

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ РАБОТКИ ИНТЕРНЕТ-ВИТРИН. 6	
1.1. Анализ существующих интернет-магазинов.....	6
1.2. Сравнительная характеристика и выбор среды разработки.	9
1.3. Анализ подхода к созданию интернет-витрины для SPA.....	10
1.3.1. Выбор языка программирования для написания логики интернет-витрины.	10
1.3.2. Выбор формального языка стилей веб-страниц	12
1.3.3. Выбор библиотеки для создания SPA.....	13
1.3.4. Выбор библиотеки для написания бизнес-логики веб-приложения....	18
1.3.5. Обоснование необходимости разработки интернет-витрины для SPA	20
2. РАЗРАБОТКА ШАБЛОНА WEB-РЕСУРСА.....	24
2.1. Этапы разработки дизайна интернет-витрины на сайте Figma.....	24
2.2. Написание компонентов, стилей и логики фильтрации товаров в среде разработки WebStorm.....	25
2.3. Инструкция использования шаблона в проекте.....	31
2.4. Размещение кода интернет-витрины для SPA на площадку GitHub.	32
ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	37
ПРИЛОЖЕНИЯ.....	39

ВВЕДЕНИЕ

Последнее время набирает популярность создание веб-приложений. Технология создания веб-приложений стала активно использоваться в конце 90-х - начале 2000-х годов. Появление в 2009 году Node JS позволило разработчикам не переключаться между языками программирования и создавать веб-приложения в разы быстрее. Также серверам не приходилось затрачивать большое количество вычислительной мощности на обработку большого количества запросов пользователей. JavaScript, использующийся в создании веб-ресурсов – однопоточный и асинхронный, благодаря этому все операции не блокируют остальные. Такая высокая скорость позволяет предприятиям значительно экономить свои финансы для разработки своего продукта.

Тематика данной работы актуальна в наше время, так как веб-страницы, которые мы видим почти каждый день, занимают большую часть нашего свободного времени. Одностраничные веб-приложения (англ. Single Page Application) – это веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JS. Разработанный шаблон поможет современным малым предприятиям экономить часть денег на разработку и продвижение.

Целью данной выпускной квалификационной работы является разработка шаблона для реализации интернет-витрин на основе веб-приложения.

Для разработки данного сайта, необходимо решить следующие задачи:

- проанализировать интернет-витрины популярных интернет-магазинов;
- исследовать и сравнить существующее программное обеспечение сред разработки кода;
- провести анализ предметной области существующего программного обеспечения в среде одностраничных веб-приложений;
- разработать макет интерфейса интернет-витрины;
- написание дизайна, компонентов и логики фильтрации товаров;
- разработка шаблона интернет-витрины для одностраничных веб-приложений;

– разместить код шаблона на общедоступную площадку GitHub.

Объектом исследования является шаблон, представляющий собой интернет-витрину для одностраничных веб-приложений.

Предметом исследования является процесс разработки шаблона интернет-витрины для SPA с помощью библиотек JavaScript, выбор языка программирования и изучение выбранной библиотеки.

Все технологии, которые будут применяться для создания шаблона интернет-витрины, соответствуют современным стандартам разработки веб-приложений. Внешний вид и удобство подобных веб-приложений также важны для создания приятного впечатления. Пользователи не любят долго ждать обработки их заявок на покупку услуг или товаров, но, не взирая на данную проблему, множество компаний не задумываются о том, что такое решение приведет к приросту продаж по причине скорости отклика таких приложений.

1. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ РАБОТКИ ИНТЕРНЕТ-ВИТРИН

1.1. Анализ существующих интернет-магазинов

В наши дни интернет-магазины благополучно влияют на прибыль компаний. Интернет-магазин представляет собой одну из разновидностей электронной коммерции. Производители размещают информацию о своих товарах и услугах на веб-сайтах. Для простых людей покупка товара или услуги через Интернет очень удобна. Не нужно никуда идти и часами ходить по магазину в поисках нужного товара. Теперь достаточно просто по критериям отсортировать товары и выбрать нужный. Для лучшего анализа следует обратить внимание на самые распространенные среди масс интернет-магазины. В качестве примера будут рассмотрены такие интернет-магазины как: WildBerries, Ozon и Яндекс.Маркет.

1) WildBerries

WildBerries – первый в списке крупнейших коммерческих игроков. Около 400 тысяч продаж за сутки в конце 2018 года. Учитывая такой объем продаж, анализ данного сайта необходим.

Характеристика:

- Регистрации аккаунта. Нужен только номер телефона и карты;
- Удобство ЛК. Местами функционал показывает себя с плохой стороны, так как не обращали внимание на конкурентов. По итогу – ЛК весьма функциональный, при этом довольно неудобный;
- Продажи. Хороший интернет-магазин, так как продажи в действительности в полтора раза больше, чем у Ozon. На некоторые категории, которые распространены среди женщин, примерно в три раза больше.

2) Ozon

В сфере онлайн продаж является самым старым игроком рынка, так как был основан еще в 1998 году. Последние годы продажи Ozon растут до 70, а иногда и до 100 процентов. В последнем месяце 2018 года, во время предпраздничных распродаж, у Ozon было в районе 120 тысяч заказов ежедневно.

Характеристика:

- Регистрации аккаунта. У Ozon простой, понятный и приятный процесс регистрации;
- Удобство ЛК. Интерфейс хорошо реализован, благодаря этому ЛК понятен и удобен в использовании;
- Продажи. У Ozon ситуация чуть хуже, чем у WildBerries, но объем все равно выше, если брать иные офлайн и онлайн магазины.

3) Яндекс.Маркет

Анализ данного интернет-магазина будет немного отличаться от его конкурентов, так как Яндекс – крупная компания, имеющая большое количество веб-сервисов. Данный сервис, включающий в себя три сервиса «Гуру», «Товары» и «Подбери.ру», появился в 2002 году.

Характеристика:

- Регистрация аккаунта. При регистрации аккаунта у пользователя открывается доступ ко всем веб-сервисам Яндекса. Зарегистрировать почтовый ящик Яндекс не занимает много времени;
- Удобство ЛК. ЛК у Яндекс привязан ко всем его веб-сервисам, что является большим плюсом;
- Продажи. На октябрь 2020 года на Яндекс.Маркет по рекламной модели представлено более 230 миллионов предложений от более чем 25 тысяч магазинов, а также более 1,3 миллионов товаров от 7 тысяч партнеров доступных для покупки.

Перед тем, как начать разработку интернет-витрины, следует определить, что интернет-магазин – сложный проект. Во-первых, это программный продукт, который нужно разработать. Во-вторых, это ИС, которая должна решать поставленные задачи компании, необходимая в разработке. Зачастую, следует определить структуру данных, их взаимосвязь и методы обработки, спроектировать интерфейсы.

Для того, чтобы проще понять данную онлайн отрасль, следует провести анализ дизайна и функциональности похожих по сфере продаж ресурсов в Интернете. В связи с этими особенностями появится возможность понять, что работает и привлекает, а что не работает и отталкивает пользователей.

После определения критериев анализа была составлена таблица информационного наполнения интернет-витрины. Эта таблица показывает, какие фильтры поиска товаров используют различные ресурсы. Соответственно, если данный раздел присутствует, то в ячейке напротив фильтра, находящейся под строкой наименований интернет-магазинов, ставится «+», если же отсутствует, то «-». После данного анализа можно будет понять с каким сайтом работать комфортнее.

Общее количество критериев было достаточно велико, причем некоторые варианты встречались большое количество раз.

Окончательный результат часто встречаемых фильтров поиска нужного товара представлен в таблице 1.

За основу было взято три самых популярных интернет-магазина, а именно Яндекс.Маркет, WildBerries и Ozon. Была взята самая популярная категория – гардероб.

Таблица 1. Поиск различий между фильтрами товаров

Название интернет-магазина / Наименование фильтра	Яндекс.Маркет	WildBerries	Ozon
Пол	+	+	+
Цвет	+	+	+
Размер	+	+	+
Производителю	+	+	+
Материалы	+	+	+
Цена	+	+	+
Типа размера товара	+	-	-
Наличие скидки	+	+	+

Просмотрев таблицу 1, видно, что фильтры на представленных веб-сервисах имеют мало различий. Как было упомянуто ранее, Яндекс – большая компания, имеющая многочисленные веб-сервисы. Эта особенность помогает Яндекс.Маркет хоть и на немного, но отрываться от конкурентов.

Расположение блока с фильтром товаров различается, но не сильно. Бывают случаи, когда интернет витрина занимает всю площадь экрана пользователя, что заставляет пользователя теряться и больше «бегать» глазами по монитору в поиске нужного фильтра. Примером такого интерфейса является популярный сайт WildBerries. Данный интерфейс является очень громоздким.

Просматривая существующие интернет-витрины очевидно, что большинство сайтов упускают такую приятную для пользователей часть как темный дизайн оформления, который набирает популярность в последнее время. Дизайн и масштаб интернет-магазинов изучен. Настало время рассмотреть среды разработки, в которых будет создаваться непосредственно сам шаблон.

1.2. Сравнительная характеристика и выбор среды разработки

На сегодняшний день существует большое количество редакторов кода и сред разработки. Они обладают большим функционалом и способствуют ускорению написания кода. Ниже проведены положительные стороны самых популярных сред разработки и редакторов кода для поставленной задачи.

Рассмотрим редактор кода Atom от GitHub.

Текстовый редактор Atom хорош уже в базовой конфигурации, а дополнительные расширения могут сделать его идеальным. Встроенный пакетный менеджер облегчает поиск, установку и создание модулей. По умолчанию установлены 8 тем оформления, а также есть доступ к темам, созданным другими участниками сообщества. Основные возможности:

- умное автодополнение;
- удобный файловый менеджер;
- разделение рабочей области для удобства анализа нескольких файлов благодаря многопанельному интерфейсу;
- удобные механизмы поиска внутри проекта.

Редактор кода Visual Studio Code.

Удобный и быстрый редактор, с которым работать – одно удовольствие. Он не привязан к конкретной ОС, нетребователен к ресурсам компьютера и имеет возможность устанавливать дополнительные расширения. Бесплатная базовая версия предоставляет всю необходимую функциональность:

- подсветка синтаксиса распространенных языков;
- автодополнение;
- множественное выделение (возможность поставить указатель курсора в нескольких частях кода и печатать в них одновременно);

- фолдинг (возможность сворачивать блоки кода);
- удобные комбинации горячих клавиш;
- возможность разделить рабочую область на несколько окон;
- большой выбор бесплатных тем интерфейса от сторонних разработчиков.

Рассмотрим IDE от JetBrains. WebStorm.

Компания создает специализированные среды разработки для разных языков и задач. WebStorm – одна из них. Представляет из себя IDE ориентированную на создание веб-сайтов и веб-приложений. Достоинства подобных IDE для веб-разработки в том, что они мощны и ориентированы под конкретный язык со всеми особенностями синтаксиса и логики. Это интеллектуальные редакторы с широчайшими возможностями отладки кода, тестирования и рефакторинга кода. Все IDE (англ. Integrated Development Environment) позволяют легко перемещаться по проекту и поддерживают работу с системами контроля версий. [10]

Из вышеперечисленных вариантов лучше выбрать WebStorm. Данная IDE обладает большим функционалом, нежели редакторы кода, а также создана для разработчиков ориентирующихся на создании веб-приложений и веб-сайтов. После определения среды разработки необходимо продумать варианты к подходу программирования, определению библиотеки для разработки веб-приложений и задуматься над дизайном интернет-витрины.

1.3. Анализ подхода к созданию интернет-витрины для SPA

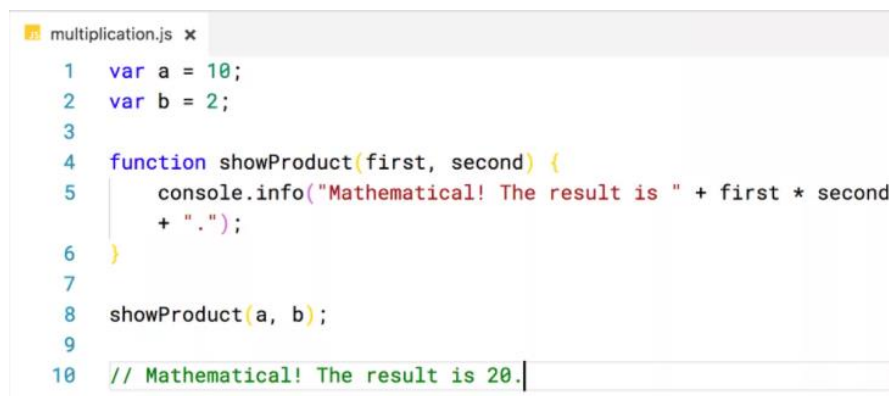
1.3.1. Выбор языка программирования для написания логики интернет-витрины

В наши дни разработка одностраничных веб-приложений имеет различные подходы. Зачастую, в качестве создания клиентской стороны приложения используются JavaScript и TypeScript. Эти языки типичны для создания веб-ресурсов. Перейдем к рассмотрению данных языков.

JavaScript – это язык сценариев (скриптов) на странице в вашем браузере. JS имеет большую историю и изначально назывался LiveScript. Язык взял за основу синтаксис других языков – C и Java. При открытии пользователем браузера и запросе нужной страницы, этот запрос отправляется на сервер. Сервер отправляет в браузер простые HTML и CSS файлы. ОС содержит браузер, и этот браузер

содержит страницу, а страница состоит из JS DOM структуры, поэтому она работает на сервере. Такие браузеры, как Safari, Opera и Chrome, содержат движок JS, а именно V8. Нужен он для того, чтобы браузер понимал JS и реагировал на события, прописанные в сценариях (скриптах). Однако, сам JS не поддерживает чтение и запись в файлы. [5]

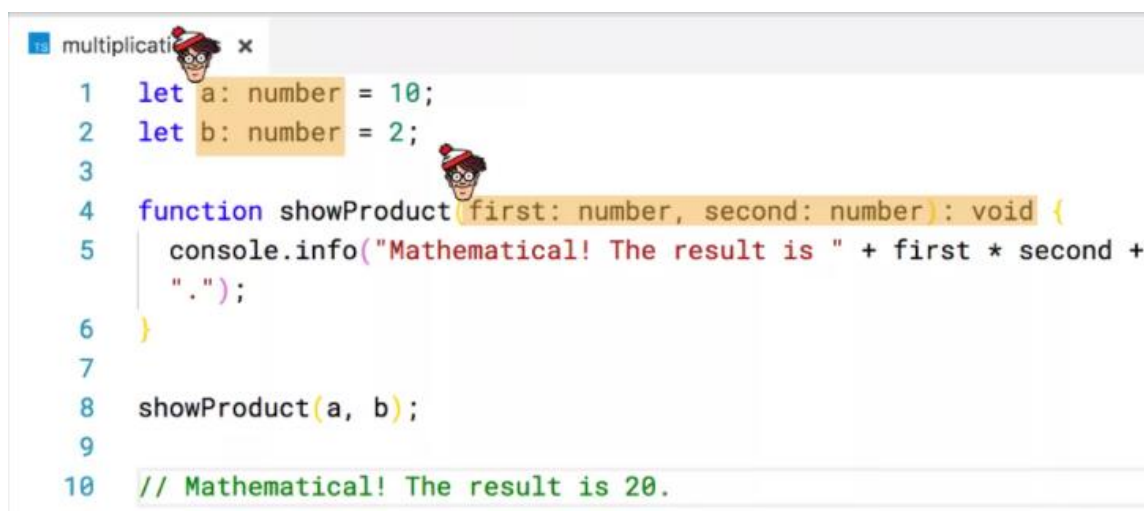
TypeScript – это расширенный JS, разработанный компанией Microsoft и появившийся в 2012 году. Он имеет весь функционал JS, а также использует собственный компилятор для преобразования файла TS в файл JS. TS легко интегрируется в проекты, которые изначально разработаны на JS. TS также обеспечивает проверку статического типа. Это позволяет программисту проверять и назначать переменные и типы функций. Эта функция упрощает чтение кода и предотвращает ошибки. TS имеет такие типы данных, как String, Number, Boolean, Null, Array, Enum, Tuple и Generics. На рисунках 4 и 5 отображена наглядная разница между JS и TS. [6]



```

multiplication.js x
1  var a = 10;
2  var b = 2;
3
4  function showProduct(first, second) {
5      console.info("Mathematical! The result is " + first * second
6          + ".");
7  }
8  showProduct(a, b);
9
10 // Mathematical! The result is 20.
  
```

Рисунок 4 – Пример кода на JS



```

multiplicati x
1  let a: number = 10;
2  let b: number = 2;
3
4  function showProduct(first: number, second: number): void {
5      console.info("Mathematical! The result is " + first * second +
6          ".");
7  }
8  showProduct(a, b);
9
10 // Mathematical! The result is 20.
  
```

Рисунок 5 – Пример кода на TS

Для того, чтобы проследить различия между данными языками, можно просмотреть Таблицу 2.

Таблица 2. Сравнение JS и TS

Язык программирования / Возможности языка программирования	JavaScript	TypeScript
Типизация	Динамическая	Статическая
Скорость	Быстрый	Производительный
Эффективность	Малоэффективен	Эффективен
Простота изучения	Легок в освоении	Сложен в освоении
Скорость изучения другого, например низкоуровневый, языка после изучения данного	Долго	Не требует много времени
Возможность избежать ошибки до включения приложения	Нет	Да
Возможность создавать приложения не только для интернета	Нет	Да

Проанализировав таблицу 2 видно, что TS значительно преуспевает в большинстве аспектов для программирования. Хотя TS и основан на классах, он способен поддерживать наследование, инкапсуляцию и модификатор как объектно-ориентированный язык программирования, он позволяет пользоваться функциональным программированием. Также TS имеет все те же методы и встроенные функции, что и JS. Это позволяет легче переключиться на него программистам, хорошо знающим методы и функции JS.

Главное преимущество TypeScript в том, что он безопаснее с точки зрения защиты кода. Возможность создания интерфейсов для объектов также позволяет оптимизировать код и ускорить работу всего приложения. В целом код мало отличается от JavaScript, но важно то, что типы переменных строго фиксированы. Выбрав язык написания логики для фильтрации товаров можно задуматься о подходе к написанию стилей.

1.3.2. Выбор формального языка стилей веб-страниц

На сегодняшний день можно выбирать между подходами к разработке стилей для веб-ресурсов. Для начала следует ознакомиться с определениями.

CSS (англ. Cascading Style Sheets «каскадные таблицы стилей») — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL.

SASS (англ. Syntactically Awesome Stylesheets «синтаксически прекрасные стили») — модуль, включенный в Haml. SASS — это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей.

Язык SASS имеет два синтаксиса:

- SASS — отличается отсутствием фигурных скобок, в нем вложенные элементы реализованы с помощью отступов;
- SCSS — использует фигурные скобки, как и сам CSS.

В выборе подхода следует определиться с масштабом проекта. Если проект невелик, то следует использовать простой CSS. В случае, если проект большой и требует дополнительную поддержку, то препроцессоры являются лучшим решением, так как помогут сократить затраченное время до 50%. [4]

Выбирая же между препроцессорами, стоит обратить внимание на простоту использования. В этом смысле SCSS гораздо удобнее, потому что имеет синтаксис как в CSS. Определившись с подходом к стилизации, следует узнать о возможностях создания одностраничных веб-приложений.

1.3.3. Выбор библиотеки для создания SPA

Самой надежной и популярной библиотекой для разработки одностраничных веб-приложений на сегодняшний день является React. У React есть аналоги, такие как Angular и Vue. Эти библиотеки позволяют создавать одностраничные веб-приложения.

Для работы с каждой из этих библиотек необходимо иметь на компьютере Node.js у которого имеется пакетный менеджер NPM. С его помощью можно создать папку внутри которой будет находиться проект в виде, задуманном разработчиками (рисунок 1, 2, 3).

Angular — это среда JS Model-View-ViewModel (далее MVVM), основанная в 2009 году, которая отлично подходит для создания интерактивных веб-приложений.

Начальный вид веб-приложения на Angular представлен на Рисунке 1.

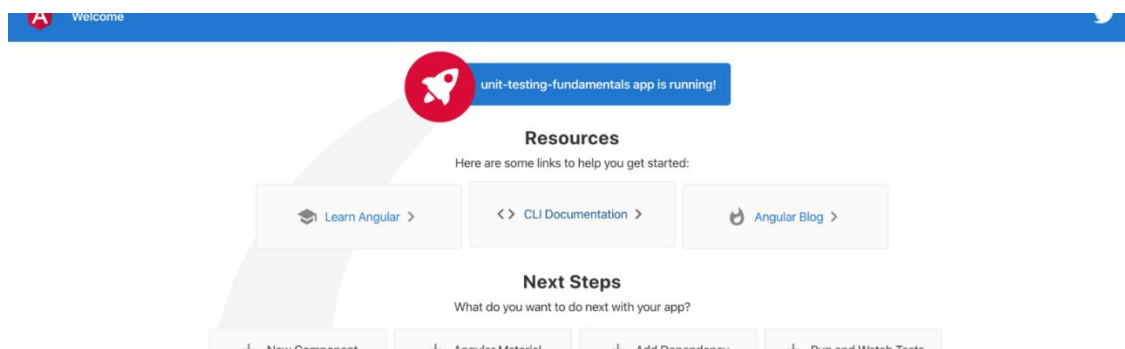


Рисунок 1 – Начальный вид веб-приложения на Angular

О Model-View-ViewModel. Шаблон MVVM имеет три основных компонента:

- 1) Модель, которая представляет бизнес-логику приложения.
- 2) Представление пользовательского интерфейса.
- 3) Представление-модель, в котором содержится вся логика построения графического интерфейса и ссылка на модель, поэтому он выступает в качестве модели для представления.

Положительные стороны Angular:

- Angular используется вместе с TS, он имеет исключительную поддержку для этого; [13]
- Angular-language-service — обеспечивает интеллектуальные возможности и автозаполнение шаблона HTML-компонента; [13]
- новые функции, такие как generation Angular, использующие библиотеки NPM из CLI (терминал), generation, и разработка компонентов, использующая Angular; [13]
- подробная документация, позволяющая разработчику получить всю необходимую информацию, не прибегая к помощи его коллег. Однако это требует больше времени для обучения; [13]
- односторонняя привязка данных, которая обеспечивает исключительное поведение приложения, что сводит к минимуму риск возможных ошибок; [13]

- MVVM, которая позволяет разработчикам работать отдельно над одним и тем же разделом приложения, используя один и тот же набор данных; [13]
- внедрение зависимостей от компонентов, связанных с модулями и модульностью в целом; [13]
- структура и архитектура, специально созданные для большой масштабируемости проекта. [13]

Негативные стороны Angular:

- разнообразие различных структур (Injectables, Components, Pipes, Modules и т. д.) усложняет изучение по сравнению с React и Vue.js, у которых есть только «Component»; [13]
- относительно медленная производительность, учитывая различные показатели. С другой стороны, это можно легко решить, используя так называемый «ChangeDetectionStrategy», который помогает вручную контролировать процесс рендеринга компонентов. [13]

Компании, использующие Angular: Microsoft, Autodesk, MacDonald's, UPS, Apple, Adobe, GoPro, YouTube, Paypal, Nike, Google, Telegram и многие другие.

React — это библиотека JavaScript, разработанная Facebook в 2013 году, которая отлично подходит для создания современных одностраничных приложений любого размера и масштаба.

Начальный вид веб-приложения на React представлен на Рисунке 2. [2]

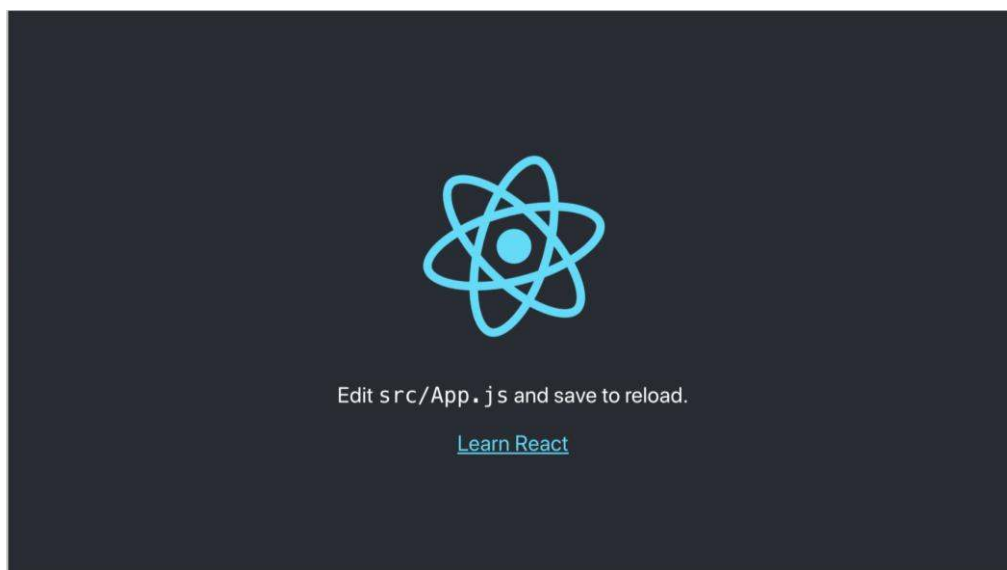


Рисунок 2 – Начальный вид веб-приложения на React

Положительные стороны React:

- легко изучить, благодаря простому дизайну, использованию JSX (HTML-подобный синтаксис) для шаблонов и очень подробной документации. Разработчики тратят больше времени на написание современного JavaScript и меньше беспокоятся о коде, специфичном для библиотеки; [13]
- очень быстрая, благодаря реализации React Virtual DOM и различным оптимизациям рендеринга; [13]
- отличная поддержка рендеринга на стороне сервера, что делает его мощной платформой для контент-ориентированных приложений; [13]
- первоклассная поддержка Progressive Web App (PWA) благодаря генератору приложений `create-react-app`; [13]
- привязка данных является односторонней, что означает меньше нежелательных побочных эффектов; [13]
- Redux, самая популярная платформа для управления состоянием приложений в React, ее легко учить и использовать; [13]
- React реализует концепции функционального программирования (FP), создавая простой в тестировании и многократно используемый код; [13]
- приложения могут быть созданы с помощью TS, имеющими встроенную поддержку JSX; [13]
- переход между версиями, как правило, очень прост: Facebook предоставляет «кодовые модули» для автоматизации большей части процесса; [13]
- навыки, полученные в React, могут быть применены к разработке на React Native с помощью которого можно разрабатывать приложения для мобильных устройств. [13]

Негативные стороны React:

- сообщество делится по способам написания CSS в React, которые разделяются на традиционные таблицы стилей (CSS Modules) и CSS-in-JS (т.е. Emotion и Styled Components);
- React отходит от компонентов на основе классов, что может стать препятствием для разработчиков, которым более комфортно работать с ООП;

– React не однозначен и оставляет разработчикам возможность выбирать лучший способ развития. Это может быть решено сильным лидерством проекта и хорошими процессами; [13]

– смешивание шаблонов с логикой (JSX) может сбить с толку некоторых разработчиков при первых знакомствах с React. [13]

Компании, использующие React: Facebook, Instagram, Netflix, Whatsapp, Dropbox, Microsoft и другие.

Vue.js — это JS-библиотека, основанная в 2013 году, которая идеально подходит для создания высокоадаптируемых пользовательских интерфейсов и сложных одностраничных приложений. [13]

Начальный вид веб-приложения на Vue представлен на Рисунке 3.

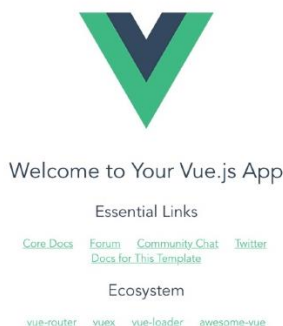


Рисунок 3 – Начальный вид веб-приложения на Vue

Положительные стороны Vue.js:

– может быть осуществлен быстрый переход от других библиотек к Vue.js из-за сходства с Angular и React с точки зрения дизайна и архитектуры; [13]

– Vue.js можно использовать как для создания одностраничных приложений, так и для более сложных веб-интерфейсов приложений. Важно, что небольшие интерактивные элементы можно легко интегрировать в существующую инфраструктуру без негативных последствий; [13]

– наличие усиленного HTML означает, что Vue.js имеет много характеристик схожих с Angular, а это, благодаря использованию различных компонентов, помогает оптимизации HTML- блоков; [13]

- Vue.js имеет очень подробную документацию, которая может ускорить процесс обучения для разработчиков и сэкономить много времени на разработку приложения, используя только базовые знания HTML и JavaScript; [13]

- Vue.js может помочь в разработке довольно больших шаблонов многократного использования, которые могут быть сделаны почти за тоже время, что и более простые; [13]

- Vue.js весит около 20 КБ, сохраняя при этом свою скорость и гибкость, что позволяет достичь гораздо лучшей производительности по сравнению с другими платформами. [13]

Негативные стороны Vue.js:

- иногда у Vue.js могут возникнуть проблемы при интеграции в огромные проекты, и пока еще нет опыта возможных решений, но они обязательно появятся в ближайшее время; [13]

- Vue.js по-прежнему занимает довольно небольшую долю рынка по сравнению с React или Angular, что означает, что обмен знаниями в этой среде все еще находится на начальной стадии. [13]

Компании, которые используют Vue.js: Xiaomi, Alibaba, WizzAir, Gitlab, Laracasts, Adobe и другие.

Каждая из вышеперечисленных библиотек поддерживает разработку на TypeScript, однако, лучше отдать предпочтение React, так как он имеет важное отличие, а именно – JSX. JSX (или TSX) позволяет писать теги из HTML внутри JS (или TS) кода. Помимо этого, React позволяет пользоваться функциональным программированием, что также облегчает написание кода. В дополнение к перечисленному, React позволяет пользоваться стилистическими модулями. При их использовании можно задавать одинаковое имя классов в разных частях проекта, но при компиляции названия будут отображаться с дополнительно сгенерированными ключами.

1.3.4. Выбор библиотеки для написания бизнес-логики веб-приложения

Существует немало менеджеров состояний для веб-приложений. Основные – Redux и MobX. Эти библиотеки служат для управления состояниями компонентов

и их данных. В настоящее время менеджеры состояний также содержат в себе всю логику (алгоритмы) веб-приложений. Обе библиотеки могут использоваться совместно с Angular, React и Vue. Можно в любое время перейти от одного подхода создания бизнес-логики к другому. Различия данных библиотек приведены ниже.

О Redux. Redux является производным архитектуры Flux, но в отличие от данной архитектуры, использует одно глобальное хранилище поверх нескольких для сохранения состояний. Вместо диспетчера в Redux используются чистые функции. Для большинства разработчиков, использующих объектно-ориентированное программирование, использование и понимание функционального программирования весьма затруднительно. Состояние в Redux является неизменным. Для того, чтобы изменить состояние всегда возвращается новое состояние. Все сущности могут ссылаться друг на друга как в базе данных по ID и это лучшая практика. Единственная сущность с состояниями – источник правды. Многочисленные редукторы позволяют изменять неизменяемое состояние.

О MobX. На данную библиотеку влияет не только объектно-ориентированное программирование, но и реактивное программирование. Состояние оборачивается в наблюдаемый объект. Наблюдаемое значение позволяет получать обновления после изменения данных. Состояние в MobX изменчиво, таким образом состояние изменяется напрямую. Состояние остается денормализованным и имеет большую вложенность. MobX использует несколько хранилищ.

Для новичка, несомненно, будет проще воспользоваться Redux, так как функциональное программирование гораздо понятнее. Она маленькая и большую часть времени придется сталкиваться только с простыми объектами и массивами JS. Это ближе к простому JS, чем MobX, поэтому легче тестировать и отлаживать приложение. Также преимуществом Redux является его использование на стороне сервера. Это значит, что можно подключиться к серверу взаимодействовать с данными оттуда. MobX же строится на абстракциях, из-за этого труднее понимать его основные механизмы, хоть он и является действительно альтернативой Redux. Теперь необходимо понять разницу между веб-сайтом и веб-приложением, а также взглянуть на их преимущества и недостатки.

1.3.5. Обоснование необходимости разработки интернет-витрины для SPA

Веб-приложение – это интерактивные компьютерные приложения, разработанные для сети интернет, позволяющие пользователям вводить, получать и манипулировать данными с помощью взаимодействия. Такие программы обычно имеют очень тесную связь с сервером, отправляя на него множество запросов. Могут быть встроены в веб-страницы, либо сами веб-страницы могут быть приложениями. Например, Facebook, Gmail, YouTube, Ebay, Twitter, Amazon.

Веб-приложения используют имя пользователя и пароль для аутентификации. Позволяют своим посетителям обмениваться мгновенными сообщениями (чат-платформы, соцсети, блоги), создают контент на основе пользовательских предпочтений, обеспечивают к нему неограниченный доступ, используют мини-встроенные программы для развлечений.

Основные отличия:

- веб-сайт является источником информации, в то время как веб-приложение работает в интерактивном режиме;
- функции, задачи, пользовательский интерфейс веб-приложения гораздо сложнее;
- веб-приложение может быть составной частью сайта либо отдельным ресурсом;
- разработка веб-сайта является более легким, недорогим проектом. Для разработки веб-приложения требуются большая вычислительная мощность, а также различные платформы и инструменты;
- веб-приложение является более ресурсоемким, так как может взаимодействовать с пользователем и выполнять различные действия.

В современном мире интернета почти не осталось сайтов, которые не имели бы никакого элемента интерактивности. И наоборот, многие веб-приложения часто включают в себя поиск информации. Тем не менее, веб-сайты по-прежнему являются информационными источниками, а веб-приложения остаются пользовательскими инструментами.

SPA — это веб-сайт, который постепенно смещает со своих позиций классические многостраничные сайты. Так происходит из-за того, что они более

простые в разработке, быстрее работают, имеют больший потенциал в интерактивности и нравятся пользователям. Однако у них есть слабое место — SEO-оптимизация. Пока еще не все браузеры могут с ними нормально работать, поэтому, чтобы сделать такие приложения дружественными для SEO, нужно прибегать к ряду ухищрений. Классические-сайты в этом плане более простые и надежные. К сожалению, отечественные интернет-магазины на основе SPA пока не встречаются на нашем рынке. Существует лишь пара зарубежных случаев, а именно: Ebay и Amazon.

Большим недостатком веб-приложения является худший его анализ SEO. SEO (англ. Search Engine Optimization) – процесс структуризации и организации сайта для повышения трафика на веб-ресурсе, путем повышения его положения и частоты появления на первой странице поисковых сервисов. Учитывая то, что процесс автоматизирован, содержимое должно быть структурировано и форматировано способом, с помощью которого машина будет понимать это самое содержимое.

Проблема заключается в том, что SPA представляет собой JavaScript-программу, работающую в браузере. В случае, если машина, встроенная в поисковую систему, не умеет исполнять скрипты, она не сможет получать все «страницы» и их содержимое. Однако имеется 2 главных решения данной проблемы:

- 1) Изоморфный (универсальный) SPA.
- 2) Пререндеринг.

Изоморфный сайт автоматически определяет отключен ли JS у пользователя или нет. В случае, если JS отключен, все скрипты выполняются на стороне сервера и по результату поисковик получает статические HTML и CSS файлы, которые отправляются клиенту. При таком раскладе все необходимые для SEO атрибуты присутствуют во время загрузки. В другом варианте все происходит стандартно, как и должно, рендеринг страниц и компонентов происходит на стороне клиента. Этот подход является лучшим, но на сегодняшний момент очень трудно его реализовать.

Достоинствами данного подхода являются следующие его особенности:

- пререндер способен выполнять все типы современного JavaScript и выдает статический HTML;
- пререндер поддерживает самые последние веб-нововведения;
- требуется минимальная модификация кодовой базы веб-приложения, если вообще таковая потребуется;
- снижаются затраты на разработку и поддержку веб-сайта;
- простота имплементации.

Недостатки данного подхода:

- мало пригоден для страниц, которые отображают часто изменяющиеся данные;
- не приемлем для страниц, которые содержат персональные данные пользователей. Впрочем, такие страницы мало полезны для SEO и не должны индексироваться;
- в случае большого сайта со значительным количеством страниц, процесс пререндеринга может занимать значительное время;
- пререндер-сервисы платные.

Пререндеринг представляет собой процесс предварительного рендеринга страниц и компонентов веб-приложения с целью их дальнейшего просмотра машиной анализа. Существует сервис для предоставления такой возможности SPA. Этот сервис перехватывает запросы машины к веб-приложению и определяет ее. Когда он определяет, что к сайту обращается именно машина, то он заранее генерирует обычную страницу с всем содержанием SPA. Используется данный подход и со старыми браузерами, не имеющими подобных машин.

Наличие данной проблемы несомненно сказывается на притоке пользователей, однако, существующие решения проблемы помогают легко справиться с ней для продвижения сайта в поисковиках.

Подводя итог по данному разделу, можно отметить, что благодаря динамическому обновлению с помощью JavaScript или же TypeScript, во время использования не нужно перезагружать или подгружать дополнительные страницы. На практике это означает, что пользователь видит в браузере весь основной контент, а при прокрутке или переходах на другие страницы, вместо

полной перезагрузки нужные элементы просто подгружаются. В процессе работы пользователю может показаться, что он запустил не веб-сайт, а приложение на компьютере, так как оно мгновенно реагирует на все его действия, без задержек. Такого эффекта удастся добиться с помощью продвинутых библиотек: Angular, React, Vue. Примеры динамических приложений: Gmail, Google Maps, Facebook, GitHub. Далее приступаем к поэтапной разработке шаблона.

2. РАЗРАБОТКА ШАБЛОНА WEB-РЕСУРСА

2.1. Этапы разработки дизайна интернет-витрины на сайте Figma

Для начала следует определиться с интерфейсом. Разработка любого сайта начинается в иерархическом порядке. От родительского компонента к дочернему. Начало разработки макета нужно начинать с общего внешнего вида интернет-витрины. После создания общего внешнего вида нужно приступить к созданию стилей для дочерних блоков. Весь процесс происходит поэтапно и иногда занимает немало времени. Как было упомянуто выше, большинство интернет-магазинов забывают о таком важном моменте как «Темная тема». Темный интерфейс набирает популярность в последнее время и все больше сайтов и приложений применяют данный стиль. Расположение блоков также должно быть понятным для пользователя.

На рисунках 4-6 представлены этапы создания интерфейса интернет-витрины. Увидеть «скелет» представленной витрины можно на Рисунке 4.

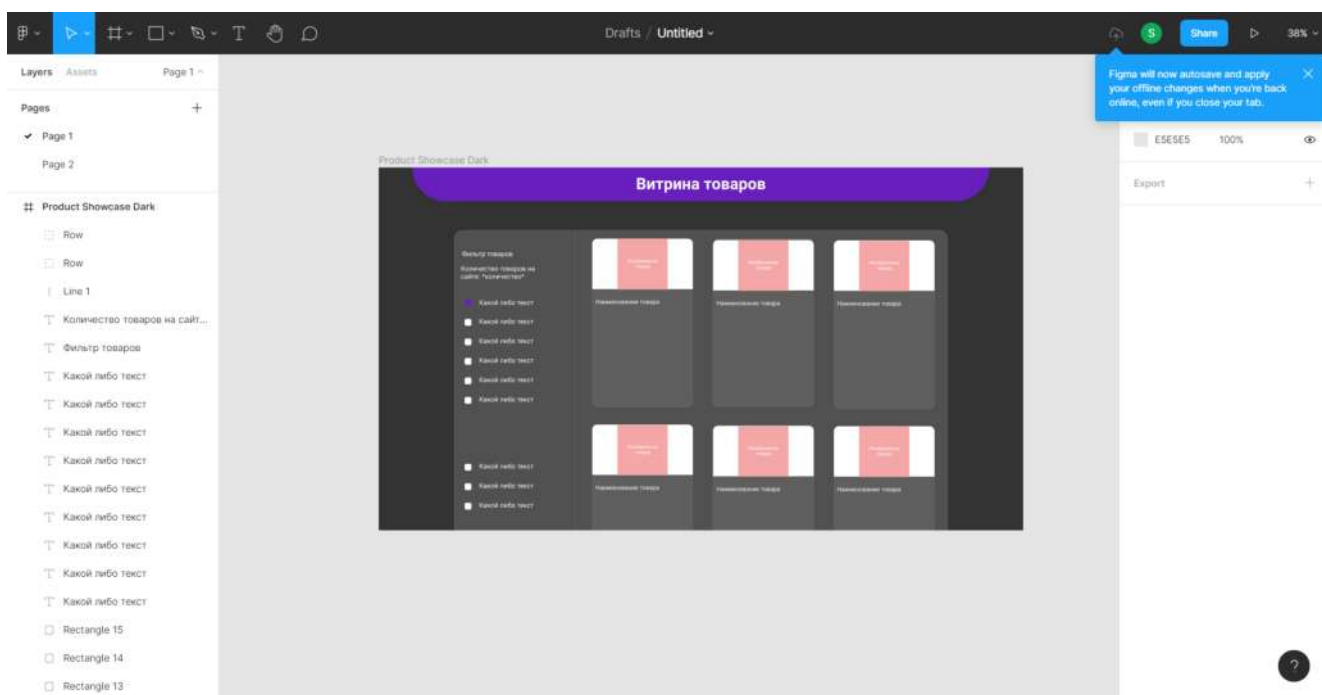


Рисунок 4 – Создание дизайна витрины в веб-приложении Figma

Помимо дизайна витрины, также важно разработать страницу представления всей информации о товаре. За основу расположения элементов была взята страница просмотра товара сайта Яндекс.Маркет.

Данную страницу можно увидеть на рисунке 5.

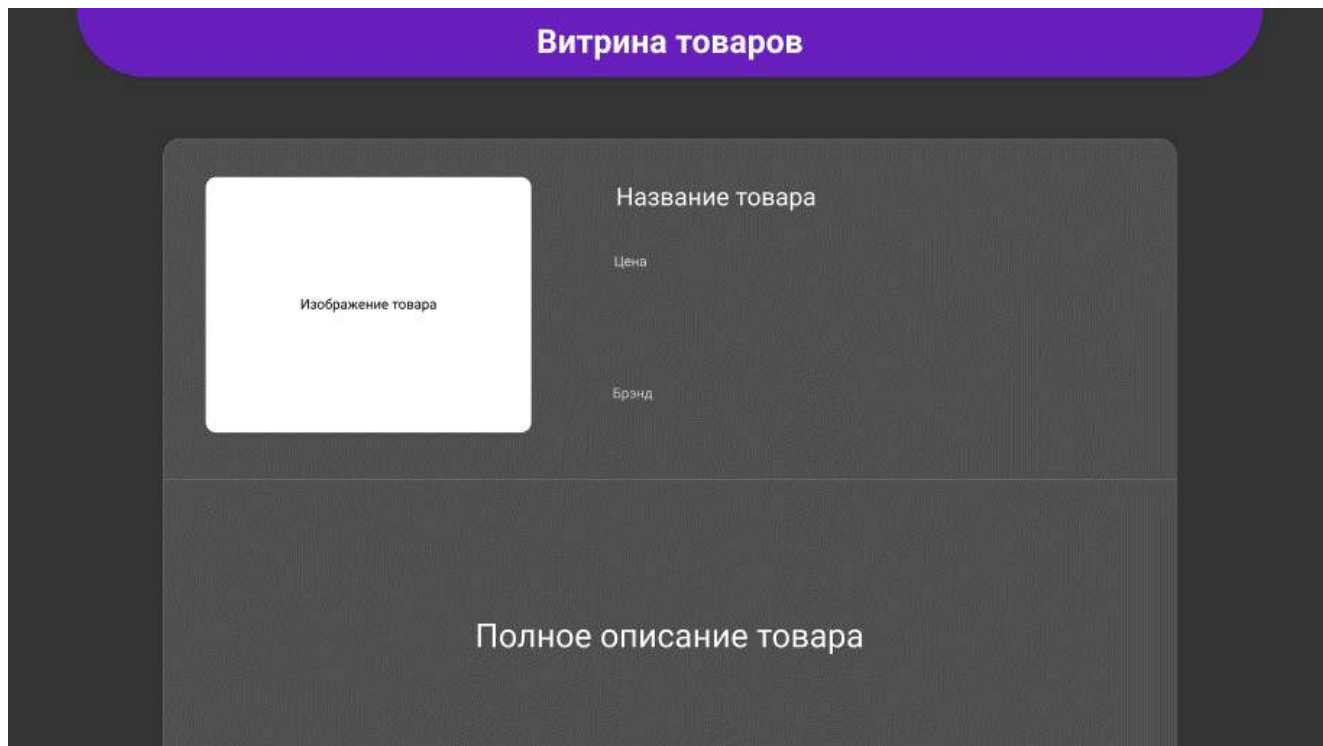


Рисунок 5 – Дизайн страницы «Страница просмотра товара»

Для приведенного выше стиля витрины был создан дизайн кнопок фильтрации товара.

Рисунок 6 показывает, как выглядит данная кнопка.

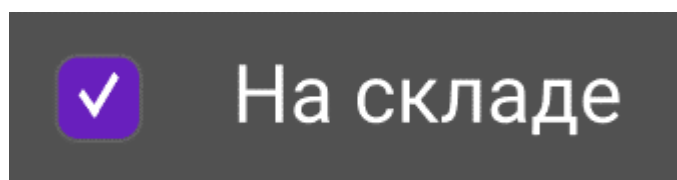


Рисунок 6 – Дизайн страницы «Кнопка фильтра»

Созданный дизайн прост и приятен в восприятии интернет-магазина. Решение использовать темные тона также приятно скажется на потоке пользователей. После создания интерфейса для интернет-витрины и ее компонентов можно приступать к написанию кода в среде разработки WebStorm.

2.2. Написание компонентов, стилей и логики фильтрации товаров в среде разработки WebStorm

На рисунках 7-14 показан код файлов из компонентов интернет-витрины.

После разработки дизайна следует приступить к написанию кода веб-приложения и стилей. Ниже приведены скриншоты с «отрывками» кода. Тут

показано как выглядит написание кода с использованием JSX в React и написание стилей с помощью SCSS, а также скриншоты стороны логики приложения, в которой написан алгоритм отслеживания выбранных фильтров и их последующего отображения в веб-приложении – Redux.

Скриншоты кода компонентов:

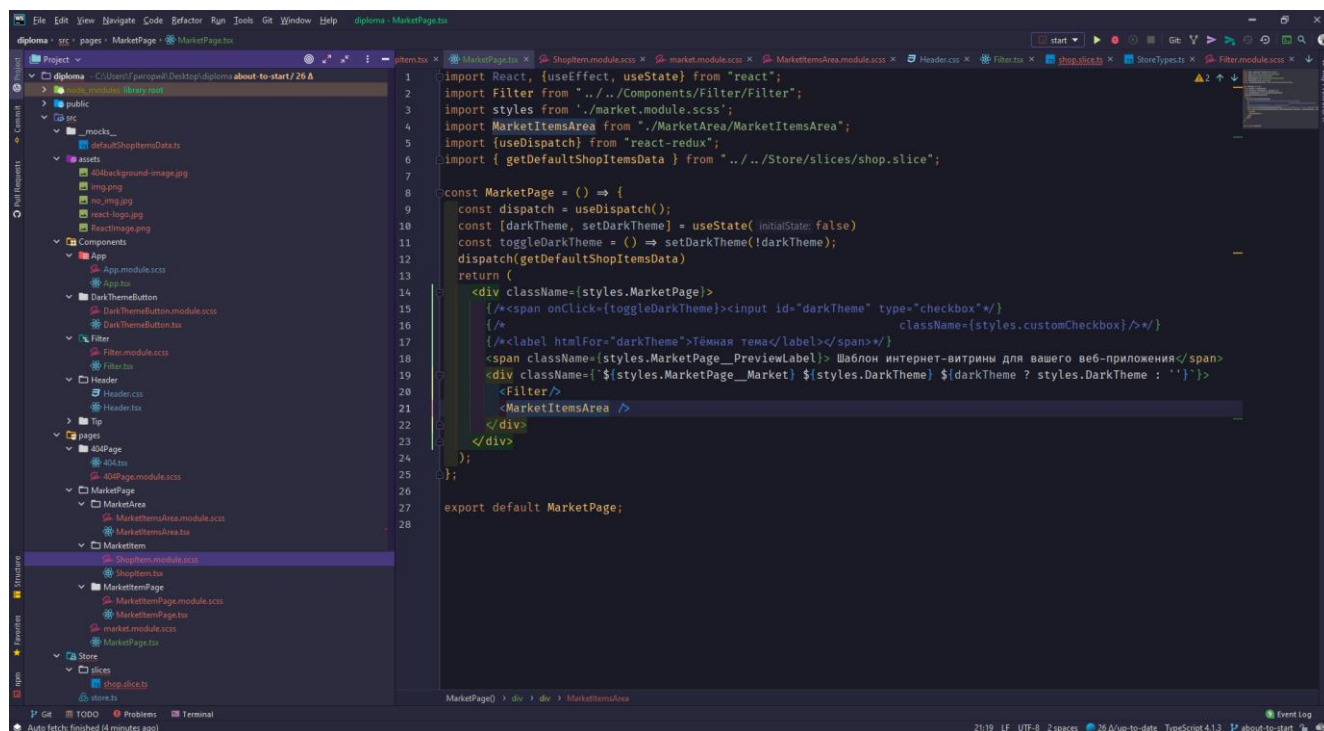


Рисунок 7 – Код компонента MarketPage.tsx

На рисунке 7 показано как создаются компоненты в React с помощью функционального программирования (здесь используется синтаксис JS, а именно стрелочные функции), их экспортирования и способность импортирования сторонних функций и модулей. Также видно, как комментируется код. Дополнительно, здесь отображено взаимодействие с менеджером состояний Redux. Посредством встроенных функций в React и дополнительной библиотеки react-redux. Еще можно обратить внимание на то, что при изменении состояния в менеджере состояний существует возможность интерактивно дописывать классовые имена блокам. [8]

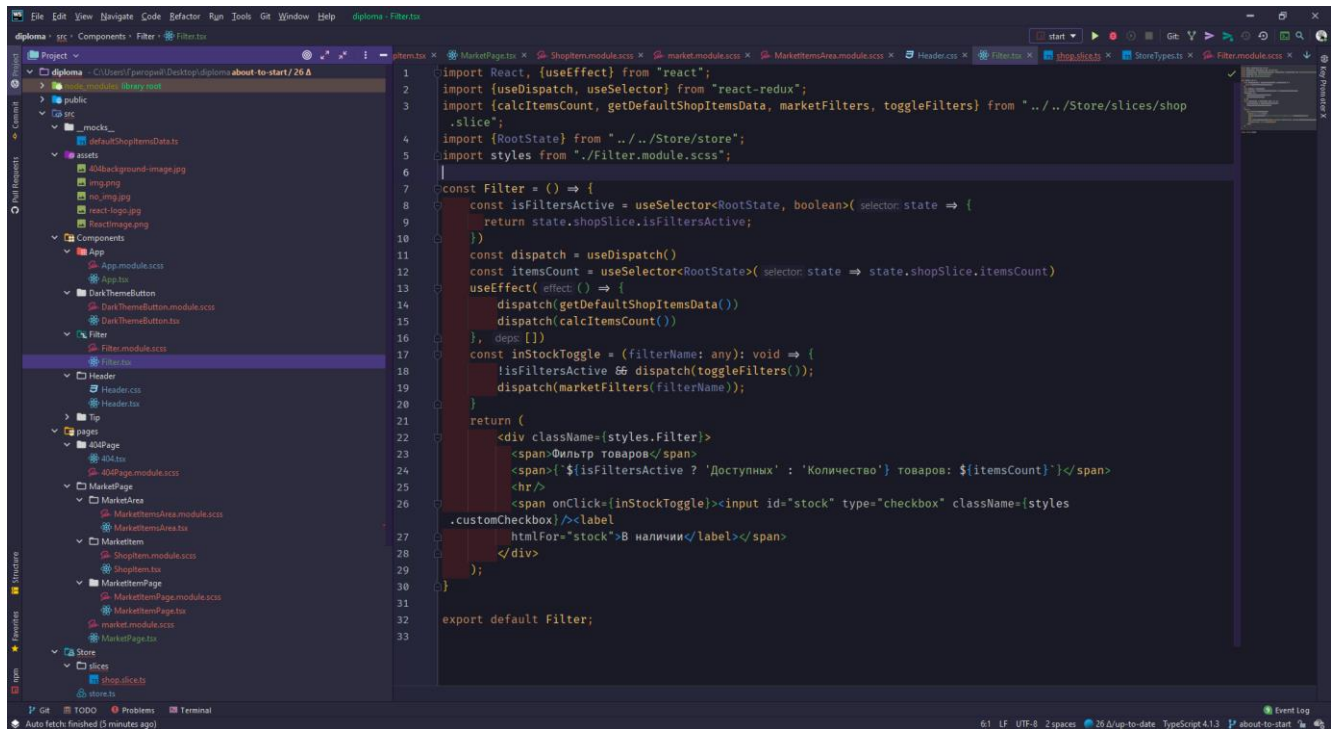


Рисунок 8 – Код компонента Filter.tsx

На рисунке 8 отображено, каким образом нажатие по кнопке фильтра отправляет данный в менеджер состояний, а также как применяется функция «useEffect», который запускает вписанные в него операции по загрузке компонента в веб-приложение. [7]

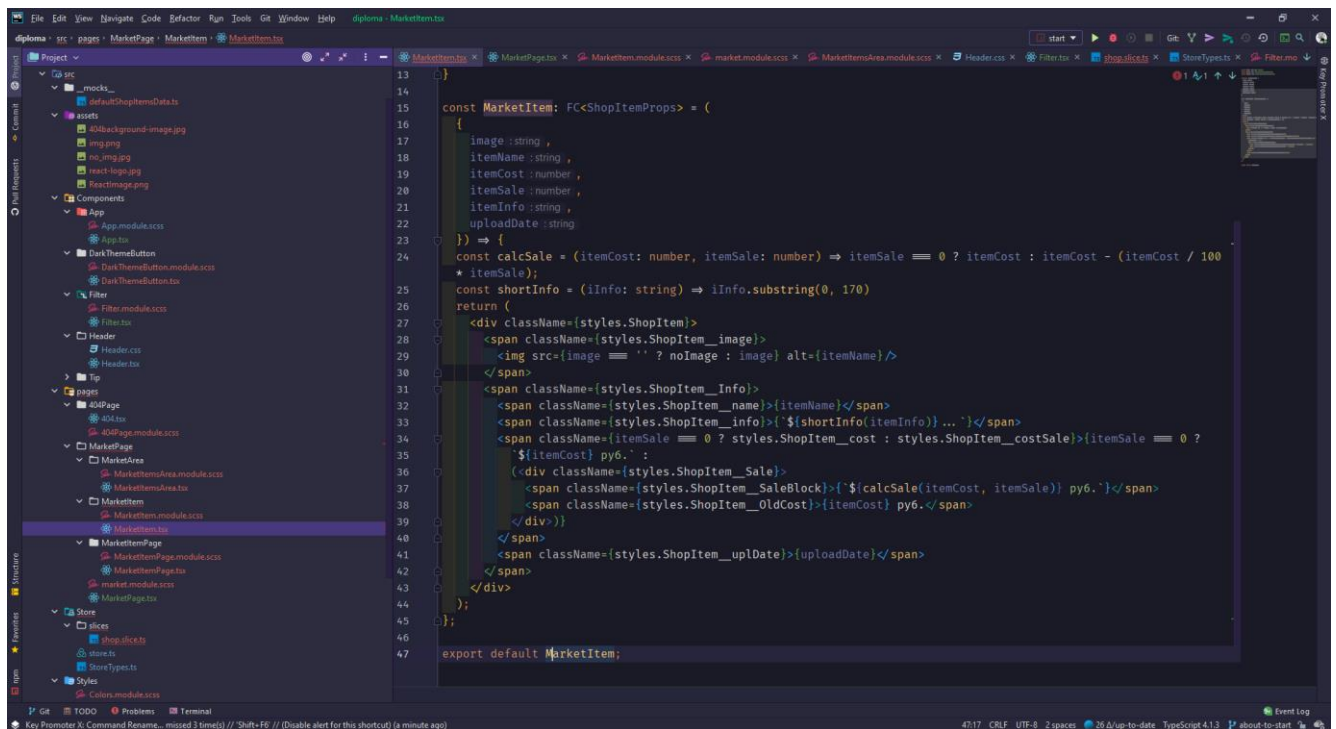


Рисунок 9 – Код компонента MarketItem.tsx

На рисунке 9 написана структура компонента товара и взаимодействие его с бизнес логикой. В случае, если у товара имеется скидка, производится замена одного блока на другой и высчитывается цена, учитывающая скидку. Также отображается лишь часть содержимого для того, чтобы не засорять витрину огромным количеством текста.

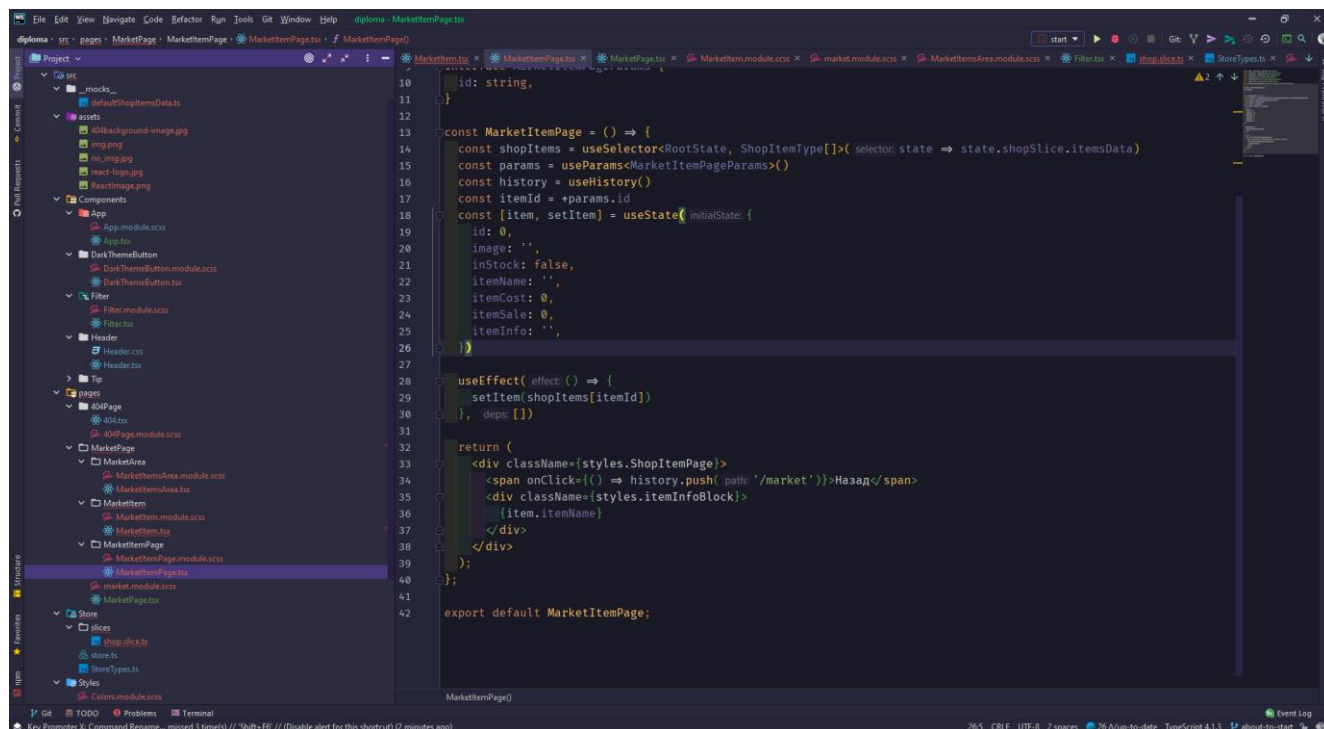


Рисунок 10 – Код компонента MarketItemPage.tsx

На рисунке 10 представлена структура элемента товара, а конкретно имеющиеся поля: id, картинка, имеется ли на складе, название товара, цена товара, скидка на товар и подробная информация о товаре. [8]

Еще в данном компоненте есть кнопка «Назад», которая отправляет запрос в браузер с переходом на указанную ссылку. То есть возвращает пользователя на страницу интернет-витрины. Ниже представлены скриншоты кода стилей на Рисунок 11 – Стилизация товара внутри витрины с помощью SCSS

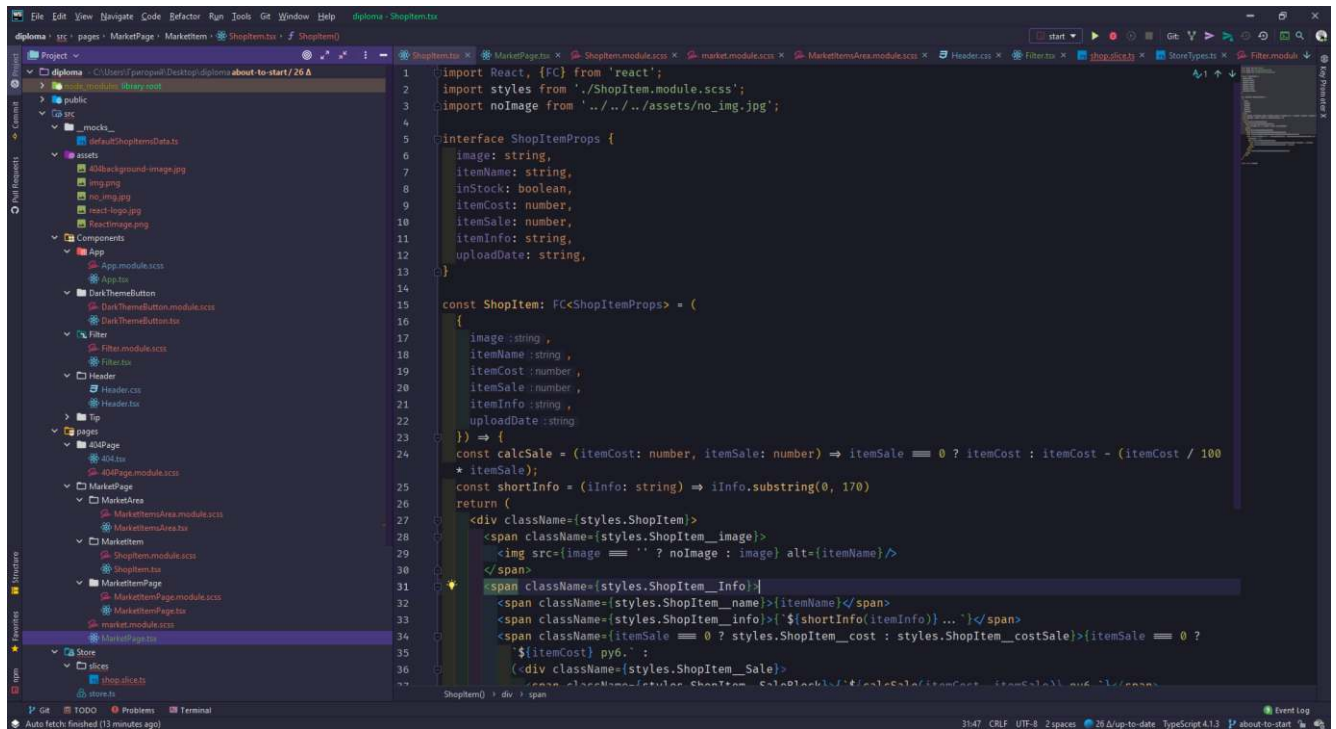


Рисунок 11 – Стилизация товара внутри витрины с помощью SCSS

На рисунке 11 на второй линии кода показано как стили импортируются в компонент React. Таким образом, осуществляется стилизация блоков компонента. Не обязательно использовать styles для того, чтобы указать библиотеке React на использование стилей SCSS. Можно использовать и другие названия, просто данная практика является более понятной для дальнейшей поддержки кода. [14]

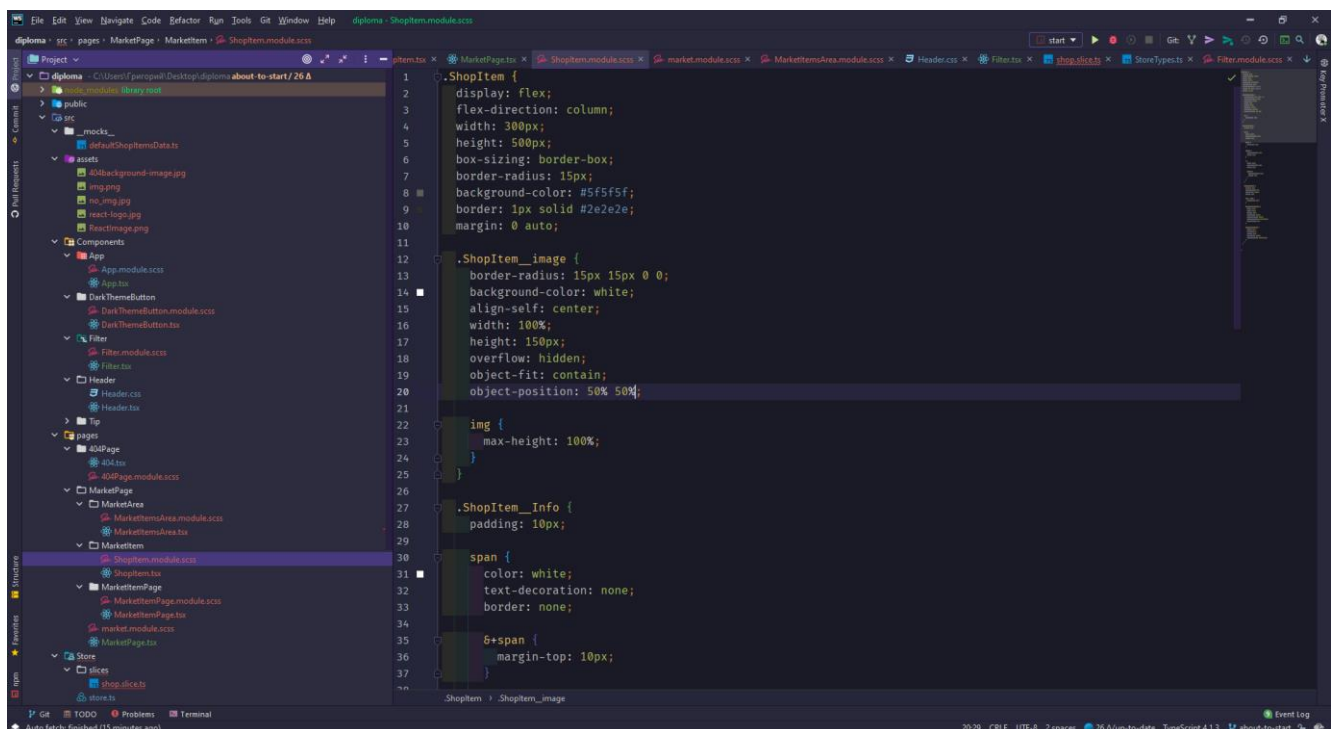


Рисунок 12 – Стилизация страницы товара с помощью SCSS

На рисунке 12 показан синтаксис SCSS.

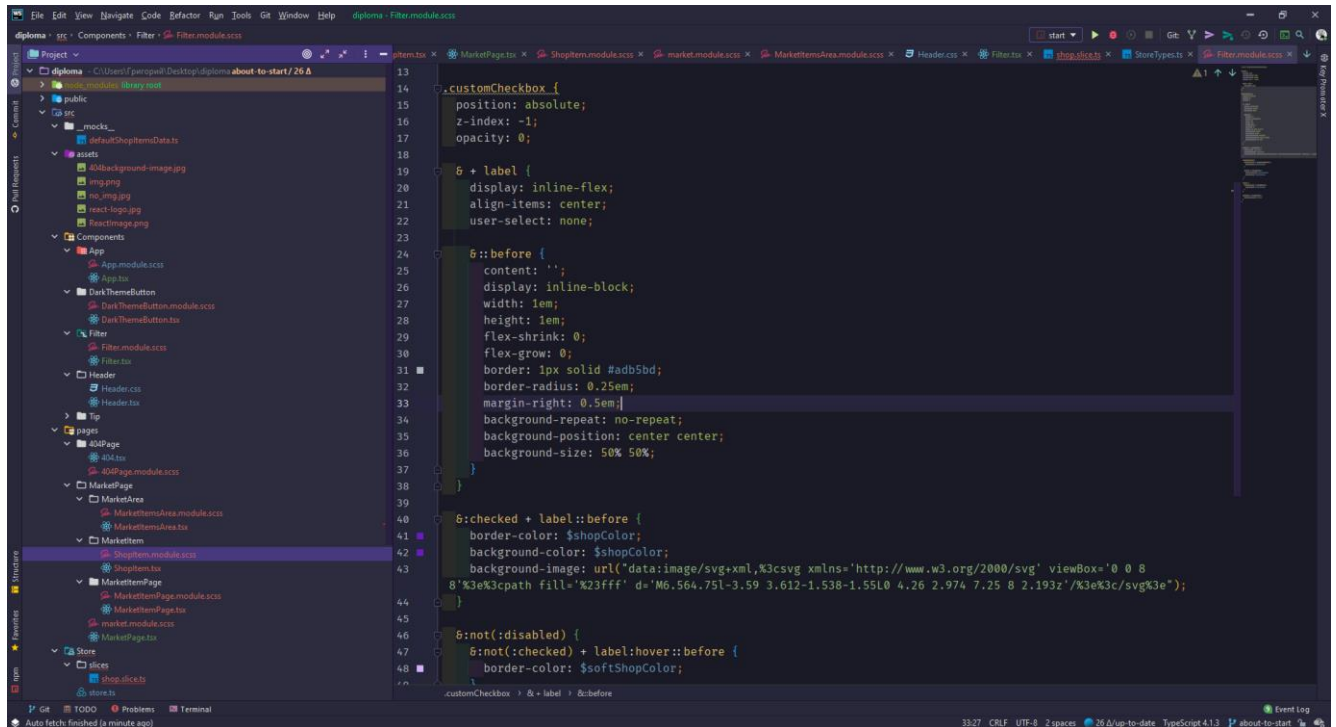


Рисунок 13 – Стилизация фильтра с помощью SCSS

Рисунок 13 отображает стили для создания кнопки фильтра. Данная кнопка имеет несколько состояний и для каждого применяются свои стили.

Скриншоты кода алгоритма фильтрации товаров:

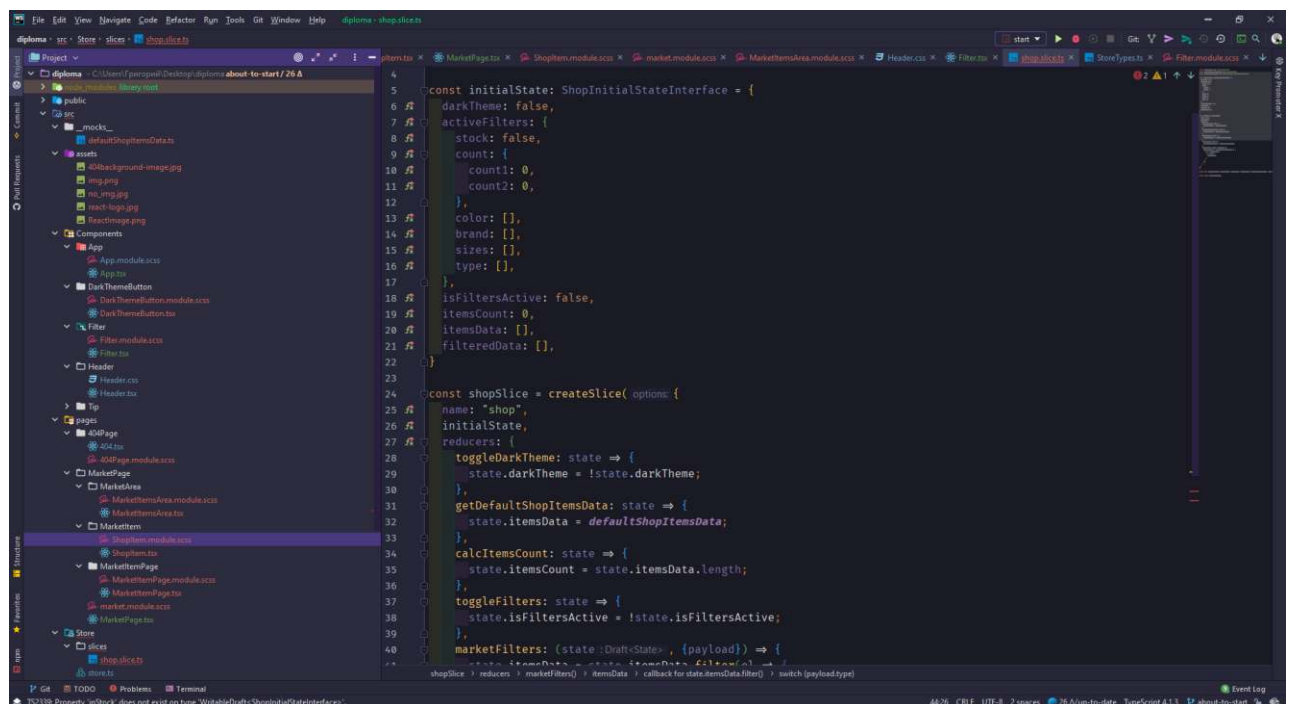


Рисунок 14 – Создание логики фильтрации товаров в менеджере состояний Redux

На рисунке 14 частично показана логика интернет-витрины: получение данных, начальное состояние бизнес-логики, различные функции. Вместо использования классического подхода к созданию глобального хранилища был применен модуль Redux, который называется `redux-toolkit`. С помощью него появляется возможность разделять одно глобальное хранилище данных. Таким образом можно изолировать различную логику приложения для того, чтобы избежать повторы названий переменных. [2]

После того, как интернет-витрина написана, можно выложить ее код на площадку GitHub. В случае, если разработкой занимается несколько программистов, на этой площадке есть возможность контролировать версии приложения и отмечать ошибки других пользователей.

2.3. Инструкция использования шаблона в проекте

Удобство использования GitHub заключается в подходе хранения данных. Скачивание проекта на свой компьютер занимает мало времени, что позволяет командам разработчиков быстрее взаимодействовать с кодом, а в случае обновления проекта достаточно ввести команду в терминале “`git clone *ссылка на проект*`”. Для дальнейшей установки модулей, используемых в разработке, например, React и Redux, нужно написать в терминале команду “`npm i`”. [15]

Код разработанного шаблона был размещен на площадке GitHub. Данный репозиторий открыт для просмотра другим пользователям.

Для использования шаблона нужно:

- 1) Перейти по ссылке github.com/ScoRe187/diploma.git;
- 2) Скачать код данного веб-приложения;
- 3) Вставить файлы в свое веб-приложение.

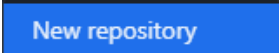
В будущем планируется добавить возможность интегрировать данную интернет-витрину с помощью пакетного менеджера NPM для простоты «вливания» в свое веб-приложения. При помощи NPM сторонние разработчики смогут использовать компоненты витрины также, как были использованы функции и модули из других библиотек.

2.4. Размещение кода интернет-витрины для SPA на площадку GitHub

На рисунках 15 – 17 можно увидеть каким образом размещается код на площадке GitHub с пояснениями.

Для начала регистрируемся и авторизуемся на сайте. Затем создаем новый репозиторий. Для создания своего репозитория пользуемся следующей инструкцией:

1) На рисунке 15 отображена иконка  , на которую необходимо нажать, для появление интерактивного окна.

2) Рисунок 16 показывает появившееся модальное окно, в котором имеется несколько кнопок. Вам нужно нажать на кнопку «New repository». 

3) На рисунке 17 видно, как можно настроить репозиторий под свои нужды.

4) Для загрузки данного репозитория к себе на компьютер можно воспользоваться терминалом и вписать команду «git clone». Чтобы проинициализировать Git в свой проект нужно вписать в терминал «git init -у», при этом предварительно установив Git на свой компьютер и авторизовавшись в программе.



Рисунок 15 – иконка «Плюсик» рядом с аватаром профиля

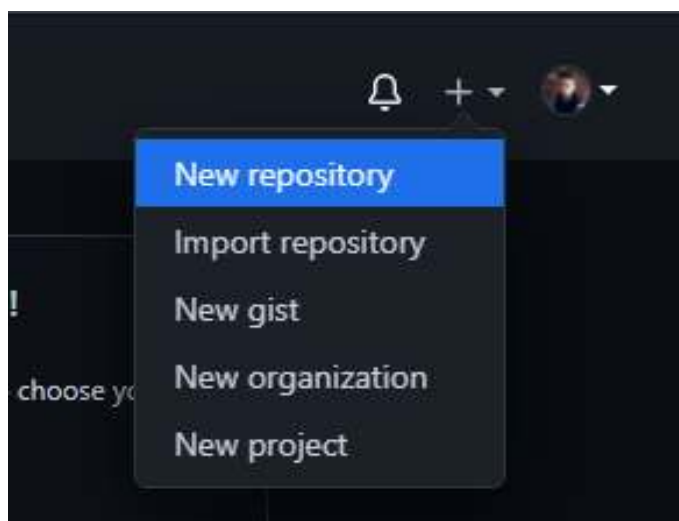


Рисунок 16 – Ссылка на создание нового репозитория

На рисунке 16 – Ссылка на создание нового репозитория показаны возможности взаимодействия с загрузкой на площадку GitHub своих фалов. Помимо создания нового репозитория, можно импортировать уже существующий, например, если был утерян предыдущий аккаунт. Можно также создать организацию и новый проект, в которые могут подключиться дополнительные пользователи для совместной разработки.

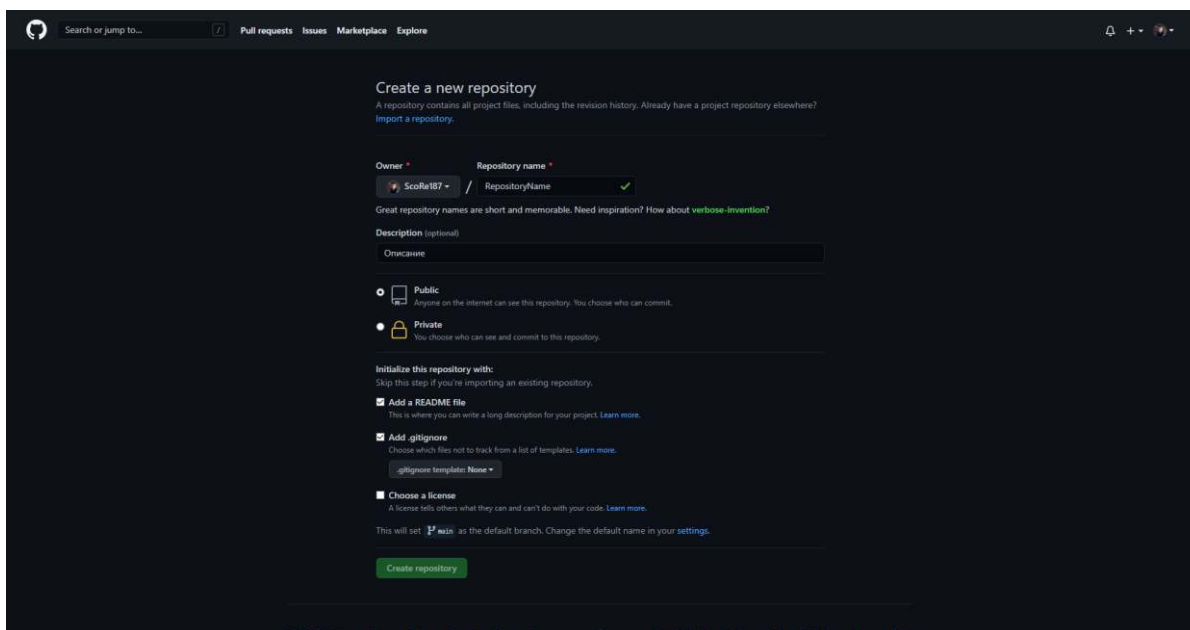


Рисунок 17 – Создание репозитория

На рисунке 17 – Создание репозитория показано как создавать новый репозиторий. Вы можете назвать репозиторий самостоятельно, также написать описание к репозиторию, задать степень защиты, а еще выбрать следует ли добавить в новый репозиторий файл, который нужно прочитать перед использованием кода и файл, с помощью которого система Git будет понимать какие файлы не следует загружать в репозиторий.

Установка данного шаблона не займет много времени. Даже один специалист сможет разобраться с кодом для его правильного внедрения в уже существующее веб-приложение компании.

ЗАКЛЮЧЕНИЕ

Разработка собственного веб-приложения в наше время требует больших затрат, учитывая возможность сэкономить, используя Node JS. Разработанный шаблон будет также помогать предприятиям с ограниченными финансами и сроком выпуска готового программного продукта.

Разработка выпускной квалификационной работы на тему: «Разработка шаблонов для реализации интернет-витрин на основе веб-приложения» соответствует профессиональным стандартам и требованиям к области профессиональной деятельности выпускников специальности «09.02.05 Прикладная информатика (по отраслям)». В данной работе при изучении объекта исследования применены практические знания и умения в создании веб-ресурсов.

В результате выполнения выпускной квалификационной работы был разработан шаблон для одностраничных веб-приложений, позволяющий экономить время и деньги предприятиям с ограниченным бюджетом или временем, желающим как можно быстрее реализовать собственный продукт.

В ходе разработки приложения были изучены: построение блоков на сайтах, язык программирования TypeScript, препроцессор SCSS для быстрого написания стилей, работа с библиотеками React и Redux.

Для достижения цели выпускной квалификационной работы были выполнены проведен анализ интернет-магазинов было замечено несколько ошибок в подходе к созданию дизайна. Некоторые моменты были исправлены в данном шаблоне.

Было проведено исследование и сравнительная характеристика существующих сред разработки и редакторов кода, а также библиотек по созданию веб-приложений. Лучшее решение было использовано для создания шаблона интернет-витрины.

Выполнен анализ предметной области с изучением существующих источников по созданию веб-приложений и сравнительный анализ библиотек для создания веб-приложений. Самым эффективным и простым в изучении, является React. Удобство использования JSX позволяет не тратить много времени на верстку и написание скриптов и их дальнейшее объединение из разных файлов. Однако, для

создания такого веб-приложения требуется немало места на компьютере, так как хранение и использование сторонних библиотек и компонентов требует немало места на диске. Данную проблему позволяет решить GitHub и пакетный менеджер NPM (англ. Node Package Manager). Вам не обязательно хранить все библиотеки на площадке, загрузка проекта на свой компьютер займет немного времени.

Задача по созданию шаблона интернет-витрины для одностраничных веб-приложений была решена полностью, однако, некоторые моменты можно доработать для более комфортного подключения шаблона в другие проекты. Единственным недостатком данного шаблона является отсутствие алгоритма фильтрации иных видов категорий товаров, так как данный шаблон изначально разрабатывался под интернет-магазины продающие вещи для гардероба. Данный недостаток планируется исправить в будущем, для достижения полного взаимодействия со всеми сферами продаж.

При создании шаблона для веб-приложения требуется написать алгоритм фильтрации товаров, есть вероятность столкнуться с проблемами реализации. Потребуется некоторое время на продумывание логики отображения отфильтрованных товаров. Возможно, что придется обратиться на форум с просьбой о помощи в решении возникающих проблем, таких как: написание стилей SCSS, получение данных для компонентов из менеджера состояний Redux и написание алгоритма фильтрации товаров.

Созданный шаблон также масштабируется в зависимости от размеров экрана устройства, что позволяет использовать его и в создании мобильных веб-приложений с помощью React Native.

Исследовав поставленные задачи выпускной квалификационной работы, можно сказать о том, что на данный момент количество одностраничных веб-приложений в сфере продаж крайне мало. Различия между веб-приложением и веб-сайтом, а также сильные стороны веб-приложений были описаны выше.

В выпускной квалификационной работе подробно описан алгоритм создания интернет-витрины для одностраничных веб-приложений и, дополнительно, представлены инструкции по установке данного шаблона в собственное веб-приложение и возможности загрузки собственного проекта на площадку GitHub.

Работа с данной тематикой по созданию шаблона для одностраничных веб-приложений предоставляет возможность дополнительно попрактиковаться в сфере создания SPA и получить опыт в области создания дизайна для веб-ресурсов.

Практическая значимость работы заключается в пользе для предприятий с ограниченным бюджетом. Создание шаблона позволит современным компаниям затрачивать меньше времени и денег на разработку их продукта для продаж в интернете.

Будущее стоит за веб-приложениями, так как они снижают нагрузку на интернет, что позволяет пользователям с большим комфортом пользоваться веб-ресурсом.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления (введен в действие Приказом Росстандарта от 24.10.2017 N 1494-ст.).
2. Методические указания по подготовке и оформлению выпускных квалификационных работ. Рябчикова Е.А. – М.: МКАГ, 2021. – 37 с.
3. React и Redux. Функциональная веб-разработка. Алекс Бэнкс, Ева Порселло – СПб.: Питер, 2018. – 336 с.
4. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд. Хайнеман, Джордж, Пояяис, Гэри, Сеяков, Стэнли.: Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. — 432 с.
5. Секреты CSS. Идеальные решения ежедневных задач. Веру Л. — СПб.: Питер, 2016. — 336 с.
6. Секреты JavaScript ниндзя, 2-е изд.: Пер. с англ. - СПб.: ООО "Альфа-Книга", 2017. - 544 с.
7. Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений. Борис Черный. — СПб.: Питер, 2021. — 352 с.
8. Эффективный TypeScript: 62 способа улучшить код. Вандеркам Дэн. — СПб.: Питер, 2020. — 288 с.
9. Изучаем TypeScript 3. Натан Розенталс. – М.: ДМК Пресс, 2019. – 624 с.
10. <https://bigital.ru/programma-dlja-veb-programmirovanija-html/> – сравнение программ для веб-разработки.
11. <https://dudom.ru/kompjutery/programmy-dlja-sozdanija-veb-prilozhenij/> – сравнение программ для создания веб-приложений.
12. <https://vc.ru/seo/108149-odnostranichnye-spa-i-mnogostranichnye-pwa-veb-prilozheniya> - что такое веб-приложение и в чем разница между SPA и MPA.
13. <https://habr.com/ru/post/476312/> – React или Angular или Vue – что выбрать?

14. <http://doc.knigi-x.ru/22metodichka/192161-1-federalnoe-agentstvo-svyazi-federalnoe-gosudarstvennoe-obrazovatelnoe-byudzhetnoe-uchrezhdenie-visshego-professional.php> – стилизация React компонентов с помощью SCSS.
15. <https://vk.com/idvlad98> – Влад Коноплёв создание репозитория.
16. <https://www.figma.com> – создание интерфейсов и прототипов.

ПРИЛОЖЕНИЯ

Приложение 1. Список используемых сокращений

Сокращения на английском:

- 1) DOM (англ. Document Object Model) – обработанный HTML документ, с которым может взаимодействовать браузер.
- 2) IDE (англ. Integrated Development Environment) – интегрированная среда разработки.
- 3) JS – JavaScript.
- 4) TS – TypeScript.
- 5) NPM (англ. Node Package Manager) – пакетный менеджер Node.js.
- 6) SPA – (англ. Single Page Application) – одностраничное приложение.

Сокращения на русском:

- 1) ИС – информационная система.
- 2) ЛК – личный кабинет.
- 3) ООП – объектно-ориентированное программирование.
- 4) ОС – операционная система.

Приложение 2. Программный код компонента «Страница интернет-витрины»

```

import React, {useEffect, useState} from "react";
import Filter from "../Components/Filter/Filter";
import styles from './market.module.scss';
import MarketItemsArea from "../MarketArea/MarketItemsArea";
import {useDispatch} from "react-redux";
import { getDefaultShopItemsData } from "../Store/slices/shop.slice";

const MarketPage = () => {
  const dispatch = useDispatch();
  dispatch(getDefaultShopItemsData)
  return (
    <div className={styles.MarketPage}>
      <span className={styles.MarketPage__PreviewLabel}> Шаблон интернет-витрины для вашего веб-
приложения </span>
      <div className={` ${styles.MarketPage__Market} ${styles.DarkTheme}`}>
        <Filter/>
        <MarketItemsArea />
      </div>
    </div>
  );
};

export default MarketPage;

```