

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ
ТОВАРНЫМИ ДАННЫМИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ
REACT И NODE.JS**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 5 курса 551 группы

направления 09.03.04 Программная инженерия

факультета компьютерных наук и информационных технологий

Фроловой Екатерины Владимировны

Научный руководитель:

к.ф.-м.н., доцент

_____ И. А. Батраева

подпись, дата

Зав. кафедрой:

к. ф.-м.н.

_____ С. В. Миронов

подпись, дата

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы. С каждым годом технологии продолжают развиваться, и всё больше людей и организаций начинают использовать веб-приложения для упрощения своей работы. Это особенно актуально для торговых компаний, где точная информация о товарах и возможность быстро находить нужные данные играют важную роль.

Покупатели часто сталкиваются с трудностями в поиске информации о товарах, таких как характеристики, сроки годности, условия хранения и декларации о соответствии. Клиентам компаний, закупающим товары для перепродажи (например, магазинам), необходимо знать размеры изделий — высоту, глубину и ширину — чтобы понимать, сколько места они займут на полках. Также важно иметь доступ к качественным изображениям товаров для использования в каталогах и рекламе. Такие данные нередко трудно найти или они представлены в неудобном формате, что затрудняет выбор подходящего продукта.

Это создает потребность в разработке веб-приложений, которые помогут эффективно организовать хранение и поиск информации о товарах. Пользователи должны иметь возможность быстро находить описания продуктов и получать доступ к важным документам, таким как декларации соответствия. Такие приложения не только ускоряют поиск информации, но и оптимизируют внутренние рабочие процессы компании. К примеру, если будет реализована функция загрузки деклараций по указанным кодам, сотрудник сможет оперативно получить архив с нужными документами на товары, приобретённые клиентом за определённый период, что значительно упростит и ускорит сбор информации.

Сегодня активно развиваются технологии, гарантирующие безопасность данных и удобство их обработки. Создание веб-приложения на основе современных технологий, таких как Node.js, React и PostgreSQL, позволит разработать надёжную и удобную систему как для пользователей, так и для администраторов.

Таким образом, разработка такого веб-приложения становится важной и актуальной задачей. Оно упростит работу с информацией и улучшит взаимодействие с клиентами, сделав данные о товарах более доступными и понятными. Внедрение таких технологий поможет компаниям оперативно реагировать на запросы клиентов и обеспечивать высокое качество обслуживания. Это, в свою очередь, поможет компаниям повысить свою конкурентоспособность на рынке, привлекая больше клиентов и укрепляя позицию в данной сфере.

Цель бакалаврской работы – разработка веб-приложения, которое упростит поиск и управление информацией о продуктах компании.

Поставленная цель определила **следующие задачи**:

1. Анализ потребностей пользователей: Изучить требования целевой аудитории и определить, какую информацию и в каком формате пользователи хотят получать;
2. Проектирование архитектуры приложения: Разработать архитектуру, охватывающую как клиентскую, так и серверную часть, для правильной работы всех функций;
3. Разработка клиентской части: Создать интерфейс приложения, обеспечив удобную навигацию и функциональность для поиска информации;
4. Разработка серверной части: Организовать обработку запросов на сервере, настроить управление доступом пользователей с помощью аутентификации и наладить работу с базой данных для хранения и извлечения информации о продуктах;
5. Настройка серверной среды: Разместить приложение на VPS, приобрести доменное имя, настроить SSL-сертификаты и сделать сайт доступным через поисковые системы;
6. Тестирование и оптимизация приложения: Протестировать все части приложения, выявить и устранить ошибки.

Методологические основы разработки веб-приложений представлены в работах Н. Р. Полуэктова [1] и А. М. Заяц [2].

Теоретическая и практическая значимость бакалаврской работы. Теоретическая часть данной работы посвящена анализу современных подходов к разработке веб-приложений. В рамках исследования рассматриваются методологии проектирования, архитектуры приложений и организации баз данных.

Практическая значимость работы проявляется в разработке функционального веб-приложения, которое будет способствовать более эффективному управлению информацией о товарах.

Структура и объём работы. Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 2 приложений. Общий объём работы – 54 страницы, из них 47 страниц – основное содержание, включая 19 рисунков и цифровой носитель в качестве приложения, список использованных источников информации – 22 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Проектирование приложения» посвящён процессу разработки веб-приложения.

В разделе 1.1 рассматриваются основные аспекты и технологии, связанные с созданием и поддержанием веб-сайтов и веб-приложений. В работе внимание уделяется двум ключевым направлениям: фронтенд (клиентская сторона) и бэкенд (серверная сторона) [1 - 3].

Фронтенд-разработка сосредоточена на создании пользовательского интерфейса. К основным технологиям, используемым во фронтенде, относятся HTML, который отвечает за структуру веб-страниц, CSS, обеспечивающий оформление и стилизацию, и JavaScript, добавляющий интерактивность. Также рассматриваются популярные фреймворки и библиотеки, среди которых React, позволяющий создавать динамичные интерфейсы [4].

Бэкенд-разработка фокусируется на серверной логике, управлении данными и реализации бизнес-логики. Здесь используются различные языки программирования, среди которых JavaScript в среде Node.js, а также популярные фреймворки, например, для Node.js это может быть Express.js, который упрощает разработку серверной части [1 - 3].

Кроме того, в работе рассматриваются базы данных, которые могут быть реляционными (SQL) или нереляционными (NoSQL).

Также раздел охватывает этапы размещения веб-сайта, включая покупку домена, выбор хостинга, настройку сервера и индексацию сайта.

В подразделе 1.2 рассматриваются технологии React и Node.js, а также их интеграция для создания современного веб-приложения. Объясняется, как React используется для фронтенда, обеспечивая динамичный пользовательский интерфейс, а Node.js — для бэкенда, предоставляя серверную логику и обработку запросов [4, 5].

В подразделе 1.3 описана архитектура веб-приложения, включающая распределение компонентов на фронтенд и бэкенд, а также способы взаимодействия между ними.

В подразделе 1.4 описываются этапы разработки проекта. Проект был разделён на несколько ключевых этапов, начиная с создания функционала для управления пользователями и заканчивая разработкой главной страницы, где отображается весь ассортимент продукции.

В подразделе 1.5 рассматривается структура веб-страниц приложения. Описывается интерфейс для авторизации пользователей, работы с ассортиментом товаров, управления данными и пользователями.

В подразделе 1.6 рассматривается разработка модели базы данных для приложения с использованием PostgreSQL. Разработанная схема базы данных содержит таблицы для учета пользователей, товаров, брендов, категорий и других связанных данных, обеспечивающих эффективное управление информацией в системе.

Раздел «Проектирование приложения» охватывает основные этапы разработки веб-приложения, включая выбор технологий, архитектуру системы, структуру интерфейса и модель базы данных.

Второй раздел «Разработка приложения» посвящен реализации веб-приложения.

В подразделе 2.1 описаны цели клиентской части, которые были реализованы. Они включают обеспечение безопасности и контроля доступа к данным через механизмы аутентификации и авторизации, создание удобного и интуитивно понятного интерфейса для работы с системой, а также поддержку адаптивного дизайна для корректного отображения на различных устройствах.

В подразделе 2.1.1 описана структура клиентской части проекта и основные технологии, использованные в разработке, среди которых можно выделить React для создания интерфейса, React Router для маршрутизации, Axios для работы с API, а также JWT для аутентификации пользователей.

В подразделе 2.1.2 описана реализация взаимодействия с серверной частью с использованием библиотеки Axios для выполнения запросов и обработки ошибок.

В подразделе 2.2 рассматривается серверная часть веб-приложения, разработанная с использованием фреймворка Express.js для обработки запросов, управления бизнес-логикой и взаимодействия с базой данных PostgreSQL. Ключевые компоненты включают аутентификацию пользователей с помощью JWT, обработку HTTP-запросов через API, работу с реляционной базой данных и интеграцию с внешними сервисами для выполнения таких задач, как отправка уведомлений по электронной почте.

В подразделе 2.2.1 описана структура серверной части проекта и основные технологии, использованные в разработке, среди которых можно выделить Node.js для выполнения серверного кода, Express.js для обработки HTTP-запросов и маршрутизации, Sequelize для работы с реляционной базой

данных PostgreSQL, а также библиотеки jsonwebtoken для аутентификации пользователей и Bcrypt для хеширования паролей.

В подразделе 2.2.2 описана реализация взаимодействия с клиентской частью. Для организации взаимодействия между серверной и клиентской частями используется фреймворк Express, который позволяет обрабатывать HTTP-запросы и управлять маршрутизацией.

В подразделе 2.3 описана организация серверной среды для развертывания веб-приложения с использованием виртуального частного сервера (VPS). Приведены технические характеристики выбранного тарифа и обоснование его выбора.

В подразделе 2.3.1 представлены рекомендации, на основании которых было выбрано доменное имя для веб-приложения, а также описан процесс его регистрации.

В подразделе 2.3.2 описаны шаги по настройке VPS-сервера для веб-приложения. Рассматривается процесс подключения к серверу через защищенный протокол SSH, который обеспечивает безопасную аутентификацию и передачу данных между клиентом и сервером. Описывается процесс создание нового пользователя с использованием SSH-ключей, что значительно повышает уровень безопасности по сравнению с использованием паролей. Далее описывается процесс установки и настройки веб-сервера Nginx, который отвечает за обработку HTTP-запросов и эффективное управление трафиком. Также рассматривается настройка сетевой безопасности с помощью фаервола UFW, который позволяет контролировать доступность серверных портов и защищает сервер от несанкционированных соединений. Кроме того, в разделе описывается установка Node.js для обработки серверной логики веб-приложения и взаимодействия с клиентами. Рассматривается интеграция с реляционной базой данных PostgreSQL, что позволяет надежно хранить и обрабатывать данные приложения. Завершается подраздел описанием установки SSL-сертификата, который обеспечивает

шифрование данных и подтверждает подлинность веб-сайта, тем самым повышая уровень доверия со стороны пользователей.

В подразделе 2.3.3 описан процесс индексации сайта и использование инструментов для веб-мастеров. Рассматриваются файлы robots.txt и sitemap.xml, которые регулируют доступ поисковых систем и помогают улучшить индексирование сайта.

В подразделе 2.4 описан процесс размещения клиентской части приложения на VPS. Рассматриваются два основных этапа: сборка проекта и загрузка файлов на сервер. Для сборки клиентской части приложения использовался инструмент react-scripts, который создает оптимизированную версию приложения. Для серверной части приложения, был настроен файл подключения к базе данных PostgreSQL. После завершения настройки и сборки всех частей приложения, для загрузки файлов на сервер использовался FTP-клиент FileZilla, что позволило успешно разместить файлы на сервере и сделать приложение доступным для пользователей в онлайн-режиме.

В подразделе 2.5 рассматриваются основные этапы тестирования веб-приложения, включая функциональное, пользовательское и производительное тестирование. Используя инструменты как Postman, Google Chrome и К6, была проведена проверка работоспособности функций, удобства интерфейса и устойчивости к нагрузкам, что позволило повысить качество и безопасность сайта для пользователей.

В разделе «Разработка приложения» были подробно описаны ключевые аспекты реализации веб-приложения, включая как клиентскую, так и серверную части. Обсуждаются цели и технологии, используемые в разработке, процессы настройки серверной среды и развертывания на VPS, а также шаги по индексации и тестированию приложения. В результате выполнения всех этапов было разработано веб-приложение с интуитивно понятным интерфейсом, доступное для пользователей в онлайн-режиме.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы было разработано веб-приложение, которое упростило поиск и управление информацией о продуктах компании. В процессе работы были выполнены следующие задачи:

1. Анализ потребностей пользователей: Осуществлен сбор информации о требованиях целевой аудитории, что позволило точно определить, какую информацию и в каком формате пользователи хотят получать.
2. Проектирование архитектуры приложения: Разработана структура приложения, которая обеспечивает эффективную работу как клиентской, так и серверной частей.
3. Разработка клиентской части: Создан интерфейс на React с удобной навигацией и функционалом для поиска информации. Были разработаны страницы «Ассортимент товаров», «Конструктор», «Диск» и «Управление пользователями».
4. Разработка серверной части: Организована обработка запросов с использованием Node.js, внедрена система аутентификации пользователей через JWT и настроена работа с базой данных для хранения и извлечения информации о продуктах.
5. Настройка серверной среды: Приложение развернуто на VPS, настроены SSL-сертификаты и обеспечена доступность сайта через поисковые системы.
6. Тестирование и оптимизация: Проведено тестирование, которое позволило выявить и устранить ошибки, а также оптимизировать производительность приложения.

В процессе разработки были применены современные технологии и инструменты, что значительно упростило процесс создания и тестирования приложения. Использование React, React Router, Axios, Node.js, Express.js и других технологий позволило оптимизировать функциональность и улучшить производительность приложения. Выполнение данной работы дало возможность приобрести практические навыки в области веб-разработки.

Основные источники информации:

1. Полуэктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2022. — 204 с.
2. Заяц А. М. Проектирование и разработка web приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие / А. М. Заяц, Н. П. Васильев. — СанктПетербург : Лань, 2020. — 120 с.
3. Основы фронтенд-разработки: фреймворки, DOM-дерево, взаимодействие с бэкендом. [Электронный ресурс]. — URL: <https://habr.com/ru/companies/ruvds/articles/722464/> (Дата обращения 10.04.2025). - Загл. с экр. - Яз. рус.
4. Знакомство с React. [Электронный ресурс]. — URL: <https://lenarin.github.io/tutorial/tutorial.html> (Дата обращения 24.04.2025). - Загл. с экр. - Яз.рус.
5. Введение в Node JS. — URL: <https://metanit.com/web/nodejs/1.1.php> (Дата обращения 26.05.2024). - Загл. с экр. - Яз. рус.