

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

МОДЕЛИРОВАНИЕ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Федорущенко Руслана Александровича

Научный руководитель
зав. кафедрой техн. пр.,
к. ф.-м. н., доцент

И. А. Батраева

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

Саратов 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Основное содержание работы	4
1.1 Фреймворк ReactJS	4
1.2 Фреймворк VueJS	5
1.3 Язык программирования Typescript	6
1.4 CSS-библиотеки	6
1.5 Прочие вспомогательные технологии	7
1.6 Первый проект	7
1.7 Второй проект	9
1.8 Выводы	14
ЗАКЛЮЧЕНИЕ	16

ВВЕДЕНИЕ

Актуальность темы. В современном мире информационных технологий разработка интерфейсов пользовательских приложений, или frontend-разработка, играет ключевую роль в создании веб-приложений и сайтов. Эффективный и удобный интерфейс напрямую влияет на пользовательский опыт, что, в свою очередь, определяет успех или провал продукта на рынке. Важность frontend-разработки постоянно растет, а с ней и разнообразие инструментов и технологий, предоставляемых для создания динамичных и интерактивных веб-приложений.

Актуальность темы дипломной работы обусловлена стремительным развитием веб-технологий и постоянной потребностью в квалифицированных специалистах, способных эффективно использовать новейшие инструменты для создания качественных продуктов. Результаты данной работы помогут как текущим, так и будущим разработчикам выбрать наиболее подходящие технологии для реализации своих проектов.

Цель бакалаврской работы — осуществить обзор и сравнительный анализ современных и наиболее популярных средств разработки Frontend-приложений. В рамках работы будут рассмотрены основные инструменты, такие как HTML, CSS и JavaScript, а также современные фреймворки и библиотеки, включая React, Vue.js, Redux, Tailwind и так далее.

Поставленная цель определила **следующие задачи**:

1. выбрать наиболее подходящие технологии для создания приложений;
2. осуществить обзор основных особенностей выбранных технологий, выявить их недостатки и преимущества;
3. разработать два похожих по функционалу приложения на разных средствах разработки;
4. сравнить используемые в разработке библиотеки и фреймворки по общим критериям.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы — 94 страницы, из них 72 страниц — основное содержание, включая 29 рисунков, съёмный носитель в качестве приложения, список использованных источников информации — 20 наименований.

1 Основное содержание работы

1.1 Фреймворк ReactJS

React — это JavaScript библиотека, разработанная компанией Facebook, предназначенная для создания пользовательских интерфейсов. Ее основная цель — облегчить разработку масштабируемых и быстрых веб-приложений, особенно тех, где данные часто изменяются. React работает по принципу создания компонентов, которые представляют собой независимые строительные блоки интерфейса. Каждый компонент может содержать в себе свою логику и состояние, что делает код более модульным и управляемым.

React основан на концепции компонентов. Компоненты — это строительные блоки веб-интерфейса, которые позволяют разбивать пользовательский интерфейс на небольшие, независимые и переиспользуемые части.

Вместо того, чтобы манипулировать реальным DOM напрямую, React использует виртуальное представление DOM. При обновлении состояния или данных компонента React сначала изменяет виртуальное дерево DOM, а затем сравнивает его с реальным DOM, чтобы выявить минимальное количество мутаций, которые необходимо применить.

JSX — это расширение языка JavaScript, которое позволяет писать HTML-подобный синтаксис прямо в коде JavaScript.

В React данные передаются по иерархии компонентов в одном направлении - сверху вниз. Это означает, что данные могут быть переданы только от родительских компонентов к дочерним, и изменения в состоянии или данных родительского компонента автоматически приводят к обновлению дочерних компонентов.

В React, компоненты могут принимать параметры, которые называются «props». Props — это данные, которые передаются от родительского компонента дочернему.

React Hooks — это нововведение в библиотеке React, которое было представлено в версии 16.8. Они предоставляют способ добавления состояния и других возможностей React в функциональные компоненты.

React Router — это библиотека для маршрутизации в веб-приложениях, созданных с использованием React. Она предоставляет навигационные компоненты и API для управления маршрутами и отображения различных компонентов в зависимости от URL.

React Redux — это библиотека, которая обеспечивает управление состоянием приложения в React при помощи паттерна управления состоянием Flux. Redux является одним из наиболее популярных решений для управления состоянием в React-приложениях благодаря своей простоте и эффективности.

1.2 Фреймворк VueJS

Vue.js — это JavaScript-фреймворк, предназначенный для создания пользовательских интерфейсов. Его основная цель — упростить разработку масштабируемых и динамических веб-приложений. Vue.js использует компонентный подход, где каждый компонент представляет собой независимый модуль, включающий в себя логику, шаблон и стили. Это позволяет создавать более модульный и легко управляемый код.

Vue.js использует декларативный подход для построения интерфейса пользователя. Это означает, что разработчики описывают, какие данные должны быть отображены, а Vue заботится о том, чтобы DOM соответствовал этим данным. Это упрощает процесс разработки, так как не требуется писать процедурный код для манипуляций с DOM.

Подобно React, Vue основан на идее компонентов как строительных блоков приложения.

Одной из ключевых особенностей Vue является его реактивная система данных. Vue использует прозрачные зависимости на основе getter и setter для каждого компонента, обеспечивая автоматическое обновление интерфейса при изменении данных. Эта система позволяет разработчикам легко отслеживать изменения в состоянии приложения без дополнительных усилий по синхронизации или обновлению DOM.

Vue предоставляет мощные директивы, такие как v-if, v-for, v-model, которые используются прямо в шаблонах HTML для создания динамических и интерактивных приложений.

Как и React, Vue использует Virtual DOM, чтобы оптимизировать обновления DOM.

Vue использует HTML-базирующую синтаксическую структуру для шаблонов, что делает код более читаемым и легким для восприятия.

Реактивность во Vue.js является концепцией, которая позволяет разработчикам строить интерактивные и динамические приложения.

Vue Router является официальной библиотекой маршрутизации для фреймворка Vue.js, обеспечивающей интеграцию между приложениями и браузером.

VueX — это библиотека для управления состоянием приложений на Vue.js, разработанная для обеспечения централизованного и надежного способа управления данными через всё приложение.

1.3 Язык программирования Typescript

TypeScript — это язык программирования, разработанный и поддерживаемый компанией Microsoft. Он является строго типизированным надмножеством JavaScript, что означает, что любой действующий JavaScript-код также является действующим TypeScript-кодом.

TypeScript предоставляет разработчикам мощные инструменты для создания более безопасного и масштабируемого кода. Одной из главных особенностей TypeScript является статическая типизация. Эта функция позволяет разработчикам определять типы данных для переменных, функций и объектов на этапе разработки, что способствует раннему обнаружению ошибок и повышению надежности кода.

TypeScript также вводит объектно-ориентированные концепции, такие как классы и интерфейсы, которые не полностью поддерживаются в стандартном JavaScript.

Работа с дженериками — еще одна ключевая возможность TypeScript. Дженерики позволяют создавать компоненты, которые могут оперировать с несколькими типами данных.

Интеграция с существующим JavaScript-кодом также является значительным преимуществом TypeScript. Он позволяет разработчикам постепенно переводить большие базы JavaScript-кода на TypeScript, обеспечивая при этом совместимость с существующими библиотеками и фреймворками.

1.4 CSS-библиотеки

Sass (Syntactically Awesome Stylesheets) — это препроцессор, который добавляет мощные возможности в обычный CSS, такие как переменные, вложенные правила, миксины и функции. Sass имеет два синтаксиса: старый, с отступами, известный как .sass и новый, похожий на обычный CSS, известный как .scss

Tailwind CSS — это утилитарный фреймворк для CSS, предназначенный для быстрой разработки пользовательских интерфейсов. В отличие от традиционных CSS-фреймворков, которые предоставляют целый набор стилизованных компонентов, Tailwind предоставляет набор низкоуровневых утилитарных классов, которые позволяют быстро создавать уникальные дизайны без необходимости писать много собственного CSS.

Основное преимущество использования Tailwind заключается в его модульности и масштабируемости. С помощью Tailwind разработчики могут быстро адаптировать дизайн страницы, изменяя классы в HTML, не трогая CSS файлы. Это обеспечивает большую гибкость и удобство в процессе разработки, так как можно видеть изменения в реальном времени.

1.5 Прочие вспомогательные технологии

Axios — это JavaScript библиотека для выполнения HTTP-запросов, которая широко используется во frontend и backend разработке. Одним из основных преимуществ Axios перед стандартным Fetch API, включенным в браузеры, является его способность автоматически преобразовывать JSON-данные. В то время как Fetch требует явного преобразования ответа из JSON, Axios делает это автоматически, существенно упрощая работу с данными.

Lodash — это популярная JavaScript библиотека, предоставляющая удобные утилиты для работы с массивами, числами, объектами и строками.

MockAPI — это инструмент, который предоставляет разработчикам возможность создавать макетные API для имитации серверных ответов без необходимости развертывания реального backend.

MOKKY.dev — это менее известный аналог MockAPI, созданный совсем недавно и полностью покрывающий функционал MockAPI, добавляя свои особенности, например, объединение данных по общему признаку.

ESLint и Prettier — это два популярных инструмента, используемые в разработке программного обеспечения, особенно при работе с JavaScript и другими языками, транслируемыми в JavaScript, такими как TypeScript.

1.6 Первый проект

Первый реализованный проект представляет собой интернет-магазин кроссовок, в котором пользователь может просматривать товары, переключаться

между страницами товаров, сортировать и искать товары, добавлять товары в избранное и корзину, удалять товары из избранного и корзины.

Всего приложение содержит три страницы:

- «Главная»: содержит в себе декоративный блок, рекламный блок в виде слайдера, блок с карточками кроссовок, сортировкой и поиском, а также пагинацию;
- «Избранное»: содержит карточки товаров, добавленных в избранное. При обновлении страницы или при перезаходе в браузер, карточки товаров остаются;
- «Заказы»: содержит информацию о всех заказах, оформленных пользователем. Заказы также сохраняются при обновлении страницы.

Выбранные технологии: Vue.js, Vue Router, Tailwind, axios, lodash, Mokky.dev, Auto Animate.

Главная страница отображается пользователю сразу при заходе на сайт.

Корзина реализована в виде Drawer, который открывается при нажатии на кнопку «Корзина» в навигации приложения (см. рисунок 1).

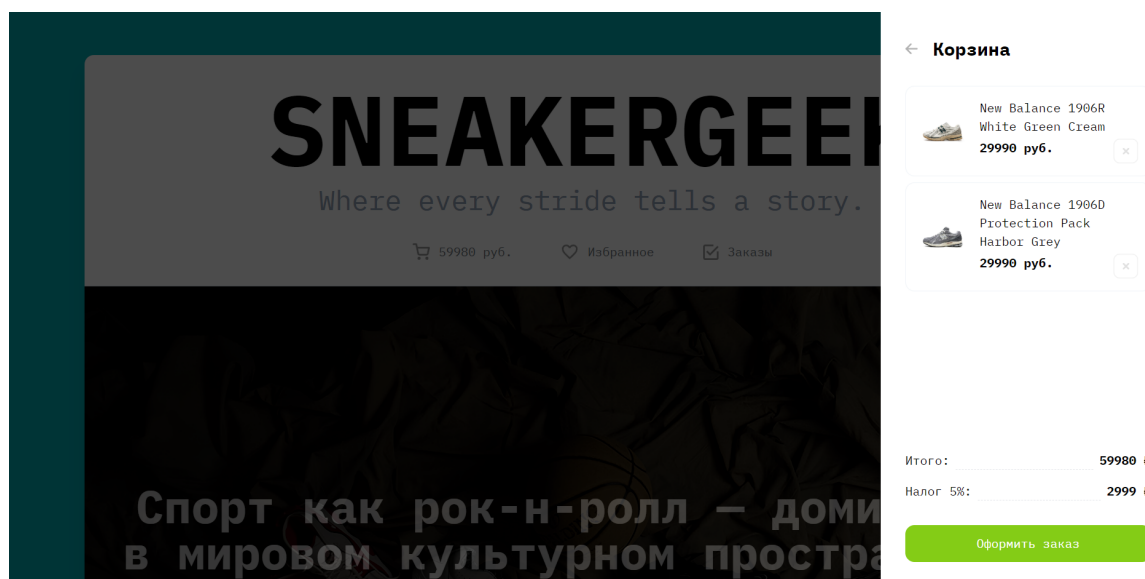


Рисунок 1 – Корзина

При первом рендере компонента Favorites при помощи директивы onMounted отправляется get-запрос на /favorites, ответ которого помещается в реактивный массив favorites. Далее при помощи v-for осуществляется проход по элементам массива, данные которых при помощи props отправляются в компонент карточки и выводятся на странице. Внешний вид страницы «Избранное» представлен на рисунке 2.

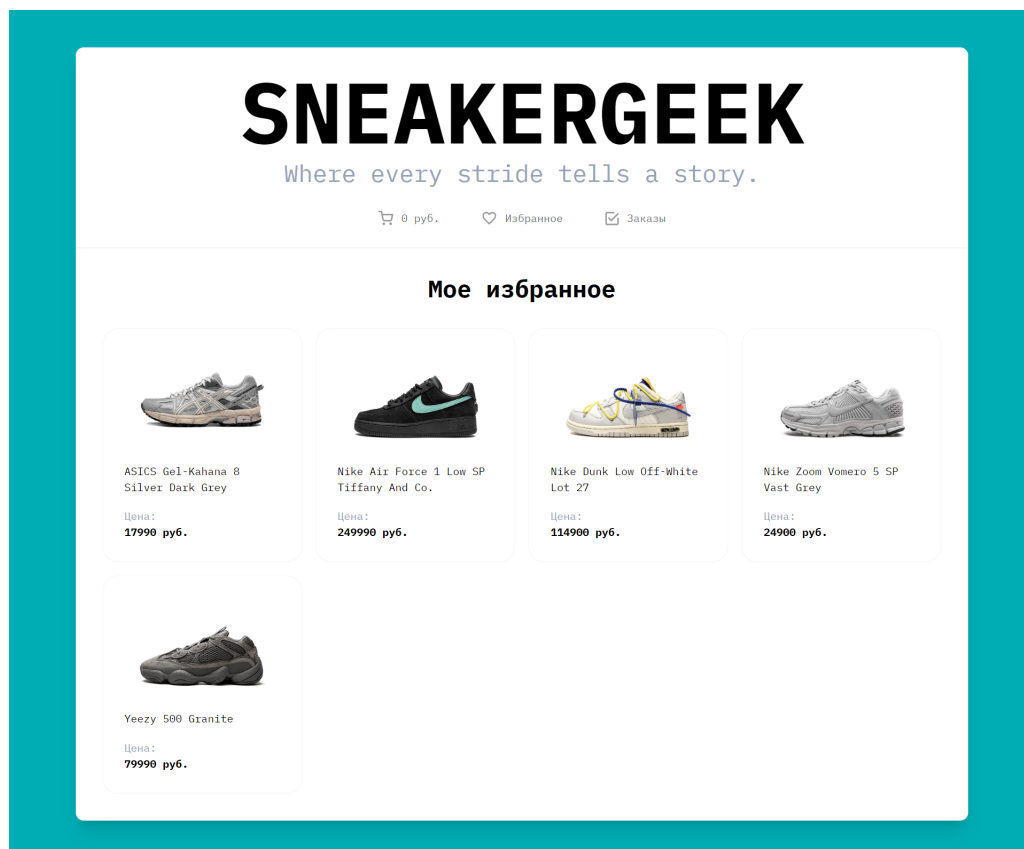


Рисунок 2 – Избранное

На странице «Заказы» логика работы с отображением данных практически повторяет логику на странице «Избранное». При первом рендере Orders отправляется get-запрос на /orders, затем данные записываются в реактивный массив orders, обход по которому и вывод заказов на экран осуществляется при помощи v-for (см. изображение 3).

1.7 Второй проект

Второй реализованный проект в рамках текущей дипломной работы представляет из себя сервис по доставке еды, в котором пользователь так же, как и в первом проекте может просматривать товары, переключаться по страницам с товарами, осуществлять сортировку, фильтрацию и поиск товаров, изменять модификаторы товара, просматривать подробную информацию о каждом товаре, добавлять товары в корзину, удалять товары из корзины, очищать корзину, просматривать акционные страницы и добавлять акционные товары.

Всего приложение содержит пять страниц:

- «Главная»: содержит в себе промо-блок в виде слайдера, блок с вакансией, блок с картой, на которой отмечены адреса заведений, блок с карточками

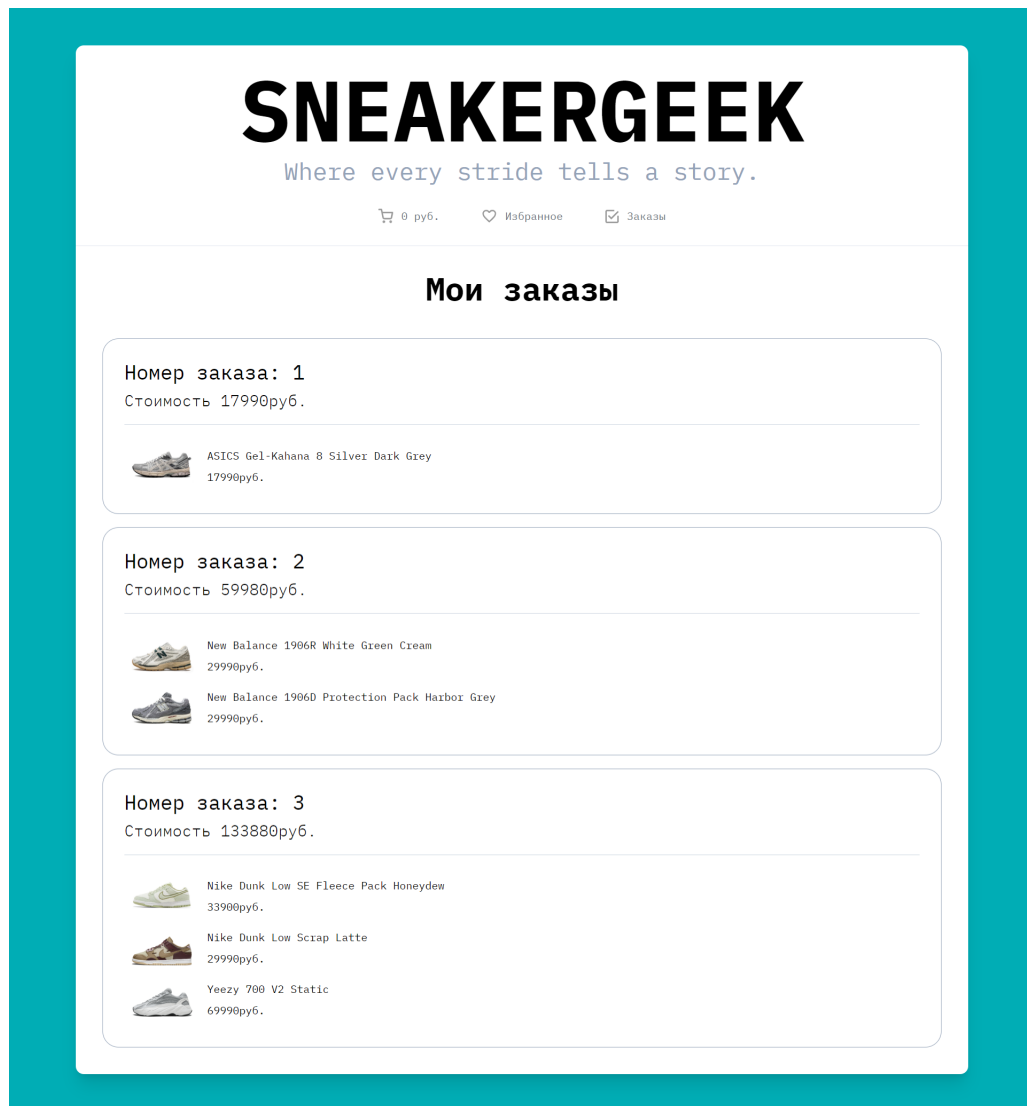


Рисунок 3 – Заказы

товаров, сортировкой и фильтрацией, а также пагинацию;

- «Страница товара»: содержит изображение и полное описание товара и его особенностей;
- «Промо-страница»: содержит список акционных предметов и их описание;
- «Корзина»: содержит список предметов, добавленных пользователем в корзину. При обновлении страницы или при перезаходе в браузер, карточки товаров остаются;
- «Ничего не найдено»: используется для обработки ошибок с запросом страниц.

Используемые технологии: React, Typescript, SASS/SCSS, Redux Toolkit, React Router, axios, lodash, mockAPI.

Главная страница поделена на блоки:

Промо-блок представляет из себя заголовок, подзаголовок и такой же

слайдер, как и в первом проекте, но с более широким функционалом и добавленным таймером пролистывания в правый нижний угол.

Карьерный блок предлагает пользователю устроиться на работу в команду ресторана, оставив свою почту для дальнейшей связи.

Блок с картой отображает адреса заведений в городе Саратов.

Блок с товарами поделен на три части, как и в первом проекте: фильтрация и сортировка, карточки товаров и пагинация.

Общий вид главной страницы представлен на рисунке 4.

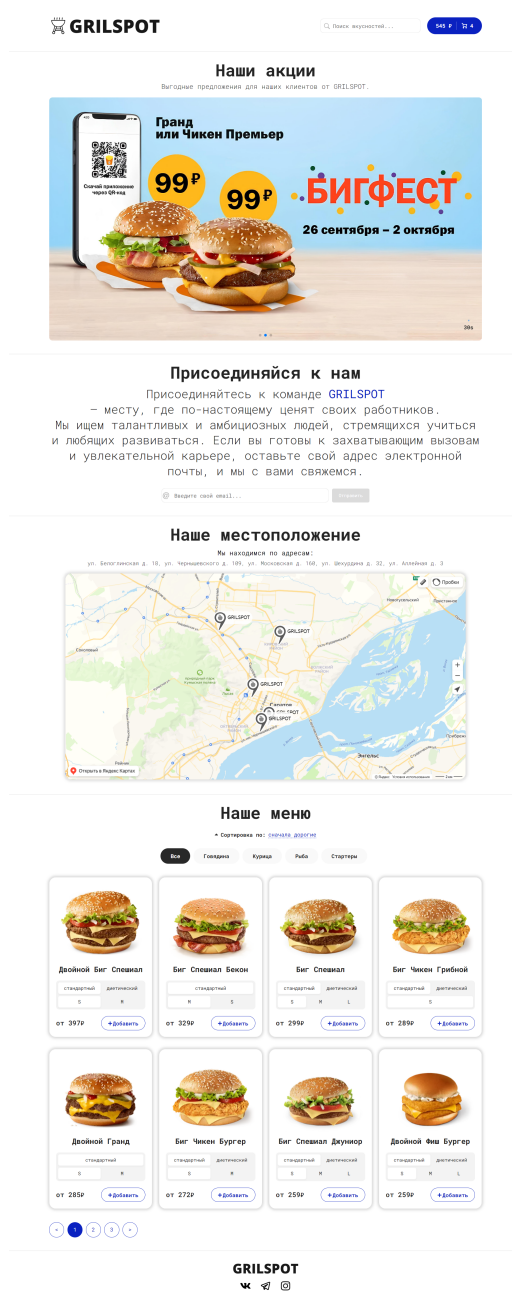


Рисунок 4 – Главная страница

В этом проекте корзина реализована в виде отдельной страницы на сайте,

которая содержит уменьшенные карточки товаров, добавленных в корзину, с их названием, типом, размером и тремя кнопками: прибавить один товар, удалить один товар и полностью удалить этот товар из корзины. Так же в корзине есть кнопки «Очистить корзину», «Вернуться назад» и «Оплатить заказ». Внизу корзины отображается общее количество предметов и их стоимость.

Общий вид корзины представлен на рисунке 5.

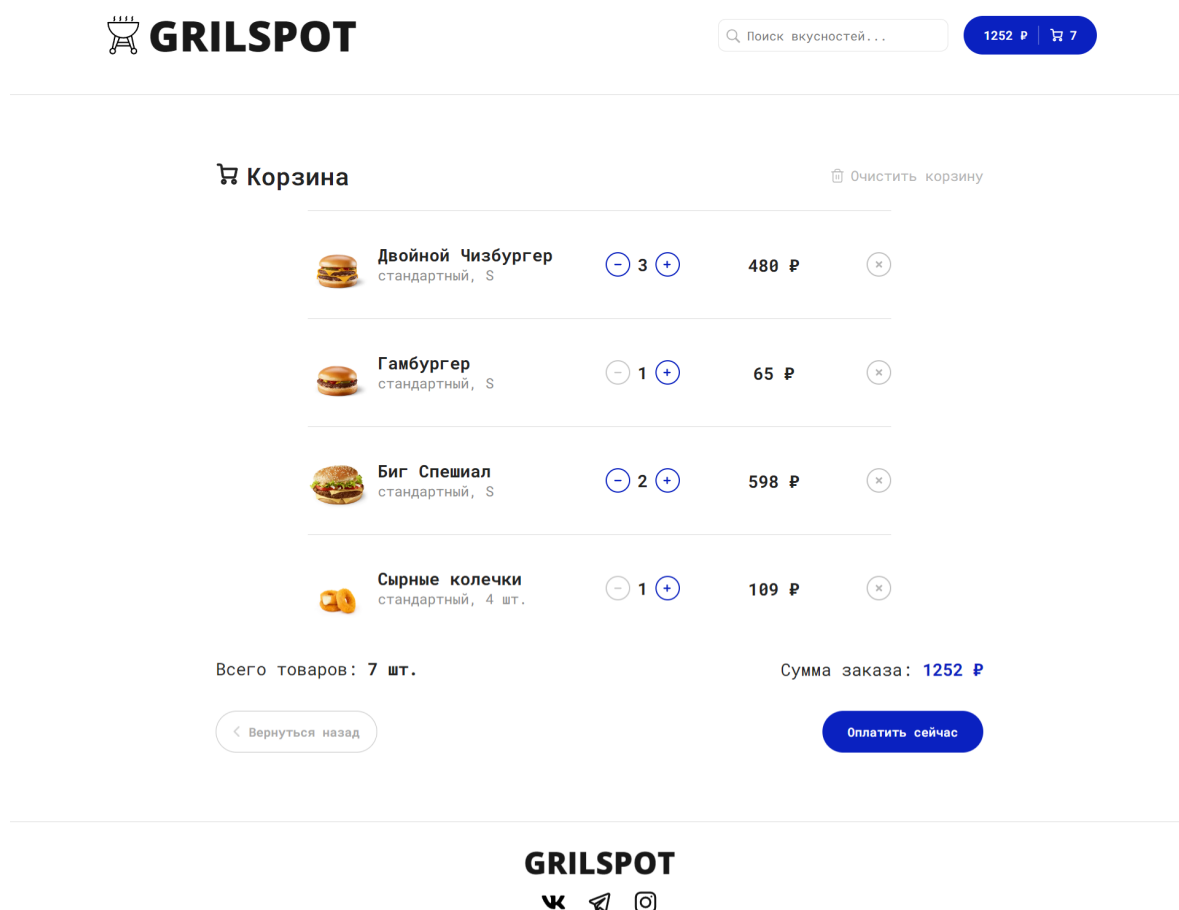


Рисунок 5 – Корзина

Страница товара содержит крупное изображение, название и подробную информацию о товаре: описание, категорию, доступные для выбора типы и размеры, а также стоимость. Внешний вид страницы товара представлен на рисунке 6.

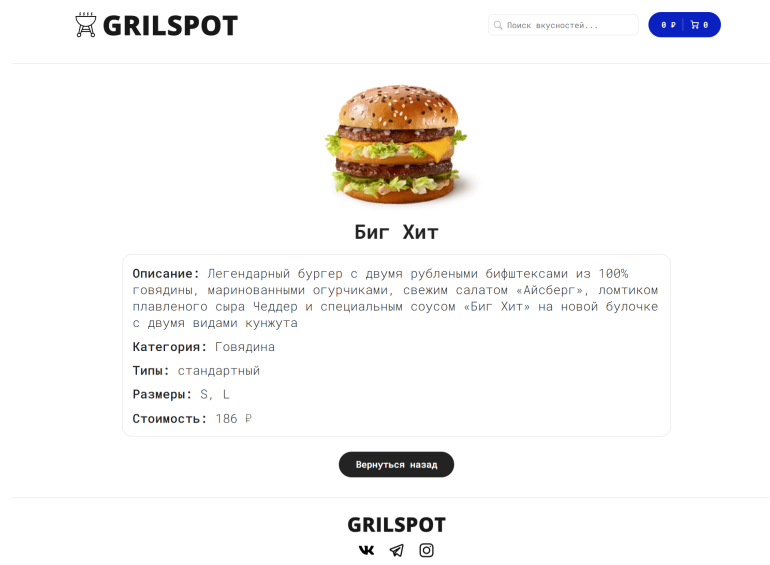


Рисунок 6 – Страница товара

Промо-страница содержит заголовок, кнопку «Вернуться назад» и две карточки товаров, участвующих в акции, у которых нельзя изменить тип и размер, а цена изменена на акционную. Также карточки имеют кнопки «Добавить в корзину» и «Подробнее».

Внешний вид промо-страницы представлен на рисунке 7.

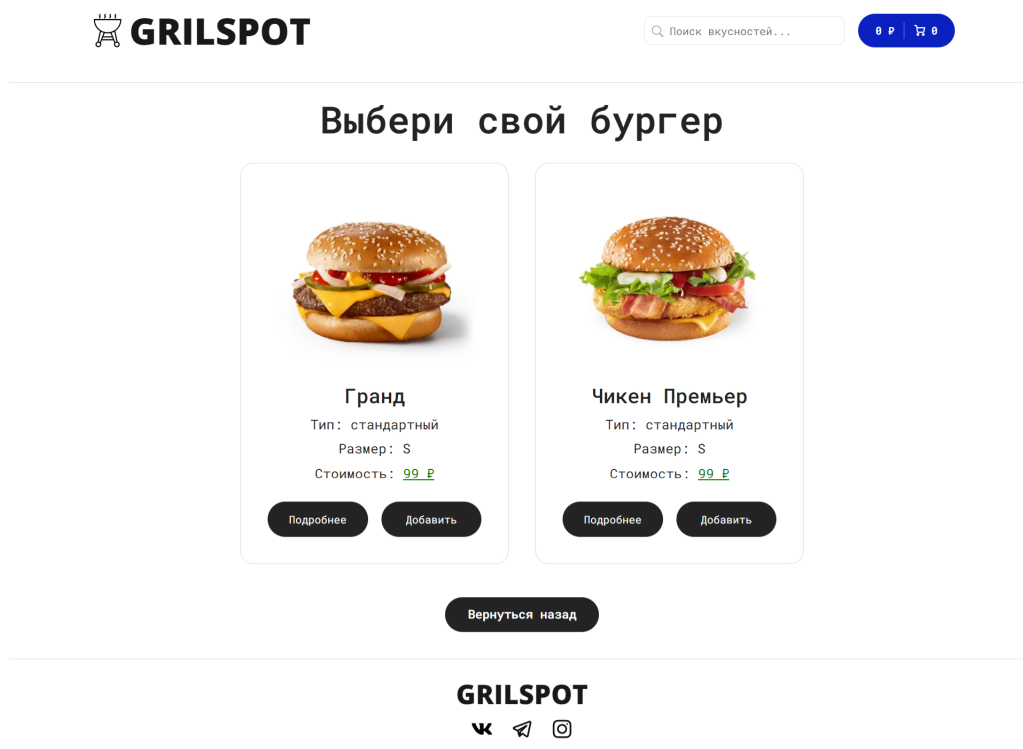


Рисунок 7 – Промо-страница

Страница «Ничего не найдено» появляется в том случае, если пользова-

тель попытался перейти по адресу, которого не существует.

Внешний вид страницы представлен на рисунке 8

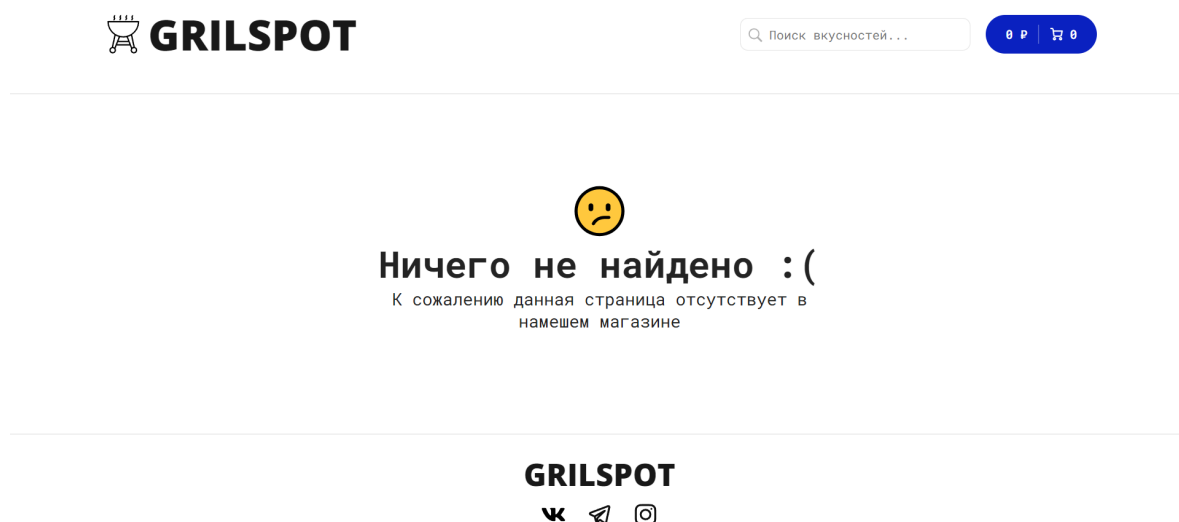


Рисунок 8 – Ничего не найдено

Для оптимизации приложения в некоторых его местах используются такие технологии, как `React.lazy`, `React.memo` и `React.useCallback`.

1.8 Выводы

Выбор между `React` и `Vue` часто зависит от конкретных требований проекта, предпочтений команды разработчиков и существующих технологий в компании. `React` предлагает более зрелую экосистему и широкие возможности для интеграции с другими библиотеками, тогда как `Vue` выделяется простотой, интеграцией и быстрым обучением.

В рамках данной дипломной работы впервые была осуществлена работа с фреймворком `Vue` и по данному опыту можно сказать, что он обладает низким порогом вхождения и множеством интересных функциональностей, которых, например, не хватает в `React`, например `provide` и `inject`.

При сравнении `Tailwind` и `SCSS` можно отметить, что оба инструмента имеют свои уникальные особенности и области применения. `Tailwind` предлагает утилитарный подход к стилизации, который ускоряет разработку и уменьшает количество кастомного `CSS`. Это делает его отличным выбором для быстрого создания адаптивных интерфейсов и прототипов, особенно в сочетании с современными JavaScript-фреймворками.

С другой стороны, SCSS предоставляет более гибкие и мощные возможности для написания сложных и структурированных стилей. Он подходит для проектов, требующих высококастомизированных решений и строгого контроля над организацией кода стилей.

В конечном итоге, Tailwind подойдет для тех, кто ценит скорость разработки и минимизацию объема кастомного CSS, в то время как SCSS будет предпочтителен для проектов, требующих высокой гибкости, структурности кода и написания сложных стилей.

Redux Toolkit можно использовать практически на любом JS-фреймворке и даже на том же Vue, но VueX использовать где-то, кроме самого Vue, нельзя. Из этого вытекает и другой плюс Redux Toolkit: так как эта библиотека используется и на других фреймворках, то информации в интернете по ней гораздо больше, чем по VueX.

Так как React Router и Vue Router библиотеки разрабатывались под конкретный фреймворк, а Vue Router представляет из себя встроенную библиотеку во Vue.js, поэтому выбор между React Router и Vue Router следует делать на основе используемого фреймворка. Если разработка происходит на React, то React Router отлично подойдет для выполнения задач, связанных с маршрутизацией. Если приложение создается на Vue.js, то Vue Router так же справится с поставленными ему задачами.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы был произведен обзор, описаны основные возможности, выявлены достоинства и недостатки современных и наиболее популярных библиотек, фреймворков и сервисов, использующихся для разных целей при frontend-разработке, таких как структурирование приложения, стилизация проекта, хранение состояний, отправки запросов, создания API, типизация и проверка кода на качество.

В результате были разработаны два приложения на разных технологиях с похожим функционалом и наполнением, включающим в себя навигацию, контентные блоки, отображение товаров, поиск, сортировку и фильтрацию товаров, корзину, пагинацию, маршрутизацию между страницами сайта и хранение данных при помощи сервисов создания API.

Благодаря реализованным приложениям произведен сравнительный анализ библиотек и фреймворков по набору критериев, присущих той или иной паре технологий. Осуществлено сравнение React и Vue, Tailwind и SCSS, Redux Toolkit и Vuex, React Router и Vue Router, а так же в конце каждого из них был подведен итог при каких ситуациях стоит сделать выбор в пользу той или иной технологии.