

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ  
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»  
(ГАПОУ ОКЭИ)

ОТЧЁТ

ОКЭИ 09.02.07. 9024. 05 П  
(код документа)

По профилю специальности 09.02.07 Информационные системы и программирование  
квалификация Разработчик веб- и мультимедийных приложений

ПМ.09 Проектирование, разработка и оптимизация веб-приложений

Производственная

(вид практики)

ГАЗ «Областной аптечный склад»

(место прохождения практики)

Количество листов 31

Дата готовности 17.04.2024

Руководитель Гиниятулина Н.В.

Разработал Галимьянова Э.Р.

Согласовано Варенникова Ю.В. Заместитель директора по ИТ

(ФИО руководителя от организации, должность, подпись)

Соответствие отчета с заданием на практику \_\_\_\_\_

Защищен \_\_\_\_\_ с оценкой \_\_\_\_\_

Оренбург 2024

# Содержание

Введение.....	3
1 Анализ предприятия.....	5
1.1 Характеристика предприятия .....	5
1.2 Анализ предметной области .....	7
2 Техническое задание .....	8
2.1 Требования к серверной части.....	8
2.2 Требование к разработке .....	9
2.3 Тестирование и документация.....	10
3 Разработка проектной документации.....	12
3.1 Разработка концепции веб-приложения .....	12
3.2 Разработка эскизного проекта веб-приложения .....	13
3.3 Разработка моделей пользовательского интерфейса .....	16
3.4 Обеспечение защиты информации.....	21
4 Разработка веб-приложения .....	22
4.1 Реализация веб-приложения .....	22
5 Тестирование веб-приложения .....	24
5.1 Тестирование кода веб-приложения .....	24
5.2 Аудит безопасности веб-приложения.....	25
6 Размещение и техническое сопровождение веб-приложения .....	27
6.1 Размещение веб-приложения.....	27
6.2 Техническое сопровождение .....	27
Заключение .....	29
Список используемых источников.....	30

					ОКЭИ 09.02.07. 9024. 05 П			
Изм.	Лист	№ докум.	Подпись	Дата				
Разаб.		Галимьянова Э.Р.			Отчёт по производственной практике		Лит.	Лист
Провер.		Гиниятулина Н.В.						
Реценз.							2	31
Н.Контр.							Отдел информационных технологий – 4661	
Утверд.								

## Введение

Необходимость создания информационных систем (ИС) может возникать как из необходимости автоматизации или модернизации уже существующих процессов, так и из необходимости полной реорганизации деятельности предприятия. Практически любая область деятельности (торговля, производство, медицина, обслуживание и т. д.) создает свою информационную систему.

В современном мире создание информационных систем становится все более важным. Они не только автоматизируют процессы, но и являются ключевым инструментом для управления и анализа данных, что особенно важно для различных сфер деятельности, от медицины до обслуживания и производства. Веб-приложения дополнительно повышают эффективность работы и обслуживания. Благодаря им, процессы обработки данных и взаимодействия с клиентами становятся более удобными, быстрыми и прозрачными.

Веб-приложение – это программное обеспечение, которое работает на удаленном сервере и доступно через Интернет посредством веб-браузера. Оно позволяет выполнять разнообразные задачи, включая работу с данными, обработку информации, взаимодействие с другими пользователями и другие функции. Веб-приложения могут иметь различные формы, от простых социальных сетей до сложных систем электронной коммерции, онлайн банкинга, интернет-форумов и т. д. Они позволяют пользователям получать информацию, взаимодействовать с контентом, проводить операции и транзакции в реальном времени. Это динамичные системы, которые обычно используют клиент-серверную архитектуру, где браузер клиента обращается к серверу для получения данных и выполнения действий.

Основные преимущества веб-приложений включают доступность с любого устройства с доступом к интернету, обновление программного обеспечения на удаленном сервере без необходимости обновления клиентской части, а также возможность централизованного управления и хранения данных.

Веб-приложения обычно разрабатываются с использованием различных технологий, таких как HTML, CSS и JavaScript для клиентской части, и языков программирования, таких как PHP, Python, Java или Ruby, для серверной части. Они также могут использовать базы данных для хранения информации и взаимодействия с другими сервисами.

Создание интернет-магазина с использованием React и Node.js остается актуальным и востребованным в современной веб-разработке. React и Node.js популярны благодаря своей отзывчивости, быстродействию, компонентной архитектуре и возможности создания одностраничного приложения (SPA). Оба фреймворка известны своей масштабируемостью, что важно для магазинов, сталкивающихся с ростом трафика и объемом данных. Node.js обеспечивает масштабирование серверной части, а React - клиентской.

В целом, интернет-магазины обеспечивают удобство и эффективность покупок, что делает их все более популярными среди потребителей. Неоспоримыми преимуществами интернет-магазинов являются привлечение новых клиентов, увеличение продаж и расширение бизнеса.

Во время прохождения производственной практики на предприятии ГАУЗ «ОАС», находящегося по адресу город Оренбург, улица Березка 24, необходимо разработать веб-приложение Интернет-магазина для продажи биологически активных добавок. Интернет-магазин продажи БАДов предоставляет организации множество преимуществ, помогая привлекать клиентов, показывать им полную информацию о продуктах и предлагать удобные условия покупки. Это способствует увеличению продаж и развитию бизнеса.

Цели, поставленные для выполнения во время производственной практики: реализация веб-приложения для Интернет-магазина, предназначенного для продажи остатков ассортимента биологически активных добавок в аптечных пунктах предприятия ГАУЗ «ОАС».

Во время прохождения практики необходимо выполнить следующие задачи:

- пройти инструктаж по технике безопасности;
- разработать техническое задание на веб-приложение в соответствии с требованиями заказчика;
- изучить предметную область;
- разработать концепции веб-приложения;
- разработать технический проект веб-приложения;
- разработать проектную документацию проекта;
- разработать эскизный проект веб-приложения;
- разработать модели пользовательского интерфейса;
- разработать перечень мероприятий по защите информации;
- разработать веб-приложения в соответствии с техническим заданием;
- разработать интерфейс пользователя веб-приложений в соответствии с техническим заданием;
- произвести тестирование кода веб-приложения;
- выполнить техническое сопровождение и восстановление веб-приложений в соответствии с техническим заданием;
- разместить веб приложения в сети в соответствии с техническим заданием;
- выполнить аудит безопасности веб-приложения в соответствии с регламентами по безопасности.

Объектом реализуемой системы является – Интернет-магазин по продаже биологически активных добавок.

Предмет реализуемой системы является - является разработка веб-приложения Интернет-магазина, предназначенного для продажи остатков ассортимента биологически активных добавок в аптечных пунктах предприятия ГАУЗ «ОАС».

# 1 Анализ предприятия

## 1.1 Характеристика предприятия

Государственное автономное учреждение здравоохранения «Областной аптечный склад», расположенное по адресу город Оренбург, улица Березка 24, необходимо для обеспечения медикаментами и товарами медицинского назначения лечебно-профилактических учреждений и населения области, закупаемым за счет средств областного и федерального бюджетов. Сегодня розничная сеть ГАУЗ «ОАС» представлена 128 обособленными аптечными подразделениями. Более 120 аптек и аптечных пунктов осуществляют льготное лекарственное обеспечение граждан. Аптечная сеть компьютеризирована, разработано и внедрено несколько программных продуктов для персонифицированного учета, анализа движения медикаментов в разрезе препаратов.

Основной целью создания Учреждения является обеспечение лекарственными средствами для медицинского применения медицинских организаций, аптек, прочих юридических лиц, индивидуальных предпринимателей и населения области.

Предметом деятельности являются:

- приобретение, прием, хранение, отпуск, реализация, перевозка лекарственных средств для медицинского применения, в том числе наркотических средств и психотропных веществ;
- приобретение, прием, хранение, отпуск и реализация изделий медицинской техники и ортопедических изделий;
- изготовление лекарственных препаратов для медицинского применения;
- льготный отпуск лекарственных препаратов и изделий медицинского назначения;
- уничтожение наркотических средств и психотропных веществ;
- накопление и уничтожение недоброкачественных лекарственных препаратов.

Для достижения поставленной цели Учреждение осуществляет следующие основные виды деятельности

- фармацевтическую деятельность;
- деятельность по обороту наркотических средств, психотропных веществ и их прекурсоров, культивированию наркосодержащих растений;
- организация информационного взаимодействия между участниками. льготного лекарственного обеспечения граждан, для которых при амбулаторном лечении меры социальной поддержки гарантированы законодательством Российской Федерации;
- сбор и обработка сведений областных регистров граждан, страдающих социально значимыми заболеваниями, отдельных групп и категорий граждан при амбулаторном лечении;

– сбор и обработка сведений, полученных от медицинских организаций и внесение их в Федеральный регистр больных, страдающих редкими (орфанными) заболеваниями и Федеральный регистр больных злокачественными новообразованиями лимфоидной, кроветворной и родственных им тканей, гемофилией, муковисцидозом, гипофизарным нанизмом, болезнью Гоше, рассеянным склерозом, а также после трансплантации органов и (или) тканей;

– осуществление оперативной проверки качества лекарственных препаратов, закупленных для льготного лекарственного обеспечения граждан, для которых при амбулаторном лечении меры социальной поддержки законодательством Российской Федерации.

Организационно-структурная схема организации отображает ее структуру и состав подразделений, а также связи их между собой. В ГАУЗ «ОАС» главой организации является директор, в подчинении которого стоят его заместители по отделам, а также главный бухгалтер. Заместитель директора по ИТ занимается вопросами обеспечения предприятия информационными технологиями, такими как различные программы и сайты, разработкой новых программ.

На рисунке 1 представлена организационно-структурная схема ГАУЗ «ОАС».

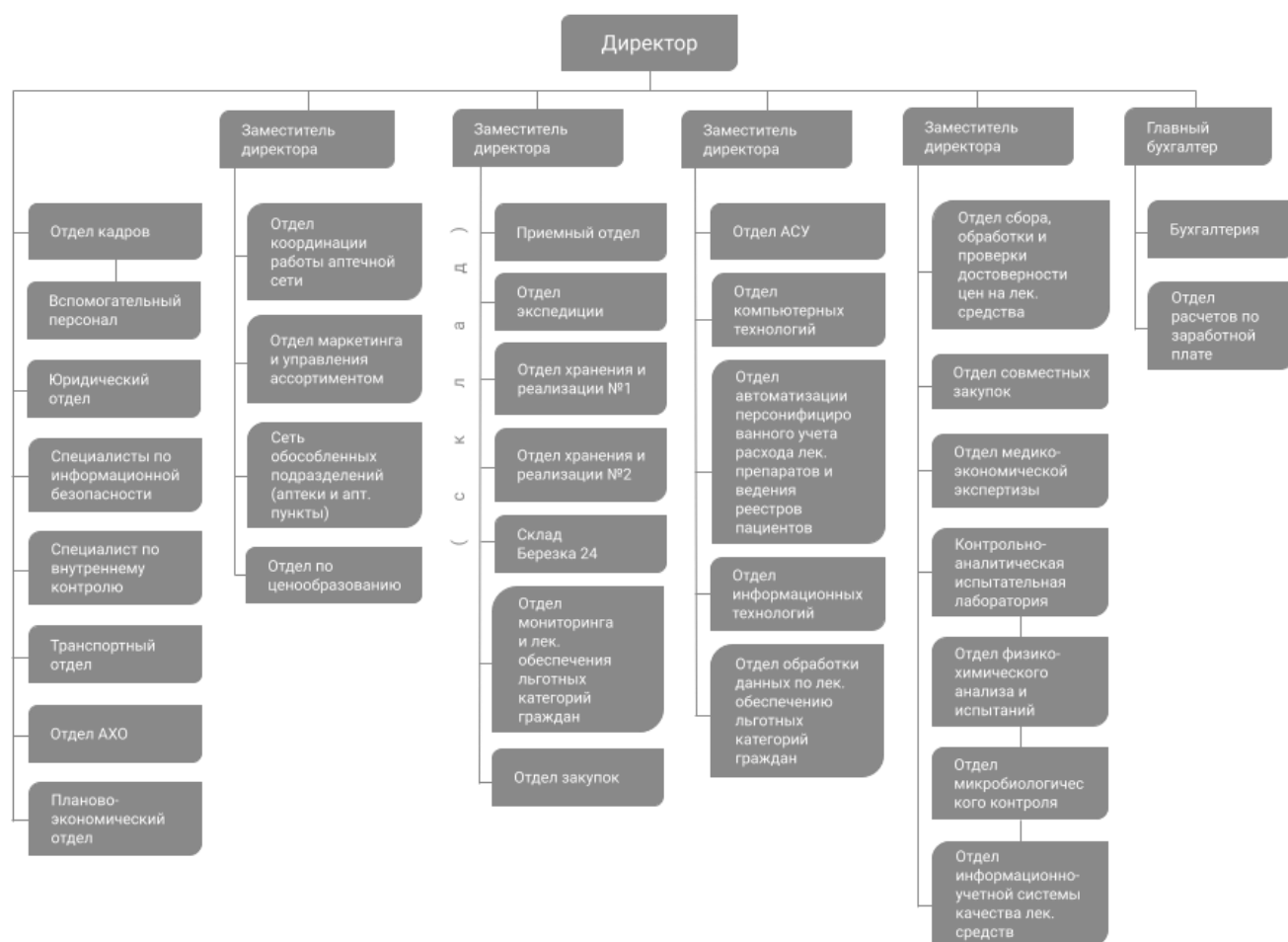


Рисунок 1 – Организационно-структурная схема ГАУЗ «ОАС»

## 1.2 Анализ предметной области

Организация выдвигает требование разработать и внедрить веб-приложение, которое будет предназначен для продажи остатков ассортимента биологически активных добавок в аптечных пунктах предприятия ГАУЗ «ОАС». Интернет-магазины для продажи БАДов — это специализированная онлайн площадка, где можно приобрести разнообразные виды биологически активных веществ и их композиций, предназначенных для непосредственного приёма с пищей или введения в состав пищевых продуктов.

Реализация проекта с использованием современных веб-технологий обеспечивает не только функциональность, но и способствует легкости разработки, обеспечивая высокую производительность и удовлетворение пользовательских потребностей.

В функционале сайта присутствуют такие возможности, как:

- просмотр основного каталога товара;
- возможность сортировки товара по каким-либо критериям;
- возможность поиска товара по торговому наименованию;
- добавление товаров в корзину только авторизованным пользователям;
- предоставление основной информации о компании;
- возможность оформления заказа только авторизованным пользователям;
- возможность просмотра информации о заказах и их статусе;
- возможность создания личного кабинета;
- возможность просмотра и обновления состояния заказа работниками аптечного пункта.

Грамотно созданный интернет-магазин является удобным инструментом маркетинга по активному продвижению товаров на рынок, что в конечном счете ведет к выполнению основной задачи интернет-магазина – увеличению прибыли от продажи товаров и большее количество новых посетителей и покупателей. При создании ИС появляется неограниченная база потенциальных клиентов, которые могут ознакомиться с ассортиментом, независимо от своего местонахождения, также появляется неограниченная виртуальная площадь витрин, позволяющая продемонстрировать весь товар, описав достоинства каждого.

Учитывая эти требования, интернет-магазин для продажи БАДов сможет привлекать больше покупателей и быть успешным в своей деятельности. Внедрение веб-сайта для предприятия позволит изменить работу аптечных пунктов, существенно повысить количество клиентов, приобретающих биологически активные добавки и расширить показ основной информации товаров. Таким образом, разработка веб-сайта с экономической точки зрения эффективна.

Таким образом, для реализации веб-приложения необходимо будет определить основные требования к его разработке.

## 2 Техническое задание

Работа выполняется на основании задания на производственную практику. Цель создания веб-приложения: реализация продаж биологически активных добавок от предприятия. Цель, поставленная на производственную практику: создание современного и отзывчивого интернет-магазина для продажи биологически активных добавок (БАДов), разработанного с использованием React.

В роли заказчика выступает компания ГАУЗ «ОАС».

Адрес фактический: 460044, Россия, Оренбургская обл., г. Оренбург, ул.Березка, 24.

Телефон / Почта: +7(3532)507-507, [office@oas56.ru](mailto:office@oas56.ru).

В роле исполнителя выступает: Галимьянова Элеонора Руслановна.

Плановые сроки начала и окончания работ: 21.03.24 – 17.04.24.

Отдел, ответственной за поддержания работоспособности приложения: отдел информационных технологий под руководством заместителя директора по ИТ.

Основными пользователями системы являются люди средней и старшей возрастных категорий, возрастом от 16 до 60 лет.

### 2.1 Требования к серверной части

К серверной части приложения предъявляется ряд некоторых требований.

Должен использоваться мощный многоядерный процессор, например, Intel Xeon или AMD Ryzen, чтобы обеспечить быструю обработку запросов. Количество ядер и тактовая частота должны быть достаточными для обработки ожидаемой нагрузки. Минимальное количество ядер для сервера интернет-магазина может быть 4 и более. Необходимо выбрать процессор с достаточно высокой тактовой частотой для обеспечения быстрой обработки запросов, процессоры с тактовой частотой от 2.5 ГГц и выше могут обеспечить хорошую производительность.

Минимальный объем оперативной памяти для сервера интернет-магазина должен составлять от 8 ГБ и выше.

Размер жесткого диска зависит от объема данных, которые необходимо хранить на сервере. Необходимо использовать SSD-накопители для ускорения загрузки страниц интернет-магазина и обеспечения быстрого доступа к данным. Минимальный размер SSD должен быть от 256 ГБ и выше.

Сервер должен иметь стабильное и быстрое подключение к интернету с достаточной пропускной способностью для обработки запросов пользователей.

Обеспечение безопасности сервера с помощью обновлений ПО, установки антивирусного программного обеспечения, фильтрации сетевого трафика и других мер защиты от вредоносных атак. При проектировании серверной инфраструктуры



необходимо учитывать возможность масштабирования системы в будущем в случае увеличения нагрузки или расширения бизнеса.

## 2.2 Требование к разработке

Данный сайт будет реализован посредством использования современных веб-технологий и языков. Под термином «технологический стек» понимают сложную комбинацию, включающую языки программирования, программное обеспечение и спектр фреймворков, применяемых для разработки IT-проекта. Архитектура любого web-приложения включает две стороны – клиентскую и серверную. Клиентской частью считаются визуализирующиеся данные, доступные пользователям на дисплее, которые используются посетителями сайта. Клиентская часть представляет собой визуализирующиеся данные, доступные пользователям на экране. Эти данные используются посетителями сайта во время их взаимодействия с ресурсом.

Технологический стек:

- фронтенд: React, React Router;
- управление состоянием: Context API, Redux;
- AJAX запросы: Fetch API, Axios;
- стилизация: CSS;
- тестирование: Jest, React Testing Library;
- сборка проекта: Webpack;
- инструменты контроля версий: Git.

Разработка серверной стороны включает в себя использование следующего технологического стека:

- базы данных, в данном случае, PostgreSQL, обеспечивающей хранение и эффективное управление данными проекта;
- веб-фреймворк Express.js для языка программирования JavaScript. Express.js, веб-фреймворк для языка программирования JavaScript, обеспечивает эффективное создание серверной части веб-приложения.

Все вышеперечисленные технологические компоненты интегрируются для создания современного и функционального веб-приложения. Их взаимодействие обеспечивает оптимальную работу как клиентской, так и серверной стороны проекта.

Для реализации некоторых функций могут быть использованы библиотеки и фреймворки. Для разработки серверной части данной системы должны быть использованы следующие библиотеки:

- bcryptjs (2.4.3) - библиотека для хеширования паролей с использованием алгоритма bcrypt. Используется для безопасного хранения паролей пользователей;
- cors (^8.5) - пакет для управления CORS (Cross-Origin Resource Sharing) в Express приложениях. Позволяет настраивать доступ к ресурсам на сервере из других источников или доменов;

- express (4.18.3) - фреймворк для создания веб-приложений на Node.js. Используется для построения серверной части системы;
- jsonwebtoken (9.0.2) - библиотека для работы с JSON Web Tokens (JWT). Используется для аутентификации и авторизации пользователей;
- multer (1.4.5) - middleware для обработки данных форм в Node.js. В данной системе используется для обработки загружаемых файлов;
- nodemon (3.1.0) - утилита, облегчающая разработку Node.js приложений путем автоматической перезагрузки сервера при изменении файлов проекта;
- pg (8.11.3) - клиент PostgreSQL для Node.js. Предоставляет API для взаимодействия с базой данных PostgreSQL из приложений Node.js;
- postgres (3.4.3) - драйвер PostgreSQL для Node.js. Позволяет устанавливать соединение с базой данных PostgreSQL и выполнять запросы к ней.

Для разработки клиентской части системы были использованы следующие библиотеки:

- @reduxjs/toolkit (2.2.1): эта библиотека предоставляет удобные инструменты для управления состоянием приложения в React. Redux Toolkit упрощает процесс создания Redux-хранилища, управления действиями и редукторами, а также интеграции с React компонентами;
- react (18.2.0): react — это библиотека JavaScript для создания пользовательских интерфейсов. В данной системе React используется для создания пользовательского интерфейса и его компонентов;
- react-dom (18.2.0): react-DOM — это пакет, который обеспечивает взаимодействие React с DOM (Document Object Model). Он используется для рендеринга React компонентов в браузере;
- react-redux (9.1.0): react-Redux — это официальная библиотека, которая интегрирует Redux с React. Она облегчает передачу состояния из Redux-хранилища в React компоненты и управление ими;
- react-router-dom (6.20.1): react-Router-DOM — это библиотека для навигации в React приложениях. Она позволяет определять маршруты приложения и связывать их с соответствующими React компонентами.

## 2.3 Тестирование и документация

После завершения разработки приложения будет проведено тестирование для проверки его функциональности, надежности и производительности.

Функциональное тестирование:

- проверка работы всех основных функций интернет-магазина, таких как просмотр каталога товаров, добавление товаров в корзину, оформление заказа;
- проверка адаптивности и отзывчивости интерфейса на различных устройствах и разрешениях экранов;
- тестирование взаимодействия с API для получения данных о товарах и обработки заказов.

Тестирование безопасности:

- проверка на уязвимости, такие как XSS и CSRF;
- тестирование аутентификации и авторизации пользователей;
- проверка безопасности обработки платежей и личной информации пользователей.

Тестирование производительности:

- проведение нагрузочного тестирования для определения максимальной нагрузки, которую может выдержать приложение;
- оценка времени загрузки страниц и отклика интерфейса при различных условиях нагрузки.

Также будет составлена документация, описывающая процесс разработки, функциональные возможности приложения, а также инструкции для пользователей и администраторов.

Техническая документация:

- описание архитектуры приложения, используемых технологий и инструментов;
- инструкции по установке и запуску приложения на локальной машине разработчика;
- описание структуры базы данных и взаимодействия с API.

Пользовательская документация:

- инструкции для пользователей по использованию интерфейса приложения: регистрация, просмотр товаров, оформление заказов;
- руководство по безопасности: рекомендации по сохранению личных данных и безопасному использованию интернет-магазина.

Административная документация:

- инструкции для администраторов по управлению контентом и товарами, добавлению новых товаров, управлению заказами;
- описание процедур обновления и поддержки приложения.

## 3 Разработка проектной документации

### 3.1 Разработка концепции веб-приложения

В функционале сайта должны присутствовать такие возможности, как:

- просмотр основного каталога товара;
- возможность сортировки товара по каким-либо критериям;
- возможность поиска товара по торговому наименованию;
- добавление товаров в корзину только авторизованным пользователям;
- предоставление основной информации о компании;
- возможность оформления заказа только авторизованным пользователям;
- возможность просмотра информации о заказах и их статусе;
- возможность создания личного кабинета;
- возможность просмотра и обновления состояния заказа работниками аптечного пункта.

Общая структура страниц сайта для клиента состоит из следующих логических страниц:

- главная страница, включающая отображение краткой информации о продукции, навигационные элементы для доступа к другим разделам приложения;
- карточка товаров, включающая отображение подробной информации о продукте, включая название, изображение, описание, цену;
- личный кабинет, включающая возможность просмотра и редактирования персональной информации пользователя, историю заказов пользователя;
- страница входа/регистрации системы, включающая формы регистрации/авторизации;
- страница оформления заказа, включающая форму выбора аптечного пункта и заполнения контактных данных
- корзина, включающая отображение добавленных в корзину товаров, возможность изменения количества товаров и удаления из корзины;
- ознакомительная страница с общей информацией про БАДы, включающая информацию о пользе и применении БАДов;
- страница контактов предприятия, включающая информацию об аптечных пунктах и форму для связи с клиентами.

Структура страниц сайта для работников ГАУЗ «ОАС» включает в себя:

- страница входа/регистрации системы, включающая формы регистрации/авторизации;
- страница с информацией о заказах конкретного аптечного пункта, включающая возможность обновления статуса заказа.

В системе должно иметься разграничение доступа пользователей к страницам. Разграничение пользователей идет по ролям, присваиваемым им при регистрации. Присваивание ролей производится администратором БД. Авторизованный пользователь имеет доступ ко всем страницам, кроме страниц добавления товара и страниц, созданных для работников аптек. Работникам аптек

предоставляется доступ к рабочим страницам с просмотром заказов в данном аптечном пункте.

Для организации доступа пользователей должен использоваться протокол презентационного уровня HTTP и его расширение HTTPS.

IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Данная методология, разработанная для описания системы в целом, представлена в приложении А на рисунке А1. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией. Данная диаграмма для разрабатываемой системы представлена в приложении А на рисунке А2.

### 3.2 Разработка эскизного проекта веб-приложения

Макет веб-сайта представляет собой первоначальный визуальный план или набросок внешнего вида сайта, который создается перед началом активной разработки. Он играет ключевую роль в процессе проектирования и разработки веб-приложения или сайта, позволяя визуализировать и обсудить концепцию дизайна и структуры. Создание эскиза играет важную роль в визуализации и концептуализации дизайна сайта. Эскизы помогают продумать компоновку контента на сайте, которая имеет ключевое значение для понятности, привлекательности и удобства использования сайта, задать основное направления развития дизайна сайта в дальнейшем его создании. В последующем, данные эскизы могут быть использованы в качестве отправной точки для разработки конечного дизайна сайта.

Прототипы главной страницы системы, страницы карточки товара, страницы авторизации/регистрации, страницы личного кабинета пользователя, страницы оформления, страницы корзины покупаемой продукции в ИС, информационной страницы про БАДы, страницы контактов организации представлены на рисунке 2-10.

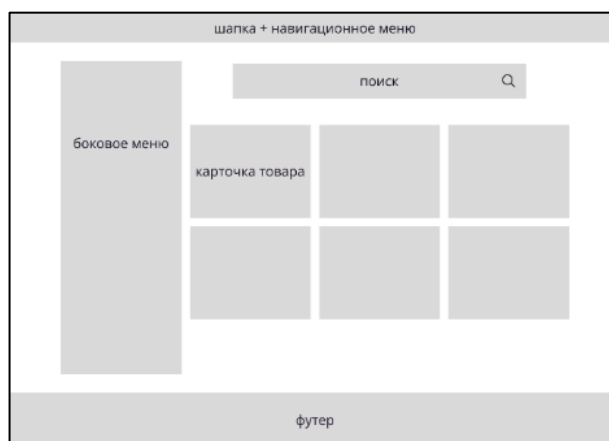


Рисунок 2 – Прототип главной страницы системы

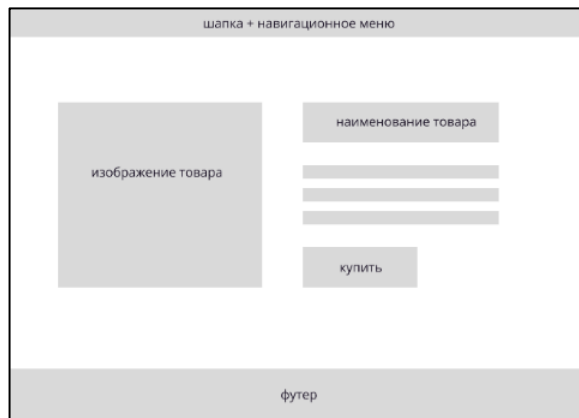


Рисунок 3 – Прототип страницы карточки товара



Рисунок 4 – Прототип страницы авторизации

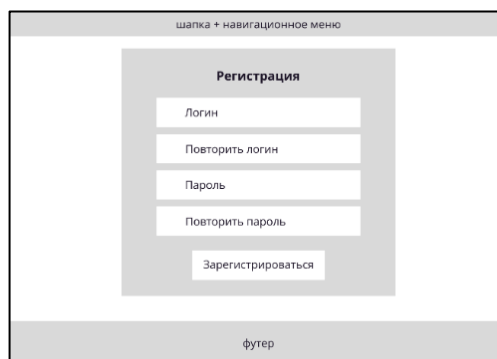


Рисунок 5 – Прототип страницы регистрации

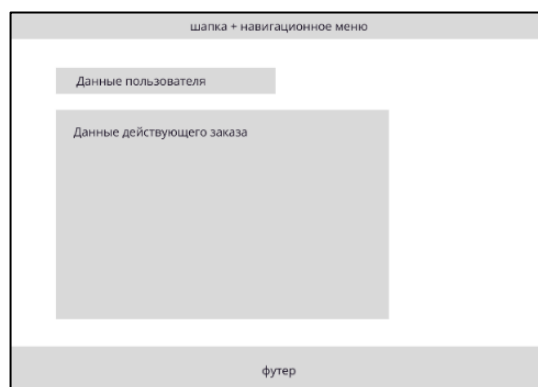


Рисунок 6 – Прототип страницы личного кабинета

шапка + навигационное меню

### Оформление заказа

ФИО заказчика

Телефон заказчика

Самовывоз/доставка

Место самовывоза/доставки

оформить заказ

футер

Рисунок 7 – Прототип страницы оформления заказа

шапка + навигационное меню

**наименование товара**

информация о товаре:

кол-во товара:

**оформить заказ**

кол-во продукции:

**цена:**

заказать

футер

Рисунок 8 – Прототип страницы корзины пользователя

шапка + навигационное меню

Заголовок страницы

фото

описание

описание

фото

футер

Рисунок 9 - Прототип страницы со статьями

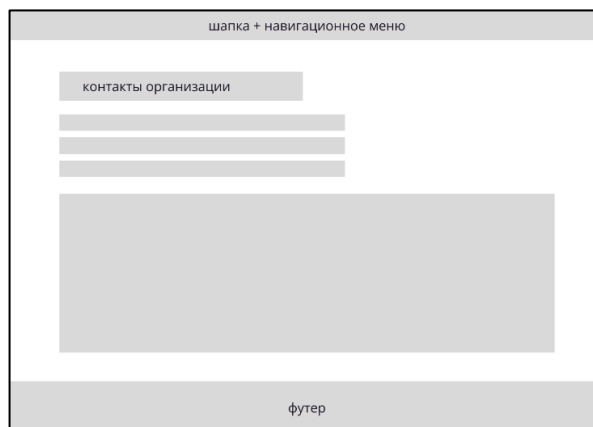


Рисунок 10 – Прототип страницы контактов организации

### 3.3 Разработка моделей пользовательского интерфейса

При работе с системой выделены 3 основных вида пользователей: неавторизованный пользователь, авторизованный пользователь и работник аптечного пункта. Диаграмма прецедентов использования (use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. Данная диаграмма изображена в приложении В на рисунке А3.

К интерфейсу и дизайну веб-сайта предъявляется ряд общих требований, а именно:

- разработка привлекательного и интуитивно понятного дизайна пользовательского интерфейса с использованием Figma;
- удобная навигация и структура страниц для улучшения пользовательского опыта;
- создание адаптивного дизайна, который будет хорошо отображаться на различных устройствах и разрешениях экранов.

Основными формами взаимодействия пользователя с системой на первом этапе будут формы авторизации и регистрации. Их макеты представлены на рисунке 11 и 12. На странице авторизации должны иметься поля ввода номера телефона и пароля. При неправильном вводе пароля должна всплывать ошибка о введении неверного пароля, при неправильном вводе номера телефона, должна всплывать ошибка о несуществовании данного пользователя в системе. Также при не заполнении обязательных полей должно всплывать сообщение, что данное поле обязательно для заполнения пользователем. На странице авторизации должна быть ссылка на страницу регистрации пользователя.

На странице регистрации пользователя должны иметься следующие поля для ввода информации: введения номера телефона, введения фамилии имени пользователя, введения пароля и поле введения повторного пароля. При несовпадении введения повторного пароля должна всплывать ошибка о том, что



пароли не совпадают. При успешной регистрации пользователя должно возвращать к странице авторизации.

Рисунок 11 – Макет страницы авторизации

Рисунок 12 – Макет страницы регистрации

В системе имеется компонент шапки товара, представленный на рисунке 13. Шапка включает в себя логотип системы, навигационное меню и иконки взаимодействия. Если пользователь не авторизован, то чтоб получить доступ к корзине будет всплывать модальное окна с просьбой авторизоваться и только тогда пользователь получает доступ к корзине. Если пользователь не авторизован, то при наведении на иконку профиля будет всплывать меню с кнопкой «войти», если пользователь авторизован, то всплывает меню с кнопкой перехода в личный кабинет и кнопкой выхода из аккаунта.

Рисунок 13 – Компонент шапки системы.

Главная страница информационной системы, представленная на рисунке 14, должна состоять из следующих элементов:

- шапка сайта с навигационным меню и футер сайта;
- боковое меню с ссылками на конкретную категорию товаров, сортирующих товар по категориям;
- строка поиска товара по торговому наименованию;
- торговые карточки товара с ссылкой на конкретную товарную позицию.

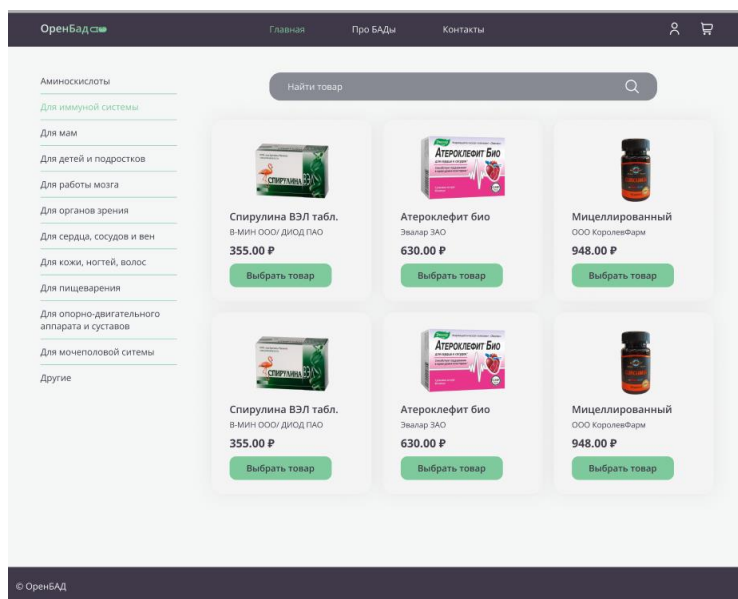


Рисунок 14 – Макет главной страницы системы

Страницы товара в информационной системе, представленная на рисунке 15, должна состоять из следующих элементов:

- изображение товара и его основная информация, включающая торговое наименование продукта, производителя, описания товара, цену и информацией о доставке и оплате товара;
- после выбора конкретной аптеки появляется информация о количестве товара в этой аптеке, счетчик добавления количества товара и кнопка добавления товара в корзину. Если пользователь не авторизирован, то всплывает окно с просьбой об авторизации, после чего появляется возможность добавления товара в корзину.

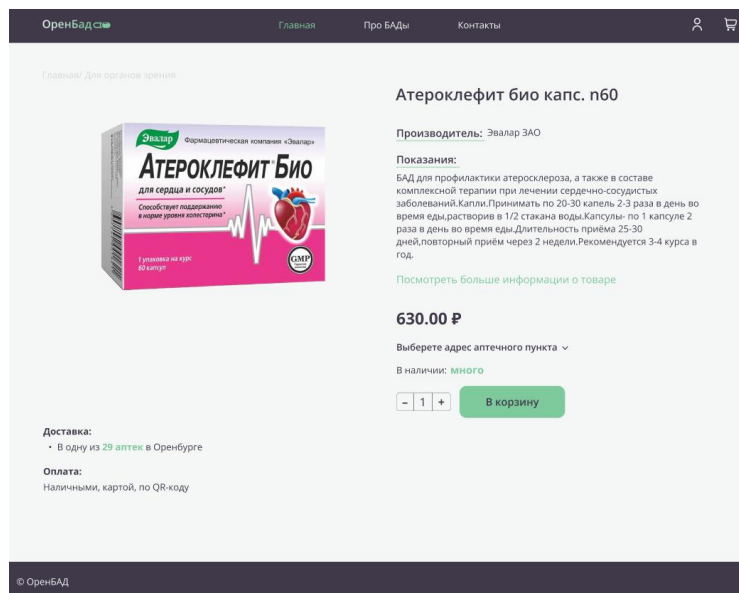


Рисунок 15 – Макет страницы карточки товара

Страница личного кабинета пользователя системы, представленная на рисунке 16, должна состоять из следующих элементов:

- данные пользователя, с возможностью перехода на форму взаимодействия для изменения данных профиля, таких как номера телефона и фамилии имени;
- данный действующего заказа пользователя (если имеется), с возможностью перехода на страницу с подробной информацией о продуктах заказа. Если заказов еще не было, должен быть текст “заказов пока нет”. В карточке заказов должен быть выведен один из статусов заказа: “в обработке”, “готов к получению”, “отменён”, которые изменяет администратор аптечного пункта.

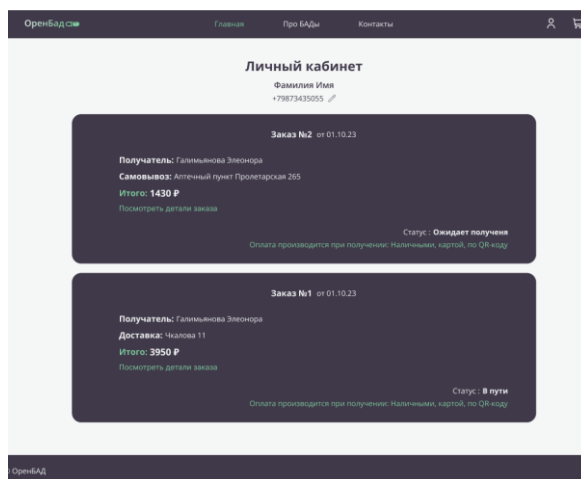


Рисунок 16 – Макет страницы личного кабинета

Страница корзины покупаемой продукции в ИС, представленная на рисунке 17, должна состоять из следующих элементов:

- карточка выбранных товаров с возможностью изменять количество товара и удалять товар из корзины с информацией о выбранном аптечном пункте

(данные об аптечном пункте передаются с карточки товара, где пользователь выбирает аптечный пункт из списка доступных);

– форма данных заказа с кнопкой заказа товаров.

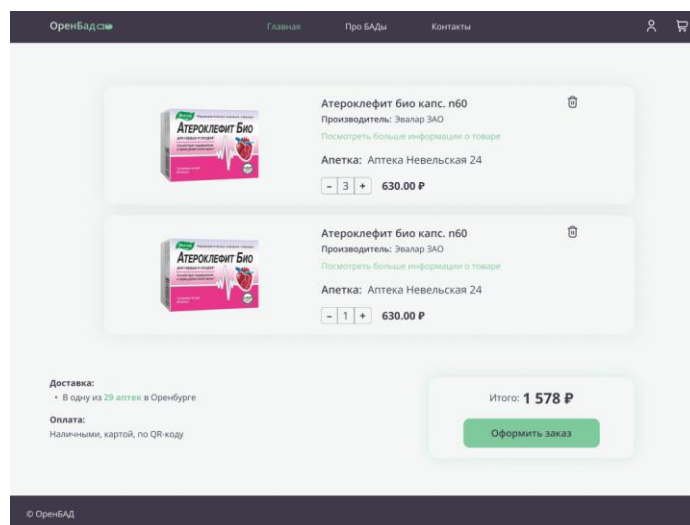


Рисунок 17 – Макет страницы корзины пользователя

Страница оформления заказа, представленная на рисунке 18, должна состоять из следующих элементов:

– форма оформления заказа с заполнением личных данных получателя товара.

Оформление заказов должно производиться только авторизованными пользователями. Если пользователь не авторизован, должно всплывать окно с просьбой авторизоваться, с кнопкой, ведущей на страницу авторизации. Оформленный пользователем заказ будет забронирован от покупок другими людьми в течении 7 дней, позже заказ отменяется администратором.

Информация в каком аптечном пункте будет производится выдача товара берется из корзины товара, в которой пользователь, добавляя товар, выбирает аптечный пункт.

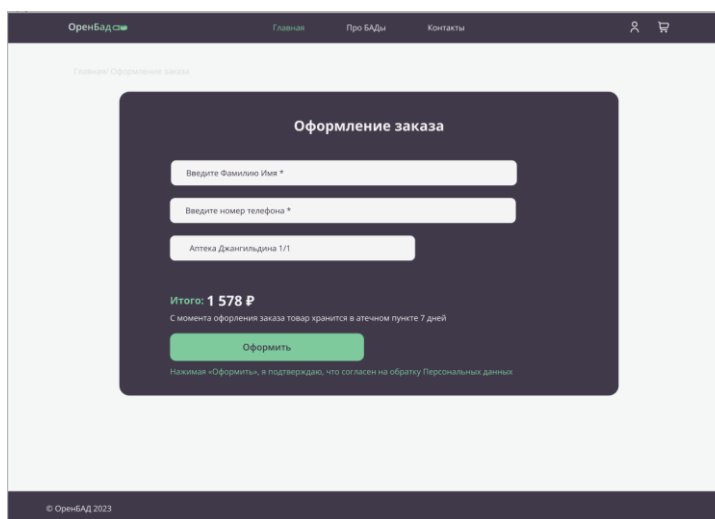


Рисунок 18 – Макет страницы оформления заказа

### 3.4 Обеспечение защиты информации

Обеспечение безопасности информации в интернет-магазине продажи БАДов на React является критически важным аспектом, учитывая конфиденциальность данных пользователей и финансовые транзакции.

К безопасности веб-сайта предъявляется ряд общих требований, а именно:

- обеспечение безопасности приложения и данных пользователей с использованием соответствующих методов шифрования паролей, защиты от атак и утечек данных;

- внедрение аутентификации и авторизации для доступа к личным данным и функционалу сайта.

Политика конфиденциальности:

- система обязуется защищать личные данные пользователей и обрабатывать их в соответствии с законодательством о защите персональных данных;

- пользователи должны иметь возможность управлять своими личными данными, включая просмотр, изменение и удаление информации из своего профиля.

Ниже приведены ключевые меры безопасности:

- шифрование данных. Все конфиденциальные данные, такие как пароли пользователей, должны быть зашифрованы при передаче между клиентом и сервером, это можно обеспечить с использованием протоколов шифрования, таких как SSL/TLS;

- аутентификация и авторизация. Реализация механизмов аутентификации для проверки подлинности пользователей при входе в систему. После аутентификации необходимо установить соответствующие права доступа для каждого пользователя, чтобы предотвратить несанкционированный доступ к конфиденциальной информации;

- защита от CSRF и XSS атак. Необходимо применять соответствующие меры для предотвращения атак типа CSRF и XSS, включая использование токенов CSRF и корректную обработку и экранирование данных, вводимых пользователями;

- безопасность базы данных. Обеспечение безопасности базы данных PostgreSQL, включая ограничение доступа к базе данных, использование параметризованных запросов для предотвращения SQL-инъекций и регулярное обновление программного обеспечения для устранения уязвимостей.

## 4 Разработка веб-приложения

### 4.1 Реализация веб-приложения

К функциональности веб-сайта предъявляется ряд общих требований, а именно:

- реализация основных функций, таких как просмотр каталога товаров, добавление товаров в корзину, оформление заказа, регистрация/аутентификация пользователей;

- обработка и валидация данных, введенных пользователями в формы;

- реализация системы управления пользователями, заказами и товарами.

К бэкенд-части веб-сайта предъявляется ряд общих требований, а именно:

- создание и настройка базы данных PostgreSQL для хранения информации о пользователях, заказах, товарах и других сущностях;

- разработка API для взаимодействия между фронтендом и базой данных, а также для интеграции с внешними сервисами.

Первоначальный этап разработки веб-приложения включает в себя планирование и анализ системы. Планирование и анализ являются ключевыми этапами в разработке веб-сайта и играют важную роль в обеспечении успешной реализации проекта.

Следующим этапом в разработке сайта являются проектирование и разработка дизайна. Необходимо разработать структуру информации и создать макеты страниц, разработать дизайн, учитывая корпоративный стиль и UX/UI принципы, выбрать цветовую палитру, шрифты и другие дизайнерские элементы.

Следующим этапом является анализ организации и создание базы данных в PostgreSQL. Созданная база данных моделирует систему для управления аптечным магазином. Схема базы данных представлена на рисунке А4.

После чего идет разработка бэкенда (Node.js). Создается серверное приложение на Node.js с использованием Express.js. Нужно установить и настроить среду разработки для Node.js. С помощью команды `npm init` создается файл `package.json` для своего приложения. Файл `package.json` позволяет указать диапазон версий пакетов, которые нужно ставить.

Важной частью является создание обработчика запросов для аутентификации (auth) и регистрации (reg) пользователей в системе.

Аутентификация (auth). Этот обработчик используется для проверки аутентификационных данных пользователя (телефон и пароль). При получении запроса извлекается телефон из тела запроса и выполняется запрос к базе данных для поиска пользователя по этому номеру телефона. Если пользователь не найден или пароль не совпадает с хэшированным паролем в базе данных, возвращается соответствующее сообщение об ошибке. Если аутентификация проходит успешно, генерируется токен доступа с помощью функции `generateAccessToken` и возвращается вместе с информацией о пользователе.

Регистрация (reg). Этот обработчик используется для регистрации новых пользователей в системе. При получении запроса извлекаются имя, телефон и пароль из тела запроса. Выполняется запрос к базе данных для проверки, существует ли уже пользователь с указанным телефоном. Если пользователь уже существует, возвращается сообщение об ошибке. В противном случае пароль хэшируется с помощью bcrypt, и данные о пользователе добавляются в базу данных. Также создается запись в таблице Basket для нового пользователя. После успешной регистрации генерируется токен доступа и возвращается вместе с информацией о пользователе и сообщением о успешной регистрации.

Также важной частью серверной части является создание обработчиков маршрутов для осуществления запросов к серверной части через фронтенд часть системы.

GET/category. Обрабатывает запросы на получение списка категорий продуктов из базы данных.

GET/products. Обрабатывает запросы на получение списка всех продуктов из базы данных с подключением таблицы Maker и Category.

GET/category/:id. Обрабатывает запросы на получение продуктов по указанной категории.

GET/product/:id. Обрабатывает запросы на получение информации о конкретном продукте по его идентификатору.

POST/reg. Обрабатывает запросы на регистрацию новых пользователей.

POST/auth. Обрабатывает запросы на аутентификацию пользователей.

POST/newproduct. Обрабатывает запросы на добавление нового продукта в базу данных, включая загрузку изображения.

Обработчики маршрутов для создания новых категорий и производителей:

POST/category. Обрабатывает запросы на создание новой категории продуктов.

POST/maker. Обрабатывает запросы на создание нового производителя.

Далее идет разработка фронтенда (React). Разработка фронтенда в среде React - процесс, который представляет собой создание интерактивных и динамичных пользовательских интерфейсов для веб-приложений.

В данной системе происходит разделение пользователей на основе их роли. Система имеет три типа пользователей: обычный пользователь (USER), фармацевт (PHARM). Роль каждого пользователя хранится в базе данных в таблице Roles.

Разделение маршрутов. Созданы несколько экземпляров маршрутизатора BrowserRouter с разными наборами маршрутов для разных ролей пользователей. Для каждого типа пользователя определен свой набор маршрутов, доступных только для этой роли. Компонент App определяет текущего пользователя по токenu и роли, полученным из состояния Redux. В зависимости от роли пользователя и наличия токена, компонент App рендерит соответствующий маршрутизатор (RouterProvider), который определен для этой роли. Если пользователь не аутентифицирован (т.е., токен отсутствует), рендерится маршрутизатор для общего доступа. Код реализации серверной и клиентской части веб-приложения представлен в приложении Б.

## 5 Тестирование веб-приложения

### 5.1 Тестирование кода веб-приложения

Для тестирования веб-приложения был выбран метод ручного тестирования. Ручное тестирование веб-приложения для продажи БАДов включает в себя проверку всех основных функций, интерфейса пользователя и обработки ошибок. Чтобы его выполнить был проведен контрольный пример.

Проверка авторизации и аутентификации:

- регистрация нового пользователя;
- авторизация существующего пользователя;
- проверка входа и выхода из учетной записи пользователя.

При попытке авторизации/регистрации в системе учитывается обязательность заполнения полей (рисунок 19).

The screenshot shows a registration form titled "Регистрация" on a dark background. It contains four input fields: "Введите номер телефона:", "Введите вашу Фамилию Имя:", "Введите пароль:", and "Повторите пароль:". The second field has a red error icon and a tooltip that says "Заполните это поле." (Fill in this field). At the bottom is a green button labeled "Зарегистрироваться".

Рисунок 19 – Обязательность заполнения полей

При вводе неправильных данных при авторизации всплывает сообщение об ошибке (рисунок 20).

The screenshot shows an authorization form titled "Авторизация" on a dark background. It has two input fields: the first contains the phone number "89873435055" and the second contains two asterisks "\*\*". Below the fields is a green button labeled "Войти". At the bottom left, there is a link labeled "Регистрация". Below the form, a red error message reads "Пользователь не найден" (User not found).

Рисунок 20 – Проверка авторизации пользователей

Проверка функциональности продуктов:

- просмотр списка продуктов;
- просмотр информации о каждом продукте, включая изображение, название, описание и цену;
- добавление продукта в корзину;
- удаление продукта из корзины;



- обновление количества продуктов в корзине;
- проверка возможности оформления заказа.

Все функции, связанные с продуктами (просмотр списка, информации о продукте, добавление в корзину, удаление из корзины, обновление количества и оформление заказа), работают корректно.

Тестирование корзины:

- проверка добавления продуктов в корзину из разных категорий;
- проверка общего количества и суммы заказа в корзине;
- проверка возможности удаления продуктов из корзины;
- проверка обновления количества продуктов в корзине.

Итоги тестирования корзины: продукты успешно добавляются и удаляются из корзины, общее количество и сумма заказа корректно отображаются, обновление количества продуктов в корзине происходит без ошибок.

Тестирование производительности:

- проверка скорости загрузки страниц и элементов интерфейса;
- проверка производительности при выполнении операций, таких как добавление продуктов в корзину или оформление заказа.

Итоги тестирования производительности: скорость загрузки страниц и элементов интерфейса соответствует ожиданиям, производительность при выполнении операций стабильна и удовлетворительна.

## 5.2 Аудит безопасности веб-приложения

Разделение ролей внутри веб-приложения.

Пользователь: имеет доступ к функциям просмотра каталога товаров, добавления товаров в корзину, оформления заказа и управления своим профилем.

Работник аптеки: имеет доступ к заказам пользователей в конкретный аптечный пункт, может управлять статусом заказа.

Разделение ролей в рамках СУБД.

Администратор базы данных: имеет полный доступ к управлению базой данных, включая создание, изменение и удаление таблиц, а также управление пользователями и их привилегиями.

Пользователь базы данных: имеет доступ только к необходимым таблицам и операциям, связанным с веб-приложением, с ограниченными привилегиями на изменение структуры или данных.

Работники аптеки: имеет доступ только к необходимым таблицам и операциям, связанным с веб-приложением, с ограниченными привилегиями на изменение статуса заказов пользователей.

Итоги аудита безопасности веб-сайта позволяют сделать следующие выводы. Обеспечение безопасности приложения и данных пользователей:

- применяются соответствующие методы шифрования паролей и защиты от атак;

– система обеспечивает защиту данных пользователей при передаче между клиентом и сервером.

Внедрение аутентификации и авторизации:

– реализованы механизмы аутентификации для проверки подлинности пользователей при входе в систему;

– установлены соответствующие права доступа для каждого пользователя после аутентификации.

Политика конфиденциальности:

– система обязуется защищать личные данные пользователей и обрабатывать их в соответствии с законодательством о защите персональных данных;

– пользователи имеют возможность управлять своими личными данными, включая просмотр, изменение и удаление информации из своего профиля.

Шифрование данных:

– применяются протоколы шифрования, такие как SSL/TLS, для защиты конфиденциальных данных при передаче между клиентом и сервером.

Аутентификация и авторизация:

– реализованы механизмы аутентификации и установлены соответствующие права доступа для пользователей после аутентификации.

Защита от CSRF и XSS атак:

– применяются соответствующие меры для предотвращения атак типа CSRF и XSS, включая использование токенов CSRF и корректную обработку и экранирование данных, вводимых пользователями.

Безопасность базы данных:

– обеспечена безопасность базы данных PostgreSQL, включая ограничение доступа к базе данных и использование параметризованных запросов для предотвращения SQL-инъекций;

– проводится регулярное обновление программного обеспечения для устранения уязвимостей.

В целом, аудит позволяет заключить, что веб-сайт обеспечивает соответствующий уровень безопасности, предусматривает защиту конфиденциальных данных пользователей и соответствует основным требованиям безопасности. Однако, для обеспечения долгосрочной безопасности, рекомендуется проведение периодического аудита и обновление мер безопасности в соответствии с изменяющимися угрозами и требованиями.

## 6 Размещение и техническое сопровождение веб-приложения

### 6.1 Размещение веб-приложения

Регистрация на Netlify. Необходимо перейти на сайт Netlify и зарегистрироваться для создания учетной записи.

Добавление проекта. После регистрации необходимо войти в свою учетную запись на Netlify и нажать кнопку "New site from Git". Далее нужно выбрать свой репозиторий Git, в котором содержится веб-приложение на React.

Настройка параметров сборки. Следующим этапом необходимо выбрать настройки сборки для нашего проекта, указать команды для сборки проекта и развертывания (обычно это `npm run build`).

Запуск развертывания. Необходимо нажать кнопку "Deploy" для запуска процесса развертывания нашего веб-приложения на хостинге Netlify (рисунок 21).

Получение URL-адреса. После успешного развертывания нашего приложения Netlify предоставит нам URL-адрес нашего веб-приложения. Этот URL-адрес можно использовать для доступа к вашему приложению в сети.

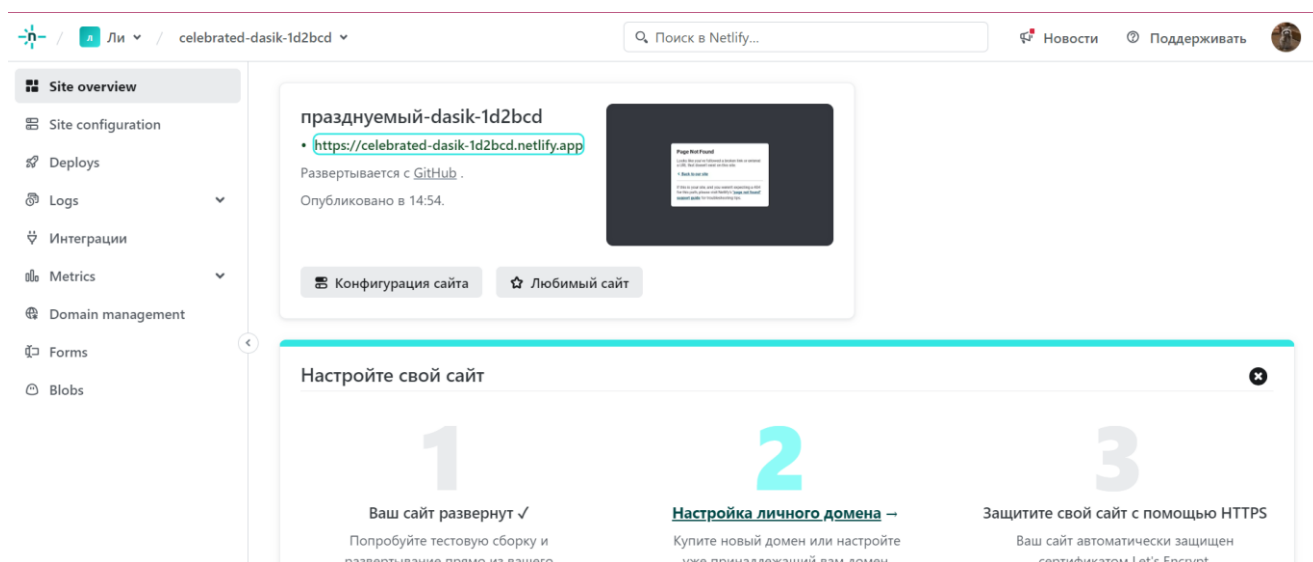


Рисунок 21 - Развертывание веб-приложения на хостинге Netlify

### 6.2 Техническое сопровождение

Инструкция по техническому сопровождению веб-приложения на React для продажи БАДов.

Регулярные резервные копии:

- выполнять резервное копирование базы данных и файлов приложения;
- рекомендуется выполнять резервное копирование еженедельно или в соответствии с уровнем активности приложения.

#### Мониторинг производительности:

- отслеживать производительность приложения, включая время загрузки страниц, использование ресурсов сервера и другие метрики;
- мониторинг производительности должен быть выполнен еженедельно или по мере необходимости.

#### Обновление зависимостей:

- проверять и обновлять зависимости приложения, включая библиотеки и фреймворки React, а также пакеты npm;
- рекомендуется выполнять проверку и обновление зависимостей ежемесячно или при появлении новых версий.

#### Управление контентом:

- обновлять содержимое приложения, такое как описания товаров, цены и изображения, в соответствии с требованиями заказчика;
- обновление контента должно быть выполнено при появлении новых товаров или изменении информации о существующих товарах.

#### Исправление ошибок:

- отслеживать и исправлять ошибки, которые могут возникнуть в процессе эксплуатации приложения;
- исправление ошибок должно выполняться немедленно после их обнаружения.

#### Анализ безопасности:

- проводить анализ безопасности приложения для выявления уязвимостей и принятия мер по их устранению;
- анализ безопасности рекомендуется проводить регулярно, например, ежеквартально или при внесении значительных изменений в приложение.

#### Обновление документации:

- поддерживать актуальную документацию по приложению, включая инструкции для пользователей и администраторов, а также техническую документацию;
- обновление документации может быть выполнено при внесении изменений в приложение или при обновлении функциональности.

## Заключение

Целью описываемого веб-приложения является осуществление продаж остатков ассортимента биологически активных добавок в аптечных пунктах ГАУЗ «ОАС» в городе Оренбург.

В ходе производственной практики были выполнены следующие задачи:

- пройден инструктаж по технике безопасности;
- разработано техническое задание на веб-приложение в соответствии с требованиями заказчика;
- изучена предметную область;
- разработаны концепции веб-приложения;
- разработан технический проект веб-приложения;
- разработана проектную документацию проекта;
- разработан эскизный проект веб-приложения;
- разработаны модели пользовательского интерфейса;
- разработан перечень мероприятий по защите информации;
- разработано веб-приложение в соответствии с техническим заданием;
- разработан интерфейс пользователя веб-приложений в соответствии с техническим заданием;
- произведено тестирование кода веб-приложения;
- выполнено техническое сопровождение и восстановление веб-приложений в соответствии с техническим заданием;
- размещено веб приложения в сети в соответствии с техническим заданием;
- выполнен аудит безопасности веб-приложения в соответствии с регламентами по безопасности.

В дальнейшем может быть реализована мобильная версия данного приложения.

Подводя итоги проделанной работы, я хочу отметить, что Интернет-магазин в современном понимании должен быть инструментом, направленным на улучшение сервиса, привлечения большего числа покупателей, уменьшение торговых и амортизационных издержек, экономичное размещение и демонстрацию товара, доступность самому широкому контенту, гибкую рекламу продаваемой продукции. Грамотно созданный интернет-магазин является удобным инструментом маркетинга по активному продвижению товаров на рынок, что в конечном счете ведет к выполнению основной задачи интернет-магазина – увеличению прибыли от продажи товаров и большее количество новых посетителей и покупателей.

Результатом моей практической работы стал готовое веб-приложение по продаже биологически активных добавок. В ходе работы я выделила основные задачи сайта и проанализировала различные возможные решения проблемы предприятия и выбрала наиболее подходящее в данной ситуации. Разработала клиентскую и серверную часть приложения и запустила сайт на хостинге.

## Список используемых источников

- 1 ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
- 2 ГОСТ 19.102-77 ЕСПД. Стадии разработки.
- 3 ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам
- 4 ГОСТ 19.402-2000 ЕСПД. Описание программы. Требования к содержанию, оформлению и контролю качества.
- 5 ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению.
- 6 ГОСТ 19.504-79 ЕСПД. Руководство программиста. Требования к содержанию и оформлению.
- 7 ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам
- 8 ГОСТ 2.106-96 ЕСКД. Текстовые документы
- 9 ГОСТ 3.1109-82. ЕСТД. Термины и определения основных понятий.
- 10 UML для бизнес-моделирования: зачем нужны диаграммы процессов. — Текст: электронный // evergreens.com: [сайт]. — Режим доступа: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (дата обращения: 01.04.2024).
- 11 What is PostgreSQL? — Текст: электронный // aws.amazon.com: [сайт]. — Режим доступа: <https://aws.amazon.com/ru/rds/postgresql/what-is-postgresql/> (дата обращения: 03.04.2024)
- 12 Базово о React: что это такое и как помогает разработчику — Текст: электронный // practicum.yandex.ru: [сайт]. — Режим доступа: <https://practicum.yandex.ru/blog/chto-takoe-react-i-kak-on-rabotaet/> (дата обращения: 22.03.2024)
- 13 ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы: Термины и определения.
- 14 ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
- 15 ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
- 16 ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
- 17 ГОСТ 7.1-2003 Система стандартов по информации, библиотечному и издательскому делу. Библиографическое описание документа. Общие требования и правила составления
- 18 Десять советов по оптимизации скорости работы вашего сайта. — Текст: электронный // habr.com: [сайт]. — Режим доступа: <https://habr.com/ru/articles/112720/> (дата обращения: 15.04.2024).

19 Диаграмма IDEF0 второго уровня. — Текст: электронный // megaobuchalka.ru: [сайт]. — Режим доступа: <https://megaobuchalka.ru/8/26772.html> (дата обращения: 31.03.2024).

20 Как дизайн влияет на продажи. — Текст: электронный // dzen.ru: [сайт]. — Режим доступа: <https://dzen.ru/a/ZGDiXFEDcnnU9KaC> (дата обращения: 22.03.2024)

21 Обзор фреймворка React.js: преимущества, недостатки и сценарии использования — Текст: электронный // serverspace.ru: [сайт]. — Режим доступа: <https://serverspace.ru/about/blog/obzor-frejmvorka-react-js/> (дата обращения: 23.03.2024)

22 ОПТИМИЗАЦИЯ ИЗОБРАЖЕНИЙ ДЛЯ САЙТА. — Текст: электронный // ashmanov.com: [сайт]. — Режим доступа: <https://www.ashmanov.com/education/articles/optimizatsiya-izobrazhenij-dlya-sajta/> (дата обращения: 26.03.2024)

23 ОСОБЕННОСТИ, ХАРАКТЕРИСТИКИ И ОБЛАСТИ ПРИМЕНЕНИЯ NODE.JS — Текст: электронный // scand.com: [сайт]. — Режим доступа: <https://scand.com/ru/company/blog/node-js-features-uses-and-benefits-of-development/> (дата обращения: 25.03.2024)

24 Правильная реализация перехода по ссылке в рамках одного сайта. — Текст: электронный // evergreens.com: [сайт]. — Режим доступа: <https://ru.stackoverflow.com/questions/740227/> (дата обращения: 10.04.2024).

25 Руководство по JavaScript. — Текст: электронный // metanit.com: [сайт]. — Режим доступа: <https://metanit.com/web/javascript/> (дата обращения: 22.03.2024).

26 Хранение картинок в БД: за и против? — Текст: электронный // habr.com: [сайт]. — Режим доступа: <https://qna.habr.com/q/21342> (дата обращения: 11.04.2024)

