

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра математической теории упругости и биомеханики

Автоматизация управления заказами в сфере оптовой торговли

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы
направление 09.03.03 — Прикладная информатика

механико-математического факультета

Галаева Дени Руслановича

Научный руководитель
к.ю.н., доцент

подпись, дата

Р.В. Амелин

Зав. кафедрой
д.ф.-м.н., профессор

подпись, дата

Л.Ю. Коссович

Саратов 2024

Введение. В наше быстро развивающееся время виртуальный мир становится неотъемлемой частью нашей повседневной жизни. Одним из ключевых элементов этого мира являются веб-приложения – программные приложения, доступные через интернет и функционирующие на удаленных серверах. Их популярность и востребованность неизменно растут, проникая в различные сферы деятельности, включая коммерцию.

Веб-приложения стали неотъемлемой частью современного бизнеса. Они предоставляют компаниям возможность расширения своего онлайн присутствия, упрощают взаимодействие с клиентами, повышают уровень обслуживания, улучшают процессы управления. В результате, использование веб-приложений становится стратегически важным для бизнеса способствуя его развитию и укреплению конкурентных позиций на рынке.

Целью данной работы заключается в разработке веб-приложения для автоматизации управления заказами в сфере оптовой торговли. Глобально задача заключается в создании интернет-магазина строительных материалов с административной панелью, что позволит управлять содержимым сайта. Веб-приложение должно также отвечать современным стандартам безопасности и использовать передовые технологии разработки.

В соответствии с поставленной целью в работе определены следующие задачи:

1. Провести анализ и определить требования по функционалу;
2. Разработать проект схемы данных;
3. Разработать серверную часть;
4. Разработать клиентскую часть;
5. Обеспечить интеграцию между серверной и клиентской частью;
6. Провести тестирование приложения.

Первый раздел включает анализ предметной области, чтобы понять

ключевые аспекты и потребности бизнеса в сфере оптовой торговли строительными материалами.

Были определены общие требования компании к итоговому веб-приложению:

- Система должна автоматически принимать и обрабатывать заказы от клиентов.
- Реализация системы уведомлений о статусе заказа для клиентов.
- Регистрация для создания личного кабинета.
- Полноценная админ-панель.
- Безопасность приложения, надежное шифрование.

Особое внимание уделялось безопасности данных и защите от несанкционированного доступа.

Также в данном разделе были рассмотрены аналогичные веб-приложения конкурентов. В открытом доступе были найдены несколько наиболее популярных интернет-магазинов по продаже строительных материалов. Анализ данных веб-приложений позволил выявить интересные и эффективные решения, которые могут быть применены при разработке интернет-магазина.

Во втором разделе описывается этап проектирования.

Проектирование веб-приложения началось с разработки ER-диаграммы базы данных, изображенной на рисунке включающей основные сущности и их взаимосвязи, такие как пользователи, товары, заказы, категории товаров. В работе они часто используются при проектировании баз данных, для определения структуры и основных элементов хранения данных в системе. Это позволило четко структурировать данные и обеспечить их целостность.



Рисунок 1 — ER-диаграмма базы данных

Выбор стека технологий PERN был обоснован необходимостью обеспечения высокой производительности и надежности приложения. PostgreSQL была выбрана в качестве базы данных за ее мощные возможности работы с данными и масштабируемость. Express и Node.js обеспечивали гибкость и производительность на серверной стороне, а React – динамичный и отзывчивый интерфейс на стороне клиента.

Термин "технологический стек" подразумевает совокупность языков программирования, фреймворков, библиотек, баз данных, инструментов и практик, применяемых для разработки определенной части веб-приложения. Выбор каждого элемента стека имеет важное значение, так как от него зависят производительность, безопасность, масштабируемость, простота поддержки и другие ключевые характеристики проекта. Поэтому подбор технологий требует тщательного анализа текущих и перспективных потребностей приложения

Архитектура веб-приложения была построена на принципе разделения клиентской и серверной частей. Это позволило достичь высокой производительности и гибкости в разработке и последующем обслуживании приложения. Важным аспектом проектирования было обеспечение безопасности, что включало использование современных методов шифрования данных и аутентификации пользователей.

В качестве архитектуры клиентской части была взята методология Feature Sliced Design (FSD). Это современный и относительно новый архитектурный подход к организации frontend-кода, который фокусируется на разделении функциональности приложения на независимые функциональные слои (feature slices).

Методология не привязана к конкретному языку программирования, UI-фреймворку или менеджеру состояния — подойдет любой.

Проект на FSD состоит из слоев (layers), каждый слой состоит из слайсов

(slices) и каждый слайс состоит из сегментов (segments). Пример на рисунке 2.

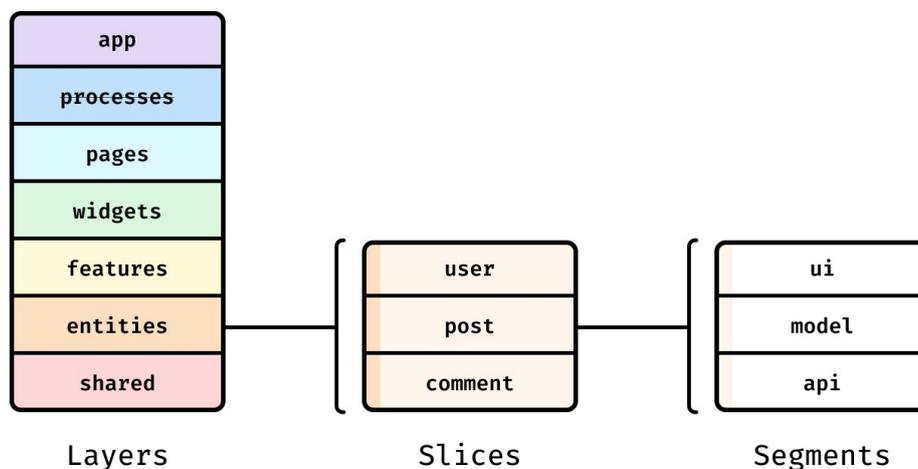


Рисунок 2 — FSD архитектура

В качестве серверной архитектуры за основу взят стандартный стиль REST-API.

REST (от англ. REpresentational State Transfer — «передача репрезентативного состояния» или «передача „самоописываемого“ состояния») - архитектурный стиль взаимодействия компонентов распределённого приложения в сети. Отличительной особенностью сервисов REST является то, что они позволяют наилучшим образом использовать протокол HTTP.

Для организации модулей кода внутри серверного приложения используется Model-View-Controller (MVC) методология.

Это схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо. На рисунке 3 изображена схема данной архитектуры.

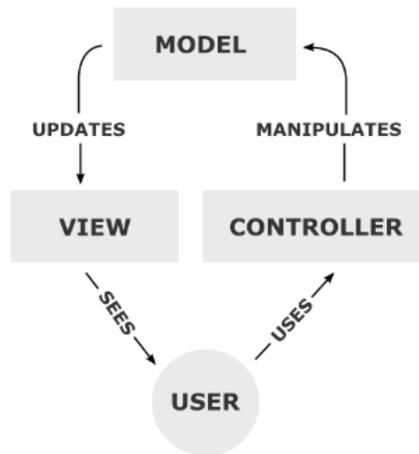


Рисунок 3 — Архитектура Model-View-Controller

Для улучшения видимости веб-приложения в поисковых системах была проведена оптимизация, включающая добавление метатегов, создание карты сайта и оптимизацию скорости загрузки страниц. Эти меры способствовали повышению рейтинга приложения в результатах поиска и увеличению потока клиентов. В качестве инструмента использовался React-snap.

React-snap позволяет после сборки осуществить запуск базового браузерного движка Chromium и передать ему получившееся приложение. Затем пройти по всем его страницам, используя список найденных ссылок так, как сделал бы это поисковой робот. При этом html-копии отрисованных средствами js страниц будут сохранены в выбранный каталог с сохранением иерархии путей, из которых они были получены относительно корня проекта.

Третий раздел содержит описание реализации веб-приложения.

Реализация веб-приложения началась с разработки серверной части. Были созданы основные маршруты для обработки запросов клиентов, реализован механизм аутентификации и авторизации пользователей, обеспечивающий высокую безопасность данных. На серверной стороне использовались Express и Node.js, что обеспечило высокую производительность и гибкость в разработке.

Клиентская часть веб-приложения была реализована с использованием React и TypeScript. Это позволило создать динамичный и отзывчивый

интерфейс, который обеспечивает удобство использования для клиентов. Основные компоненты включали формы для регистрации и авторизации пользователей, страницы каталога товаров, корзины и оформления заказа. На рисунке 4 изображена главная страница интернет-магазина.

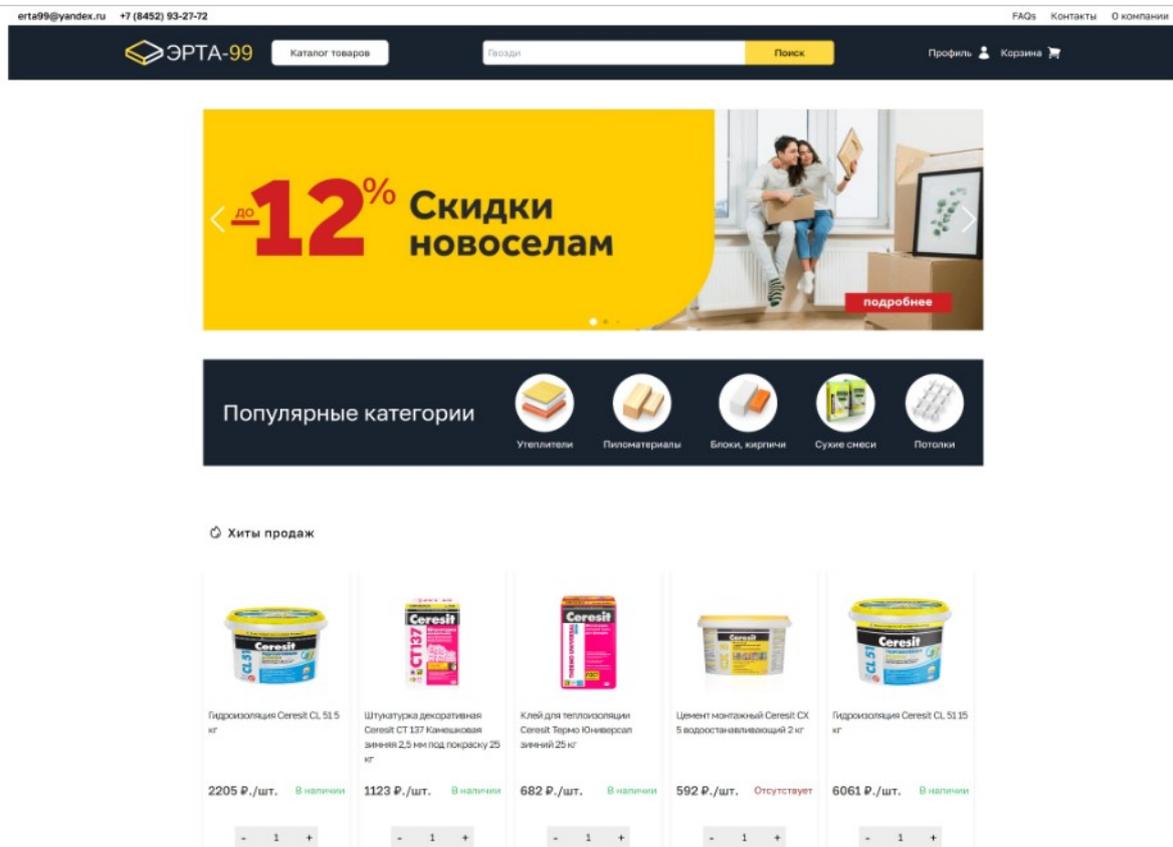


Рисунок 4 — Главная страница интернет-магазина

Одним из ключевых элементов веб-приложения стала административная панель, которая позволяла управлять товарами, заказами и пользователями. Панель включала функционал для добавления, редактирования и удаления товаров, обработки заказов и управления пользователями. Это обеспечивало удобное и эффективное управление контентом и бизнес-процессами. Итоговый интерфейс панели администратора изображен на рисунках 5 и 6.

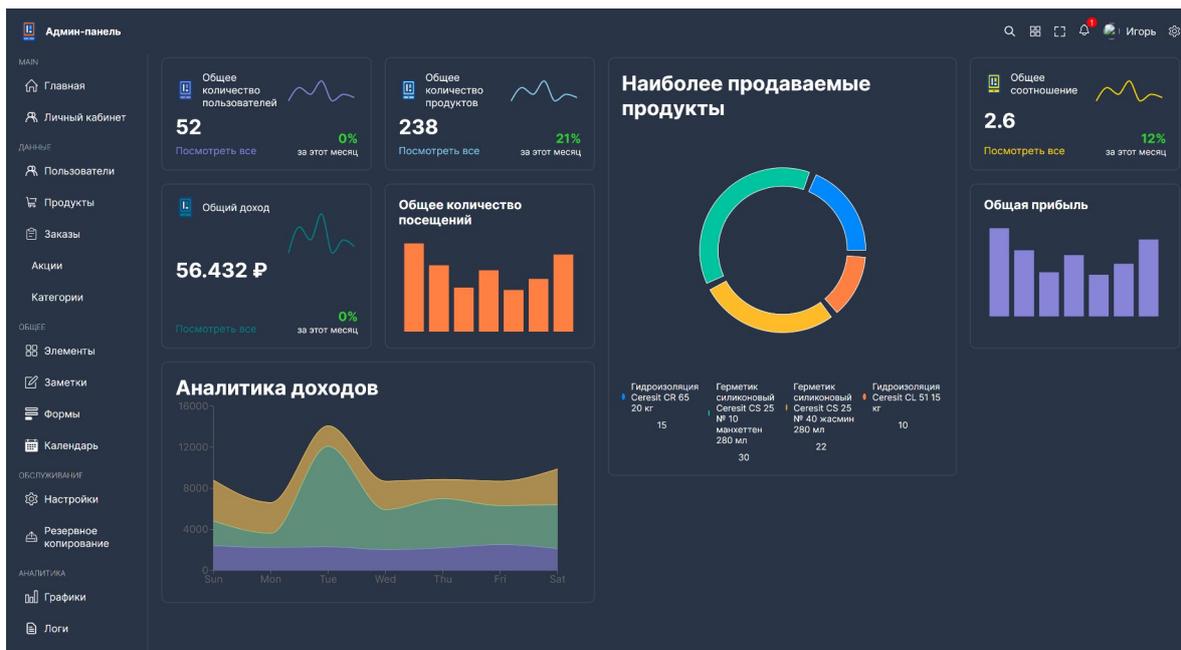


Рисунок 5 — Интерфейс панели администратора

The 'Товары' section includes a search bar and an 'EXPORT' button. The table below lists the items:

ID	Наименование	Описание	Цена	Производитель	Дата создания	В наличии	Action
1	Цемент Цемрос М500 ЦЕМ II А...		566Р	Цемрос	20.05.2024	✓	🗑️
2	Цемент Цемрос М500 ЦЕМ I 42...		580Р	Цемрос	20.05.2024	✓	🗑️
3	Цемент Цемрос М500 ЦЕМ I 42...		311Р	Цемрос	20.05.2024	✓	🗑️
4	Цемент Akkermann М500 ЦЕМ II...		620Р	Akkermann	20.05.2024	✓	🗑️
5	Цемент Akkermann М500 ЦЕМ II...		345Р	Akkermann	20.05.2024	✗	🗑️
6	Цемент Axton М400 5 кг		166Р	Axton	20.05.2024	✓	🗑️
7	Цемент Akkermann М600 ЦЕМ I ...		370Р	Akkermann	20.05.2024	✗	🗑️
8	Цемент Cemix М500 ПЦБ 1-500...		850Р	Cemix	20.05.2024	✓	🗑️
9	Микроцемент инъекционный Ми...		2700Р	Микроджет	20.05.2024	✗	🗑️
10	Краска Dulux Prof Bindo 7, ВВ (...)		6974Р	Dulux	20.05.2024	✓	🗑️

Рисунок 6 — Управление товарами в панели администратора

Важной частью реализации стало тестирование веб-приложения. Были проведены функциональные и нагрузочные тесты, которые подтвердили высокую производительность и надежность системы. Особое внимание уделялось безопасности данных и защите от несанкционированного доступа.

Заключение. В результате работы был разработан интернет-магазин строительных материалов и панель для администрирования сайта.

Разработка велась в соответствии с техническим заданием заказчика. Приложение отвечает требованиям заказчика и стандартам безопасности, благодаря использованию современных стандартов разработки и надежному алгоритму шифрования на серверной части.

На сегодняшний день имеется огромное число технологий для разработки веб-приложений. Чтобы разработать действительно качественный и современный онлайн-ресурс, необходимо использовать современные инструменты и наиболее проверенные методы разработки.

Данная работа была выполнена на стеке технологий PERN. Это стек JavaScript, который используется для более простого и быстрого развертывания полнофункциональных веб-приложений. Стек PERN состоит из 4 технологий: PostgreSQL, Express, React, Node.js. Он разработан, чтобы сделать процесс разработки более плавным и легким, а также имеет большой плюс, так как весь стек полностью реализуется на JavaScript.

Все цели и задачи, описанные во введении, выполнены в полном объеме. Проведенная работа позволила изучить процесс создания современных одностраничных веб-приложений, а также повысить свои навыки решения задач в области веб-разработки. Разработанное веб-приложение обладает перспективами дальнейшего масштабирования и совершенствования.