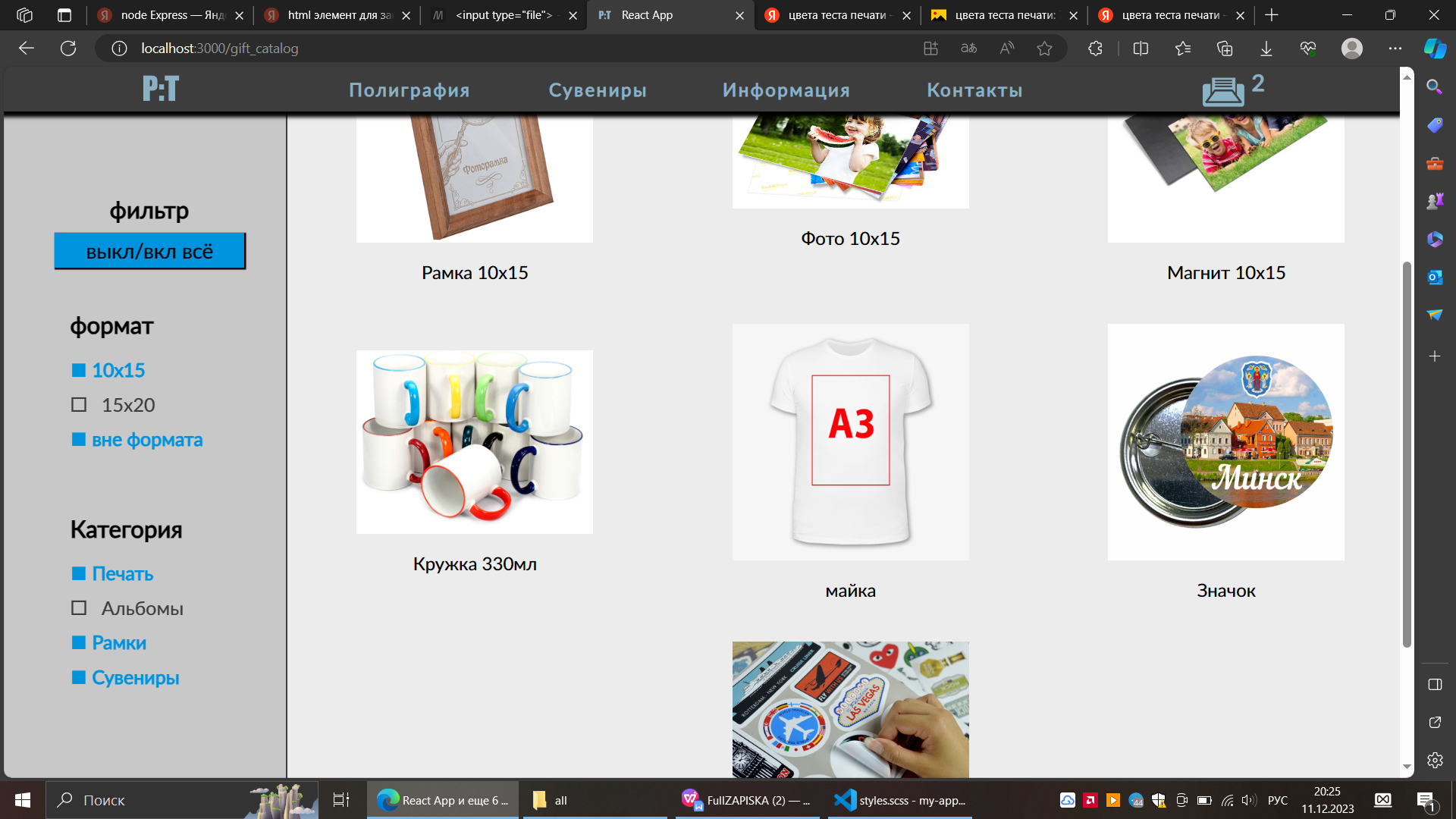


Рисунок 2.5 – Блок фильтра

Следующим основным элементом сайта будет блок корзины. Пользователь сможет добавлять продукты в корзину, удалять их из корзины, оформлять заказы. Пример блока корзины будет рисунке 2.6.

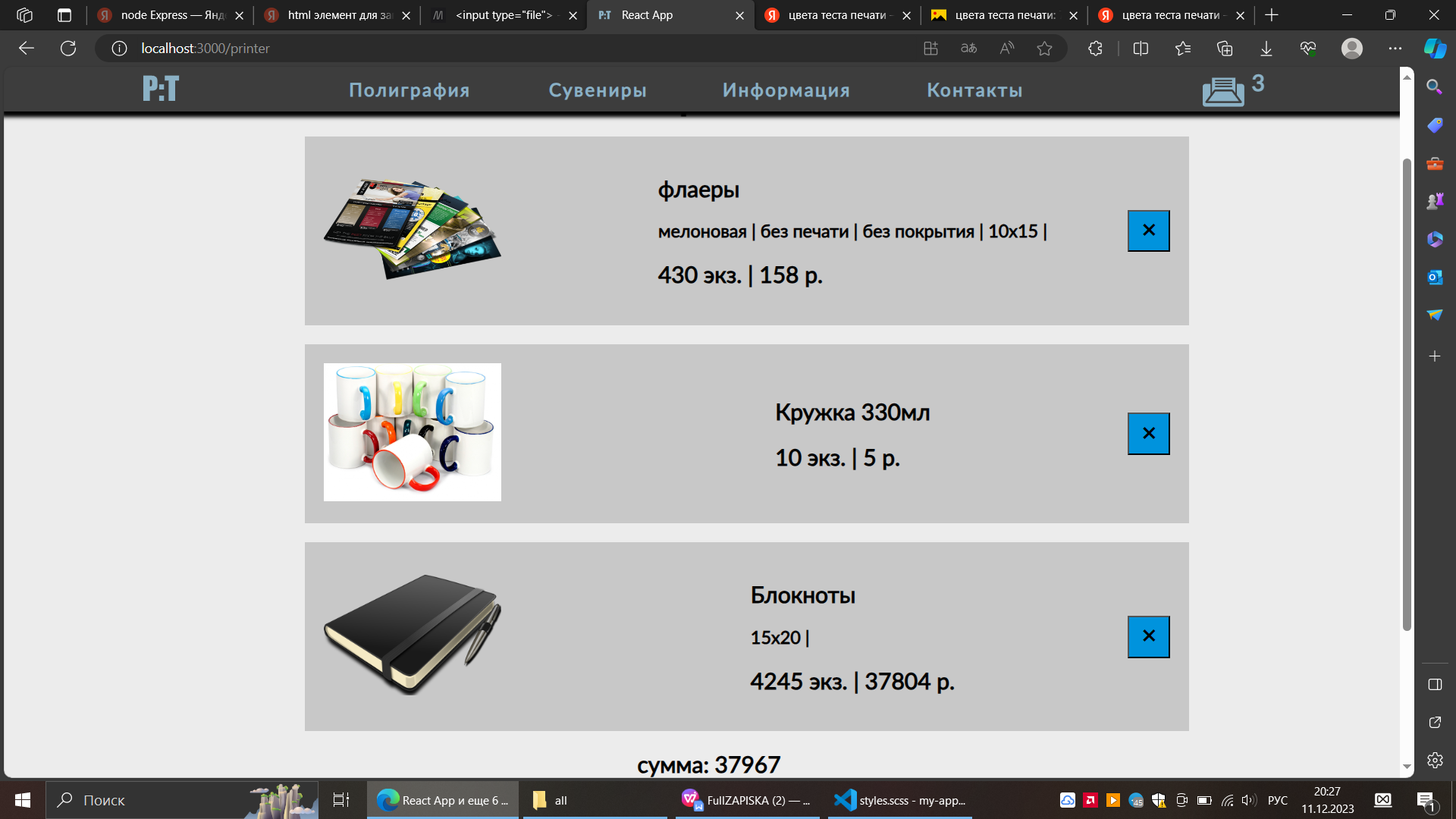


Рисунок 2.6 – Блок корзины

На сайте будут разработаны некоторые динамические эффекты и анимации, которые позволят придать сайту интерактивность и сделают его более привлекательным для пользователя. Анимация будет при наведении на кнопки.

Элемент Navbar будет состоять из логотипа сайта, навигационного меню. В навигационном меню будет 6 пункта: Главная(логотип), Полиграфия, Сувениры, Информация, Контакты, Корзина(Иконка принтера). Над корзиной будет висеть цифра с количеством товаров в корзине. На рисунке 2.7 представлено оформление Navbar.

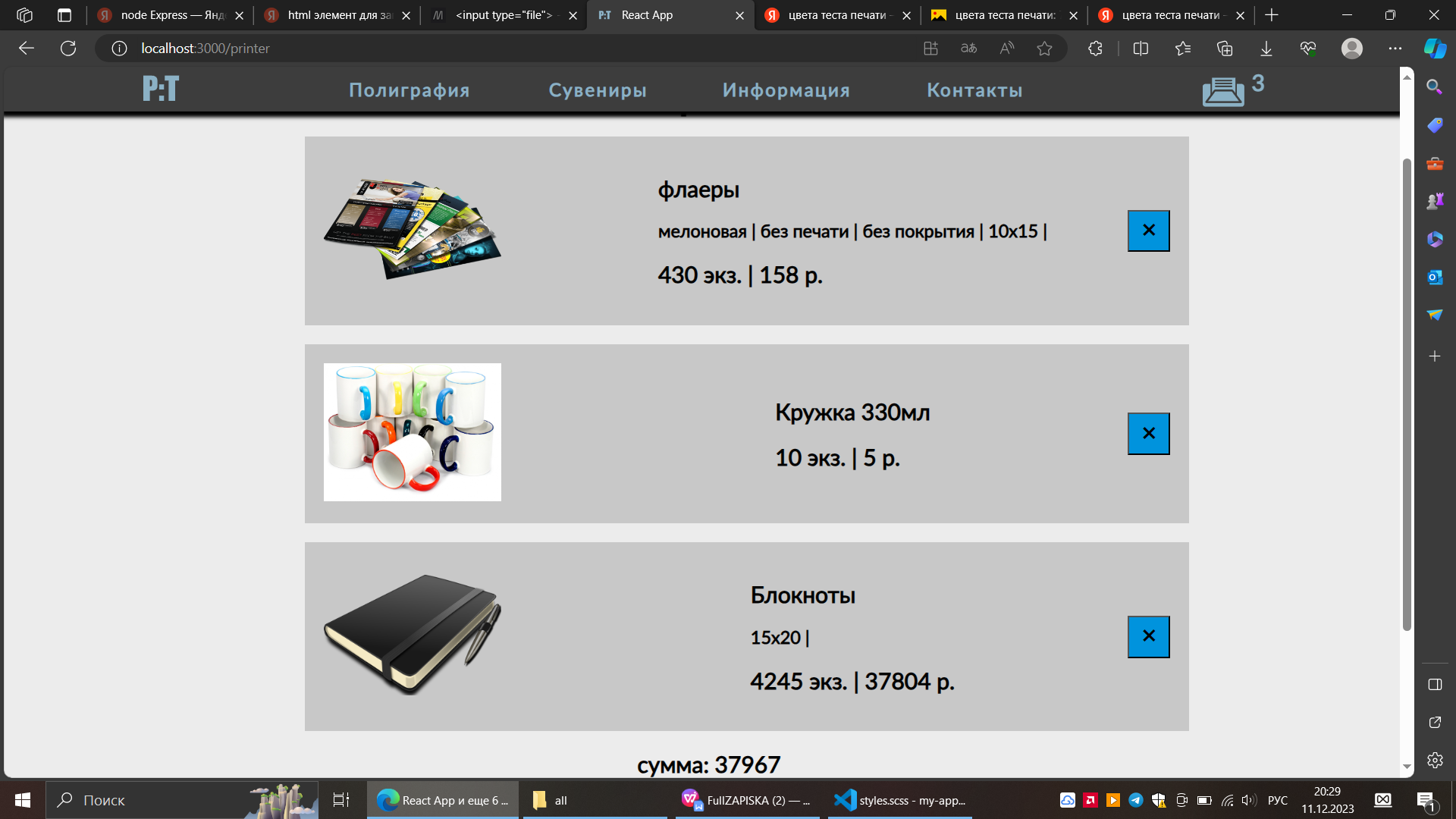


Рисунок 2.7 – Navbar сайта Polygraphy Team

На данном этапе были созданы макеты сайта, которые будут использоваться для дальнейшей разработки сайта. Были выбраны основные цвета и шрифты, которые будут использоваться на сайте. Были разработаны некоторые динамические эффекты и анимации, которые позволят придать сайту интерактивность и сделают его более привлекательным для пользователя. Самое главное, чтобы сайт был удобным для пользователя и отвечал его потребностям.

## 2.5 Выводы по разделу

Подводя итоги этого раздела можно сказать, что были разработаны макеты сайта, которые будут использоваться для дальнейшей разработки сайта. Были выбраны основные цвета и шрифты, которые будут использоваться на сайте. Были разработаны некоторые динамические эффекты и анимации, которые позволят придать сайту интерактивность и сделают его более привлекательным для пользователя. Самое главное, чтобы сайт был удобным для пользователя и отвечал его потребностям.

# 3 Реализация структуры веб-сайты

## 3.1 Структура JSX-документа

Для реализации проекты было принято решение создавать разные компоненты. Компонент — это функция, которая возращает JSX формат.  JavaScript XML (JSX) — это расширение синтаксиса JavaScript, которое позволяет использовать HTML-подобный синтаксис для описания структуры интерфейса. Как правило, компоненты написаны с использованием JSX, но также есть возможность использования обычного JavaScript. Код примера JSX-формата показан в листинге 3.1.

|  |
| --- |
| function App() {    const url = window.location.href;  const hasHello = String(!/http:\/\/.\*\/..\*/.test(window.location.href))  return (  <div className="App">  <a name="top"></a>    <header class={hasHello}>  <div>  <a href="/"><img class="logo" src="./pics/all/logo\_mini.svg" alt="img"/></a>  <nav>  <ul>  <li class="menu-item"><a href="/polygraphy\_catalog">Полиграфия</a></li>  <li class="menu-item"><a href="/gift\_catalog">Сувениры</a></li>  <li class="menu-item"><a >Информация</a>  <ul class="submenu">  <li><a href="/requirements">Требования к макетам</a></li>  <li><a href="/about\_formats">Форматы в полиграфии</a></li>  </ul>  </li>  <li class="menu-item"><a href="/contacts">Контакты</a></li>  </ul>  </nav>  <a href="/printer" class={"printIcon"}><img class="logo" src="./pics/all/printer.svg" alt="img"/><h2 class={useSelector(state => state.storeAmount)}>{useSelector(state => state.storeAmount)}</h2></a>  </div>  </header>  <article>  <div class="container">  </div>  </article>  <footer>  <h3>Ivan Drebezov | FIT | ISIT - 2 | 2023</h3>  </footer>  </div>  );  }  export default App; |

Листинг 3.1 – Структура JSX-формата

Самым главным компонентом будет с названием файлом и функции App. Он будет содержать основные маршруты по сайту. Для этого нужно использовать пакет React-router-dom. Почему лучше использовать его, чем обычный тег <a>? Потому что React это SPA. SPA (single page application) – это веб-приложение, работающее на одной странице. Оно подгружает все необходмые javascript и css файлы при первой загрузке страницы, а затем все общение между клиентом и сервером сводится к минимуму. Т.е. при таком подходе большая часть работы сайта производится на стороне клиента, а если нужно получить данные с сервера, то это обычно делается с помощью JSON. Код примера маршрутов будет показан в листинге 3.2.

|  |
| --- |
| <Routes>  <Route path="/" element={<Home/>} exact />  <Route path="/polygraphy\_catalog" element={<Polygraphy\_catalog/>} exact/>  <Route path="/polygraphy\_catalog/:id" element={<Item\_card/>} exact/>  <Route path="/gift\_catalog" element={<Gift\_catalog/>} exact />  <Route path="/gift\_catalog/:id" element={<Gift\_card/>} exact />  <Route path="/printer" element={<Printer/>} exact />  <Route path="/requirements" element={<Requirements/>} exact />  <Route path="/about\_formats" element={<About\_formats/>} exact />  <Route path="/contacts" element={<Emploers\_catalog/>} exact />  <Route path="/form" element={<Order\_Form/>} exact />  </Routes> |

Листинг 3.2 – Структура маршрутов

Следующим компонентом будет Home компонент. Он будет содержать лого и краткую информацию о сайте. Код страницы будет показан в листинге 3.3.

|  |
| --- |
| function Home() {  return (  <div class={"Home"}>  <article>  <div class="container">  <div class="logo">  <img src="pics/all/logo\_full.svg"/>  </div>  <div class="about">  <h1>Типография "Polygraphy Team"</h1>  <p>  Polygraphy Team – самая современная и высокотехнологичная  книжно-журнальная концептуальная типография XXI века.</p>  </div>  </div>  </article>  </div>  );  } |

Листинг 3.3 – Структура Home

Следующим компонентом будет Polygraphy\_catalog компонент. В нём будет находится каталог товаров представленных на станице “Полиграфия”. По нажатию на товар можно попасть на страницу товара. Код каталога полиграфии будет показан в листинге 1.4.

|  |
| --- |
| class Polygraphy\_catalog extends React.Component  {    constructor(props) {  super(props);  this.state={myarray:item\_poly\_arr}  this.render =this.render.bind(this)  }    render()  {  let rows=[];  let iter=0;  for(const item of this.state.myarray)  {  rows.push(  <Poly\_Node  index={item.index}  goodsname={item.iName}  key={item.iName}  />)  }  return (  <div class = {"polygraphy\_catalog"}>  <h1>Полиграфическая продукция</h1>  <div class={"market"}>{rows}</div>  </div>  )  }  }  class Poly\_Node extends React.Component  {  constructor(props)  {  super(props)  this.props={goodsname:String(),index:String()}  }  render()  {  return (  <div class={"poly\_item\_card"}><a href={`/polygraphy\_catalog/${this.props.index}`}>  <div>  <img class="poly\_item" src={`./pics/polygraphy/item${this.props.index}.png`}/>  <p>{this.props.goodsname}</p>  </div>  </a></div>)  }  } |

Листинг 3.4 – Структура кода компонента Polygraphy\_catalog

После нажатия на товар, на странице будет использоваться компонент Item\_card. На странице будет картинка, название товара, поля для набора тиража и параметров товара и кнопка для добавления товара в корзину. Код Item\_card показан в листинге 3.5.

|  |
| --- |
| <div class={"itemPage"}>  <p id={"name"}>{name}</p>  <div class={"actSection"}>  <div class={"image"}>  <img class={"poly\_item"} id={"pic"} src={`./pics/polygraphy/item${index}.png`}/>  </div>  <div class={"action"}>  <Draw check={this.checkInput} DefVal={100} minVal={50} maxVal={5000} stepAmount={5}/>  {list}    <input onClick={this.props.getItem} type={"button"} value={"Заказать"} disabled={this.state.buttonDes}/>  </div>  </div>  <div class={"infoSection"}>  {/\* {ListInfo[this.props.id-1]} \*/}  {createPriceTable(item\_poly\_arr[this.props.id-1].basePrice, item\_poly\_arr[this.props.id-1].discount)}  </div>  </div> |

Листинг 3.5 – Структура код Item\_card

Следующим компонентом будет Printer компонент. В нем будет находится товары, которые были добавлены в корзину. Также будет находится блок с общей суммой товаров в корзине и кнопкой оформления заказа. На товаре будут кнопкf удалить из корзины. Код корзины будет показан в листинге 3.6.

|  |
| --- |
| <div class={"printerPage"}>  <h1>Корзина</h1>  {store\_arr}  <h2>сумма: {sum}</h2>  <button onClick={clearPrinter}>очистить</button>  <button disabled={localStorage.length==0}><a href="/form" class={`${localStorage.length==0}`}>оформить</a></button>    </div> |

Листинг 3.6 – Структура компоненты Profile

Таким образом, в этом разделе были рассмотрены основные компоненты приложения, которые будут использоваться в дальнейшем. Хотелось бы отметить, что JSX-формат очень удобный для написания кода, так как он позволяет писать код в HTML-формате, что упрощает его чтение и понимание.

## 3.2 Структура react-redux

Для того, чтобы удобно работать с данными, был использован react-redux. В нем будет хранится информация о заполнении корзины. Были созданы редьюсер и состояние по умолчанию. Редьюсер и состояние показаны в листинге 3.7.

|  |
| --- |
| const defaultState = {  storeAmount: localStorage.length,  }  const reducer = (state = defaultState, action) => {  switch(action.type){  case "ADD":  return {...state, storeAmount: state.storeAmount+1,}  case "DEL":  return {...state, storeAmount: state.storeAmount-1,}  case "CLR":  return {...state, storeAmount: 0,}  default:  return state  }  }  const store = createStore(reducer) |

Листинг 3.7 – Код редьюсера и состояния

Благодаря useSelector получаем колличество элементов в корзине и выводим цифрой возле иконки корзины. Код useSelector будет показан в листинге 3.8.

|  |
| --- |
| <h2 class={useSelector(state => state.storeAmount)}>{useSelector(state => state.storeAmount)}</h2> |

Листинг 3.8 – Структура кода useSelector

## 3.3 Добавление таблиц стилей Sass

Для стилизации был использован SASS, который позволяет использовать переменные, вложенность, миксины, наследование, импорт и экспорт. Код использования SASS показан в листинге 3.9.

|  |
| --- |
| header {  position:fixed;  width: 100%;  background-color: $lightGray;  font-family: "Lato Regular";  font-weight: bold;  box-shadow: 0px 3px 4px 3px $black ;  &[class="true"]{  box-shadow: none;  div{  @include do\_flex(90%);  justify-content: center;  align-items: center;  margin: auto;  h2{  display: none;  }  img {  position: absolute;  display: none;  width: 3.5vw;  }  nav {  @include do\_flex(75%);  align-self: center;  font-size: 140%;  letter-spacing: 2px;  font-weight: bold;  font-style: normal;  }  }  }  div{  @include do\_flex(90%);  align-items: center;  margin: auto;  &>a{  text-decoration: none;  display: flex;  h2{  color: $retro\_blue;  margin: 0px;  margin-left: 5px;  text-decoration: none;  &[class="0"]{  display: none;  }  &:visited{  text-decoration: none;  }  &:link{  text-decoration: none;  }  &:checked{  text-decoration: none;  }    }  &:hover{  filter: hue-rotate(90deg) brightness(90%) contrast(300%);  }  }  img {  width: 3.3vw;  transition: 0.1s linear;  }  nav {  @include do\_flex(60%);  align-self: center;  font-size: 120%;  letter-spacing: 2px;  font-weight: bold;  font-style: normal;  ul{  padding: 0px;  @include do\_flex(100%);  margin: 0px;  li{  display: block;  list-style: none;  width: 30%;  text-align: center;  transition: 0.1s linear;  background-color: $lightGray;  color: $retro\_blue;  a{    text-decoration: none;  display: inline-block;  color: $retro\_blue;  padding: 0.8vw;  font-weight: bold;  font-style: normal;  }  &:hover{  background-color: $retro\_blue;  color: $lightGray;  &>a{  color: $lightGray;  }  }  &:nth-child(1){  &:hover{  background-color: black;  color: #FFF500;  &>a{  color: #FFF500;  }  }    }  &:nth-child(2){  &:hover{  background-color: #0093DD;  color: $lightGray;  &>a{  color: black;  }  }  }  &:nth-child(3){  &:hover{  background-color: #DD127B;  color: $lightGray;  &>a{  color: black;  }  }  }  &:nth-child(4){  &:hover{  background-color: #FFF500;  color: black;  &>a{  color: black;  }  }  }  }  .menu-item{  position: relative;  transition: 0.1s linear;  cursor: pointer;  .submenu{  display: block;  width: 100%;  visibility:hidden;  text-align: center;  background-color: #DD127B;  position: absolute;  z-index: 3;  cursor: pointer;  li{  background-color: #DD127B;  color: black;  list-style: none;  width: 100%;  box-sizing: border-box;  text-align: center;  a{  color: $black;  }  &:hover{  background-color: $lightGray;  color: $retro\_blue;  &>a{  color: $retro\_blue;  }  }  }  }  &:hover > .submenu{  visibility:visible;  }  }  }  }  }  } |

Листинг 3.9 – Структура кода SASS

В этом разделе была рассмотрена структура SASS, что такое SASS и как он используется в проекте.

## 3.4 Выводы по разделу

Таким образом, были рассмотрены основные моменты, которые будут реализованы в проекте. В проекте будет реализована, добавление товаров в корзину, оформление заказа, получение информации о товарах, о корзине.

# 4 Тестирование веб-сайта

## 4.1 Адаптивный дизайн веб-сайта

Сегодня количество пользователей, которые заходят в Интернет с мобильных устройств очень велико, и с каждым днем их становится все больше. А потому адаптация веб-сайта под маленькие экраны – актуальная задача и для курсового проекта.

Передо мной стояла задача: создать сайт, который будет сам подстраиваться под разные типы экранов. Главными условиями адаптивности является не фиксированный размер элементов страницы.

Для создания адаптивности веб-сайта были использованы следующие инструменты: медиа-запросы (@media), единицы измерения, которые связаны непосредственно с параметрами области просмотра (представляют собой процентные величины от текущей области просмотра браузера).

После написания основной структуры страницы на JSX и внешнего стилевого оформления появился вопрос о тестировании. Веб-сайт был открыт при помощи различных браузеров. После этого было выявлено, что во всех браузерах сайт выглядит одинаково.

При отображении сайта на мобильных устройствах навигационное меню в верхнем колонтитуле меняется на выпадающее меню, открывающееся при нажатии на кнопку меню. На рисунке 4.1 можно увидеть пример адаптивного дизайна для мобильных устройств.

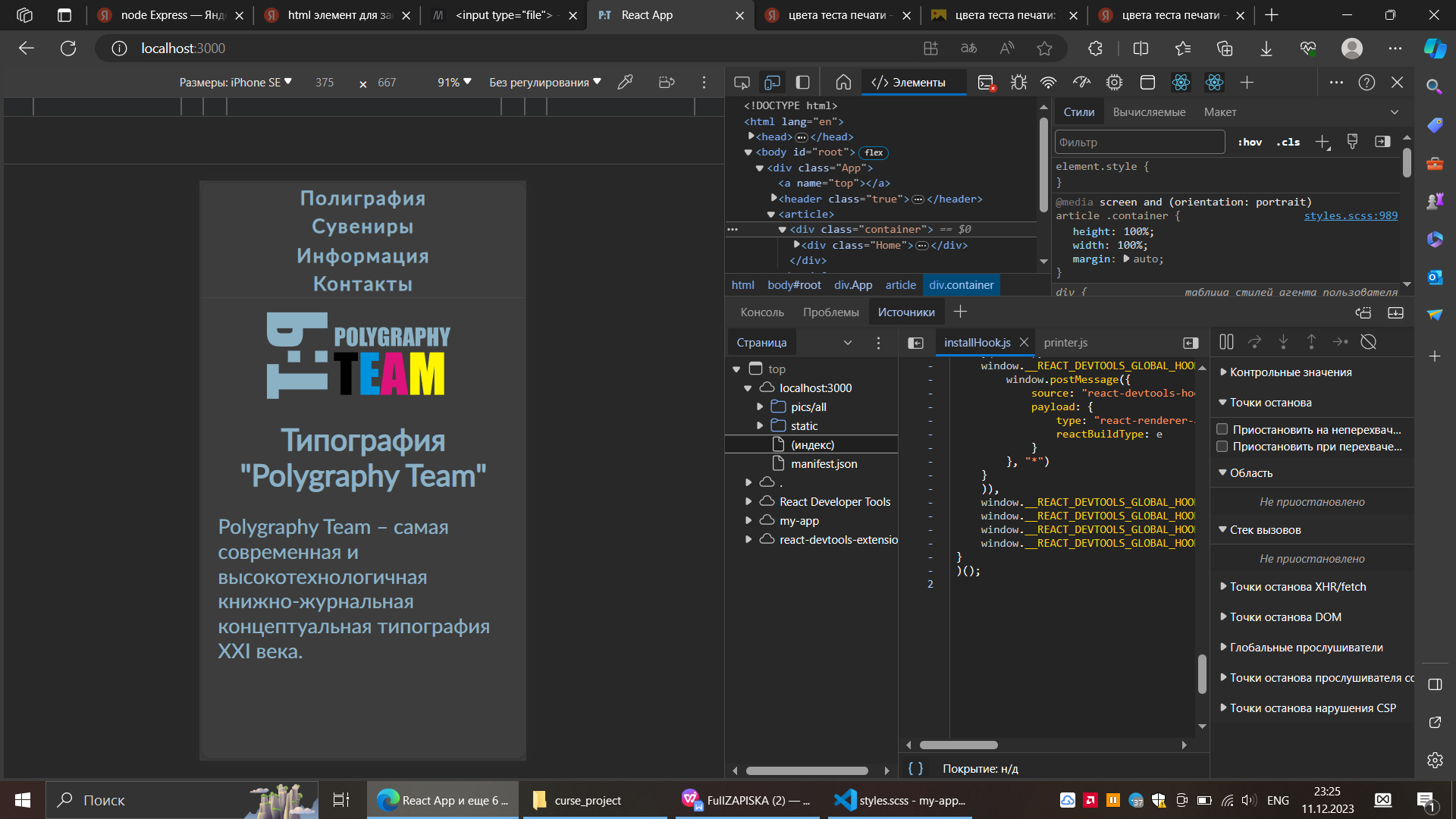


Рисунок 4.1 – Адаптивный главной страницы для телефонов

Также при использовании мобильного устройства блоки с продуктами на странице для отображения располагаются друг под другом. На рисунке 4.2 можно увидеть пример.

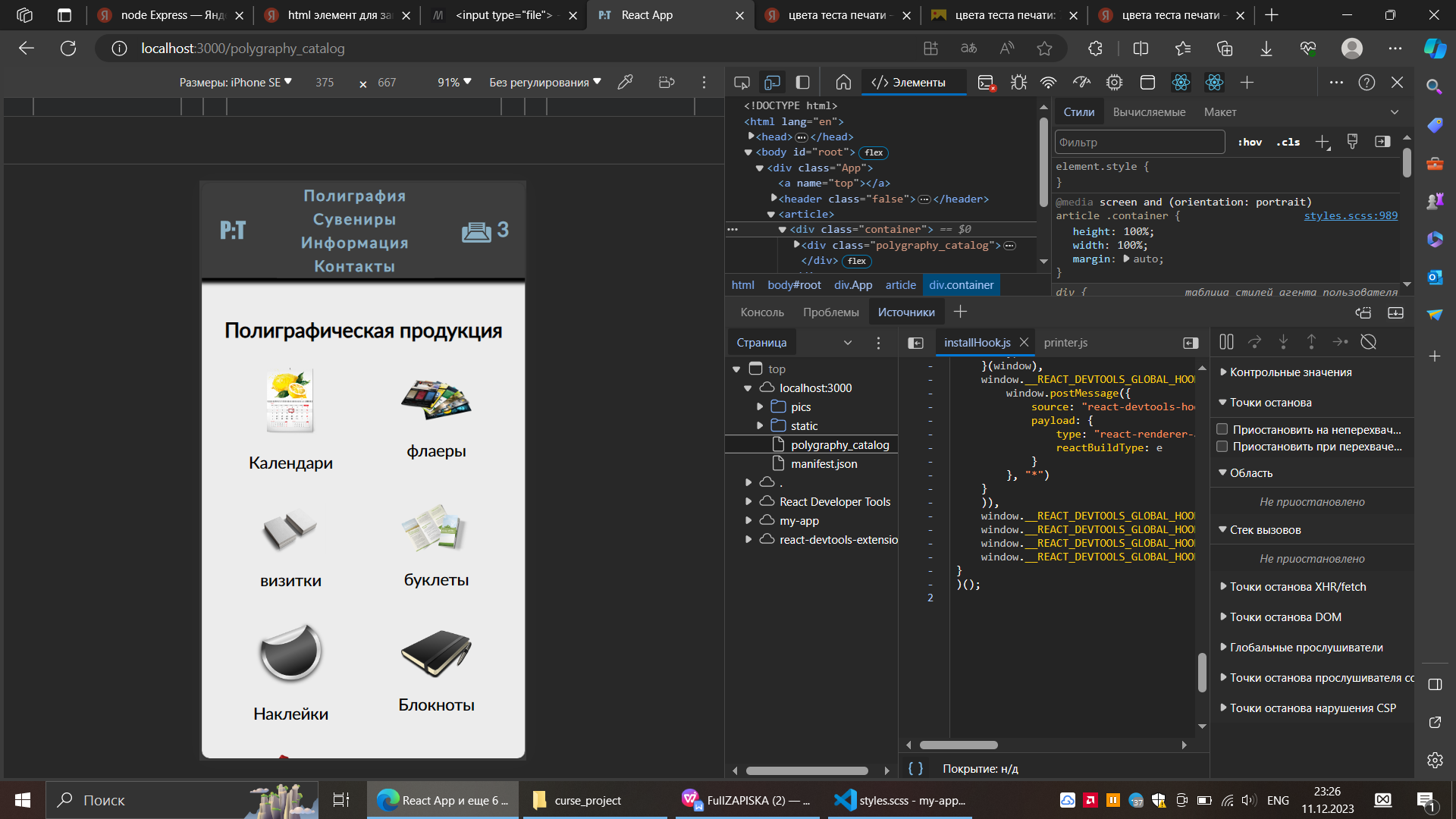


Рисунок 4.2 – Блоки с продуктами на мобильных устройствах

И таким образом можно сказать, что все блоки располагаются друг за друг, потому что пользователю было все хорошо видно на его мобильном устройстве.

## 4.2 Кроссбраузерность веб-сайта

Очень важно, чтобы наш сайт был кроссбраузерным. Кроссбраузерность – это способность веб-ресурса отображаться одинаково и работать во всех популярных браузерах, без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента.

У каждого браузера есть свои встроенные, экспериментальные или нестандартные свойства и для того, чтобы они корректно работали было принято решения о внедрении вендорных префиксов. Они были автоматически прописаны после компиляции файла с расширением sass.

При проверке веб-сайта в Microsoft Edge никаких искажений текста, анимации, переходов не было выявлено. На рисунке 4.3 можно увидеть пример веб-страницы в Microsoft Edge.

|  |  |
| --- | --- |
|  |  |
| а) | б) |

Рисунок 4.3 – Вид веб-страницы в Microsoft Edge а) мобильная версия; б) компьютерная версия

В Firefox сайт показал схожий результат: никаких отклонений как для мобильной версии сайта, так и для компьютерной не было выявлено, текст не съехал, не произошло его наложения на картинки. На рисунке 4.4 можно увидеть пример веб-страницы в Firefox.

|  |  |
| --- | --- |
|  |  |
| а) | б) |

Рис 4.4 – Вид веб-страницы в Firefox а) мобильная версия; б) компьютерная версия

Также для тестирования сайта было использовано Linter. Linter - это инструмент, который проверяет код на наличие ошибок. В нашем случае он проверяет код на наличие ошибок в HTML и CSS.

При просмотре веб-сайта с помощью таких браузеров как Microsoft Edge и Firefox общий вид веб-страницы не изменился.

Главной целью такой тщательной проверки сайта является грамотная настройка всех показателей, одной или всех страниц сайта, оперативное обнаружение и устранение всех дефектных мест на сайте, а также его адаптация под различные устройства.

Динамика сайта на разных устройствах – наиболее важная часть в написании сайта, поэтому, на данном этапе было проанализировано поведение веб-сайта при использовании с разных устройств, а также веб-ресурс был протестирован в большом количестве браузеров. По итогу выполнения данного этапа для веб-сайта был разработан адаптивный дизайн и кроссбраузерность. Работа, проделанная с сайтом, позволяет пользователям ориентироваться по сайту. Лишних элементов нет, что упрощает пользование сайтом. При пользовании сайтом при помощи мобильного устройства некоторые элементы меняют свое положение, для более удобного пользования. В ходе тестирования были проверены все ссылки (включая модальные окна), проверены формы регистрации пользователя. Была выполнена проверка контента. Также в ходе тестирования была проверена совместимость с браузерами. (Microsoft Edge, Firefox).

## 4.3 Тестирование компонентов

Для тестирования компонента Polygraphy\_catalog была использована библиотека автоматического тестирования Jest, данная библиотека позволяет осуществить тестирование js кода по определенному сценарию, ниже на листинге 4.5 показан код сценария тестирования компонента Polygraphy\_catalog. В данном сценарии проверяется корректность рендера .

import Polygraphy\_catalog from "../src/Polygraphy\_catalog";

test.todo("say Hello")

describe("ComponentToTest", () => {

    test("renders component correctly", () => {

      render(<Polygraphy\_catalog />);

    });

    test("handles click event correctly", () => {

        render(<Polygraphy\_catalog />);  });

})

Листинг 4.5 – Сценарий тестирования компонента Polygraphy\_catalog

## 4.4 Выводы по разделу

В этом разделе было проведено тестирование функционала сайта. Были проверены все функции сайта, которые должны работать корректно, а также были проверены все элементы, которые должны быть доступны для пользователя. Не было обнаружено никаких ошибок, которые могли бы повлиять на работу сайта.

# 5 Руководство пользователю

Данный веб-сайт предназначен для оформления заказов на печать. Для того, чтобы совершить заказ, необходимо зайти в нужный раздел, перейти на страницу товара, указать тираж и нужные параметры после чего добавить в корзину. Когда все необходимые товары добавлены в корзины, нужно перейти внутрь корзины, проверить данные и перейти к оформлению. Для оформления необходимо обязательно ввести имя, фамилию, адрес электронной почты и прекрепить файл с макетами, так же по желанию может быть указан номер телефона.

На странице Home пользователь может узнать краткую информацию.

На странице Polygraphy\_catalog, Gift\_catalog пользователь может просматривать товары, которые находятся в наличии.

На странице ItemCard пользователь может просматривать полностью описание товара, а также добавить товар в корзину.

На странице Printer пользователь может просматривать товары, которые он добавил в корзину, а также удалять их из корзины.

На страницах Requirements, About\_formats, Emploers\_catalog пользователь может узнать всё необходимую информацию для работы и оформления заказа

# Заключение

Целью на курсовое проектирование было создание сайта печатного центра, где пользователи смогут оформлять заказы на печать разной продукции.

В ходе курсовой работы были выполнены следующие задачи:

* Проанализировать и описать основные принципы работы React и Redux.
* Разработать приложение, используя React и Redux.
* Провести тестирование приложения.
* Провести анализ и оценку качества приложения.

А также были выполнены задачи по созданию интернет-магазина:

* Разработать возможность добавления и удаления товаров из корзины.
* Разработать структуру веб-сайта.
* Разработать возможность заказа товара.

Подводя итоги курсовой работы, можно сделать вывод, что в ходе работы были выполнены все поставленные задачи. Были изучены основные принципы работы React и Redux, что позволило разработать приложение и закрепить полученные знания.

# Список использованных источников

1. Карандаш [Электронный ресурс]. – Режим доступа: [https://www.karandash.by](https://www.karandash.by/?ysclid=lq023rzjvf145577627) – Дата обращения 29.11.2023.
2. Печатник [Электронный ресурс]. – Режим доступа: https://pechatnik.by – Дата обращения 29.11.2022.
3. Санпринт [Электронный ресурс]. – Режим доступа: <https://sunprint.by> – Дата обращения 29.11.2022.
4. Документация по React JS [Электронный ресурс]. - Режим доступа: https://reactjs.org/. – Дата обращения 02.12.2022.
5. Документация по Redux/Redux-tolkit [Электронный ресурс]. - Режим доступа: https://redux-toolkit.js.org/ . – Дата обращения 02.12.2022.
6. Документация по Sass/Scss [Электронный ресурс]. - Режим доступа : <https://sass-scss.ru/guide/> – Дата обращения 02.12.2022.

# Приложение A Листинг JSX-документа

|  |
| --- |
| <div className="App">  <a name="top"></a>    <header class={hasHello}>  <div>  <a href="/"><img class="logo" src="./pics/all/logo\_mini.svg" alt="img"/></a>  <nav>  <ul>  <li class="menu-item"><a href="/polygraphy\_catalog">Полиграфия</a></li>  <li class="menu-item"><a href="/gift\_catalog">Сувениры</a></li>  <li class="menu-item"><a >Информация</a>  <ul class="submenu">  <li><a href="/requirements">Требования к макетам</a></li>  <li><a href="/about\_formats">Форматы в полиграфии</a></li>  </ul>  </li>  <li class="menu-item"><a href="/contacts">Контакты</a></li>  </ul>  </nav>  <a href="/printer" class={"printIcon"}><img class="logo" src="./pics/all/printer.svg" alt="img"/><h2 class={useSelector(state => state.storeAmount)}>{useSelector(state => state.storeAmount)}</h2></a>  </div>  </header>  <article>  <div class="container">  <Routes>  <Route path="/" element={<Home/>} exact />  <Route path="/polygraphy\_catalog" element={<Polygraphy\_catalog/>} exact/>  <Route path="/polygraphy\_catalog/:id" element={<Item\_card/>} exact/>  <Route path="/gift\_catalog" element={<Gift\_catalog/>} exact />  <Route path="/gift\_catalog/:id" element={<Gift\_card/>} exact />  <Route path="/printer" element={<Printer/>} exact />  <Route path="/requirements" element={<Requirements/>} exact />  <Route path="/about\_formats" element={<About\_formats/>} exact />  <Route path="/contacts" element={<Emploers\_catalog/>} exact />  <Route path="/form" element={<Order\_Form/>} exact />  </Routes>  </div>  </article>  <footer>  <h3>Ivan Drebezov | FIT | ISIT - 2 | 2023</h3>  </footer>  </div> |

Листинг 1 - структура JSX—документа

# Приложение Б Листинг (SASS)

|  |
| --- |
| @import "../../../sass/value.scss";  @import "../../../css/fonts.css";  @mixin do\_flex($width){  width: $width;  display: flex;  justify-content: space-around;  }  @media screen and (orientation: landscape){    html{  scroll-behavior: smooth;  min-height: 100vh;  min-width: 100vw;  overflow-x: hidden;  }  body {  background-color: $white;  width: 100vw;  height: 100vh;  margin: auto;  font-family: "Lato Regular";  display: flex;  flex-direction: row;  align-items: stretch;  }  header {  position:fixed;  width: 100%;  background-color: $lightGray;  font-family: "Lato Regular";  font-weight: bold;  box-shadow: 0px 3px 4px 3px $black ;  &[class="true"]{  box-shadow: none;  div{  @include do\_flex(90%);  justify-content: center;  align-items: center;  margin: auto;  h2{  display: none;  }  img {  position: absolute;  display: none;  width: 3.5vw;  }  nav {  @include do\_flex(75%);  align-self: center;  font-size: 140%;  letter-spacing: 2px;  font-weight: bold;  font-style: normal;  }  }  }  div{  @include do\_flex(90%);  align-items: center;  margin: auto;  &>a{  text-decoration: none;  display: flex;  h2{  color: $retro\_blue;  margin: 0px;  margin-left: 5px;  text-decoration: none;  &[class="0"]{  display: none;  }  &:visited{  text-decoration: none;  }  &:link{  text-decoration: none;  }  &:checked{  text-decoration: none;  }    }  &:hover{  filter: hue-rotate(90deg) brightness(90%) contrast(300%);  }  }  img {  width: 3.3vw;  transition: 0.1s linear;  }  nav {  @include do\_flex(60%);  align-self: center;  font-size: 120%;  letter-spacing: 2px;  font-weight: bold;  font-style: normal;  ul{  padding: 0px;  @include do\_flex(100%);  margin: 0px;  li{  display: block;  list-style: none;  width: 30%;  text-align: center;  transition: 0.1s linear;  background-color: $lightGray;  color: $retro\_blue;  a{    text-decoration: none;  display: inline-block;  color: $retro\_blue;  padding: 0.8vw;  font-weight: bold;  font-style: normal;  }  &:hover{  background-color: $retro\_blue;  color: $lightGray;  &>a{  color: $lightGray;  }  }  &:nth-child(1){  &:hover{  background-color: black;  color: #FFF500;  &>a{  color: #FFF500;  }  }    }  &:nth-child(2){  &:hover{  background-color: #0093DD;  color: $lightGray;  &>a{  color: black;  }  }  }  &:nth-child(3){  &:hover{  background-color: #DD127B;  color: $lightGray;  &>a{  color: black;  }  }  }  &:nth-child(4){  &:hover{  background-color: #FFF500;  color: black;  &>a{  color: black;  }  }  }  }  .menu-item{  position: relative;  transition: 0.1s linear;  cursor: pointer;  .submenu{  display: block;  width: 100%;  visibility:hidden;  text-align: center;  background-color: #DD127B;  position: absolute;  z-index: 3;  cursor: pointer;  li{  background-color: #DD127B;  color: black;  list-style: none;  width: 100%;  box-sizing: border-box;  text-align: center;  a{  color: $black;  }  &:hover{  background-color: $lightGray;  color: $retro\_blue;  &>a{  color: $retro\_blue;  }  }  }  }  &:hover > .submenu{  visibility:visible;  }  }  }  }  }  }  .root, .App{  min-height: 100%;  min-width: 100%;  }  article {  min-height: 100vh;  min-width: 100vw;  section{  width: 75%;  margin: auto;  padding: 5%;  h1{  font-size: 120%;    }  h2{  font-size: 24px;  }  p{  font-size: 18px;  }  }  .container{  height: 100%;  width: 100%;  //box-shadow: 0px 0px 15px 3px rgb(20, 05, 05, 0.5);  margin: auto;  .polygraphy\_catalog{  padding-top: 7%;  padding-bottom: 15%;  width: 100%;  height: 100%;  display: flex;  flex-wrap: wrap;  justify-content: space-around;  h1{  font-size: 200%;  margin: 5%;  }  .market{  display: flex;  flex-wrap: wrap;  justify-content: space-around;  width: 90%;  .poly\_item\_card{  margin: 2.5%;  width: 20%;  a{  color:black;  display: block;  width: 100%;  height: 100%;  text-decoration: none;  display: flex;  flex-direction: row;  align-items: stretch;  div{  width: 100%;  height: 100%;  display: flex;  flex-direction: column;  align-content: stretch;  justify-content: center;  img{  transition: 0.15s linear;  margin-left: auto;  margin-right: auto;  width: 60%;  }  p{  transition: 0.15s linear;  font-size: 120%;  text-align: center;  }  }  &:hover{  div{  width: 100%;  height: 100%;  display: flex;  flex-direction: column;  align-content: stretch;  justify-content: center;  img{  margin-left: auto;  margin-right: auto;  width: 70%;  }  p{  text-align: center;  font-size: 140%;  }  }  }  }  }  }  }  .Gift\_catalog{  padding-top: 5%;    width: 100%;  height: 100%;  display: flex;  flex-wrap: wrap;  justify-content: center;  margin: auto;  h1{  font-size: 200%;  margin: 5%;  width: 100%;  }  .marketContr{  display: flex;  justify-content: space-around;  width: 100%;  .ctrl{  word-wrap: break-word;  font-weight: bold;  width: 20%;  background-color: rgb(200, 200, 200);  border-right: 2px solid $lightGray;  .controls{  margin-left: auto;  margin-right: 0;  min-height: 50vh;  max-height: 100vh;  font-size: 20px;  position: sticky;  top: 100px;  padding: 10px 10px 20px 20px;  &>p{  text-align: center;  }  p{  font-size: 25px;  margin: 25px 0px 10px 0px;  }  button{  cursor: pointer;  width: 75%;  display: block;  margin: auto;  background-color: #0093DD;  color: black;  font-family: "Lato Regular";  font-size: 22px;  border: none;  padding: 5px;  box-shadow: 1px 1px 1px 1px black;  transition: 0.1s linear;  &:hover{  box-shadow: none;  font-size: 24px;  font-weight: bold;  background-color: rgb(200, 200, 200);  color: #0093DD;  }  &:active{  font-size: 20px;  }  }  &>div{  padding: 20px 10px 20px 50px;  }  input[type="checkbox"]{  cursor: pointer;  display: none;  &:checked + label{  color: #0093DD;  user-select: none;  font-weight: bold;  &::before{  margin-right: 5px;  content:"\25A0";  font-size: 30px;  }  }  &:not(:checked) + label{  color: $lightGray;  user-select: none;  font-weight: normal;  &::before{  margin-right: 15px;  content:"\25A1";  font-size: 30px;  }  }  }  label{  cursor: pointer;  }  }  }  .market{  display: flex;  flex-wrap: wrap;  justify-content: space-around;  width: 80%;  .poly\_item\_card{  margin: 1%;  width: 30%;  a{  color:black;  display: block;  width: 100%;  height: 100%;  text-decoration: none;  display: flex;  flex-direction: row;  align-items: stretch;  div{  width: 100%;  height: 100%;  display: flex;  background-color: none;  flex-direction: column;  align-content: stretch;  justify-content: center;  transition: 0.1s linear;  img{  transition: 0.15s linear;  margin-left: auto;  margin-right: auto;  width: 70%;  }  p{  transition: 0.15s linear;  font-size: 120%;  text-align: center;  }  }  &:hover{  div{  width: 100%;  height: 100%;  display: flex;  flex-direction: column;  align-content: stretch;  justify-content: center;  img{    margin-left: auto;  margin-right: auto;  width: 60%;  }  p{  text-align: center;  font-size: 110%;  }  }  }  }  }  }  }  }  h1{  text-align: center;  font-size: 3vw;  margin: 0px;  }  table{  margin: auto;  width: 90%;  border-spacing: 0px;  font-size: 18px;  text-align: center;  td{  padding: 3px;  }    }  .emploer\_catalog{  width: 80%;  margin: auto;  padding-top: 80px;    .emploers\_table{  width: 100%;  display: flex;  flex-wrap: wrap;  justify-content: center;  .emploer\_card{  border: 2px solid black;  margin: 2%;  width: 25%;  box-shadow: 0px 2px 4px 1px black;  div{  width: 90%;  padding: 2%;  margin: auto;  img{  width: 100%;  }  }  }  }  }  .Home{  background-color: $lightGray;  article {  width: 100%;  height: 100%;  animation: loop 4s;  z-index: 2;  @include do\_flex(95%);  .container{  height: 80vh;  @include do\_flex(90%);  align-items: center;  margin: auto;  .logo {  @include do\_flex(50%);  text-align: center;  img{  align-self: center;  width: 60%;  }  }  .about{  width: 35%;  color: $retro\_blue;  h1{  font-size: 220%;  }  p{  font-size: 150%;  }  }  }  }  }  .itemPage{  padding-top: 120px;  width: 100%;  &>p{  font-size: 35px;  text-align: center;  }  .actSection{  font-size: 20px;  margin: auto;  display:flex;  align-items: center;  justify-content: center;  width: 70%;  .image{  width: 40%;  img{  width: 80%;  }  }  .action{  & > \*{  display: flex;  width: 100%;  justify-content: space-between;  align-items: center;  }  input, select, option{  font-size: 20px;  }  .draw{  display: block;  .buttons{  input[type="range"]{  appearance: auto;  cursor: pointer;  width: 100%;  transition: 0.1s linear;  accent-color: black;  &:active{  cursor: grabbing;  }  }  }  input[type="button"]{  width: 20%;  }  input[type="number"]{  width: 55%;  }  }  select{  width: 90%;  height: 40%;  }  input[type="button"]{  margin: auto;  width: 50%;  font-size: 20px;  justify-content: center;  background-color: #0093DD;  color: black;  font-weight: bold;  transition: 0.1s linear;  padding: 5px;  &:enabled:hover{  background-color: $lightGray;  color: #0093DD;  }  &:disabled{  background-color: $white;  color: $lightGray;  }  }  input[type="number"]{  padding: 5px;  }  }  }  .infoSection{  width: 80%;  margin: auto;  word-wrap: break-word;  margin-top: 40px;  margin-bottom: 40px;  h1{  font-size: 35px;  }  p{  font-size: 20px;  text-indent: 30px;  }  }  }  .form{  width: 40%;  margin: auto;  padding: 15vh;  input, select, option, button{  font-size: 20px;  }  input[type="submit"]{  margin: auto;  width: 50%;  font-size: 20px;  justify-content: center;  background-color: #0093DD;  color: black;  font-weight: bold;  transition: 0.1s linear;  padding: 5px;  &:enabled:hover{  background-color: $lightGray;  color: #0093DD;  }  &:disabled{  background-color: $white;  color: $lightGray;  }  }  }  .printerPage{  padding-top: 15vh;  padding-bottom: 10vh;  width: 50%;  margin: auto;    .printerItem{  margin: auto;  padding: 20px;  margin: 20px;  width: 90%;  display: flex;  justify-content:space-between;  align-items: center;  background-color: rgb(200, 200, 200);  .printerItemInfo{  width: 50%;  }  .printerItemPic{  width: 30%;  img{  width: 70%;  }    }  button{  transition: 0.1s linear;  width: 7%;  height: 50%;  font-size: 30px;  background:#0093DD;  aspect-ratio: 1/1;  img{  width: 80%;  }  &:hover{  box-shadow: none;  font-size: 24px;  font-weight: bold;  background-color: rgb(200, 200, 200);  color: #0093DD;  }  &:active{  font-size: 20px;  }    }  }  &>h2{  text-align: center;  }  &>button{  cursor: pointer;  width: 75%;  display: block;  margin: auto;  background-color: #0093DD;  color: black;  font-family: "Lato Regular";  font-size: 22px;  border: none;  padding: 5px;  box-shadow: 1px 1px 1px 1px black;  transition: 0.1s linear;  &:hover{  box-shadow: none;  font-size: 24px;  font-weight: bold;  background-color: rgb(200, 200, 200);  color: #0093DD;  }  &:active{  font-size: 20px;  }  &:disabled{  background-color: $white;  color: rgb(180, 180, 180);  }  a{  display: block;  width: 100%;  text-decoration: none;  color: black;  &:hover{  color: #0093DD;  }  }  .true{    font-size: 22px;  display: block;  width: 100%;  text-decoration: none;  background-color: $white;  color: rgb(180, 180, 180);  pointer-events: none;  cursor: default;    }  }  }  }  }  footer {  background-color: $lightGray;  color: $retro\_blue;  text-align: center;  padding: 5px;  h3{  margin: 0px;  }  }  } |

Листинг 2 - структура SASS