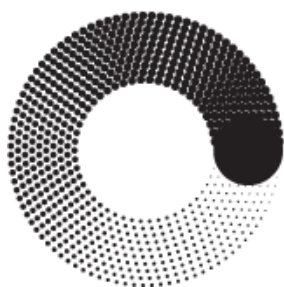


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**МОСКОВСКИЙ
ПОЛИТЕХ**

Пояснительная записка к курсовому проекту

по дисциплине «Разработка веб-приложений»

Тема: «Разработка веб-приложения для учета товаров в магазине техники»

Группа

221-329

Студент

Тютичкин Семен Владимирович

Дата

15.05.2024

Преподаватель

Кружалов Алексей Сергеевич

2024 г

СОДЕРЖАНИЕ

ЗАДАНИЕ	3
ВВЕДЕНИЕ	3
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1 Обзор существующих программных продуктов по теме работы	5
1.2 Анализ программных инструментов разработки веб-приложений	7
1.3 Формулировка цели и задач работы	10
2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ	11
2.1 Анализ целевой аудитории	11
2.2 Описание функциональности приложения	11
2.2.1 Диаграмма вариантов использования	11
2.2.2 User Stories	12
2.3 Проектирование модели данных	13
2.3.1 Сущности.....	13
2.3.2 Атрибуты.....	13
2.3.3 Физическая модель базы данных.....	13
2.4 Разработка макетов страниц.....	13
3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ	20
3.1 Выбор языка	20
3.2 Функционал веб-приложения	20
3.2.1 Поиск товаров	20
3.2.2 Добавление товара.....	21
3.2.3 Удаление товара	23
3.2.4 Изменить товар	24

ЗАДАНИЕ

на выполнение курсовой работы (проекта)

Тютичкину Семену Владимировичу,
(ФИО обучающегося)

обучающемуся группы 221-329,

направления подготовки 09.03.01 «Информатика и вычислительная техника»
по дисциплине «Разработка веб-приложений»

на тему «Разработка веб-приложения для учета товаров в магазине техники»

1. Исходные данные к работе (проекту): информационные ресурсы в сети интернет, научные публикации в открытой печати.

2. Содержание задания по курсовой работе (проекту) – перечень вопросов, подлежащих разработке:

Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения	Примечание
Раздел 1. Анализ предметной области	10%	30.03.2024	
Задача 1.1. Обзор существующих программных продуктов по теме работы	2%	24.03.2024	
Задача 1.2. Анализ программных инструментов разработки веб-приложений	5%	27.03.2024	
Задача 1.3. Формулировка цели и задач работы	3%	30.03.2024	
Раздел 2. Проектирование веб-приложения	25%	18.04.2024	
Задача 2.1. Анализ целевой аудитории	5%	03.04.2024	
Задача 2.2. Описание функциональности приложения (диаграмма вариантов использования, user story и т. д.)	10%	07.04.2024	
Задача 2.3. Проектирование модели данных (ER-диаграмма, логическая и физическая схемы БД)	5%	16.04.2024	
Задача 2.4. Разработка макетов страниц (Wireframe)	5%	17.04.2024	
Раздел 3. Разработка веб-приложения	50%	10.05.2024	
Задача 3.1. Разработка базовой структуры приложения и верстка шаблонов страниц	10%	22.04.2024	
Задача 3.2. Реализация аутентификации пользователей	10%	24.04.2024	
Задача 3.3. Реализация CRUD-интерфейса для взаимодействия с товарами	15%	27.04.2024	
Задача 3.4. Реализация фильтрации данных	5%	30.04.2024	
Задача 3.5. Реализация импорта и экспорта данных в формате JSON	10%	03.05.2024	
Раздел 4. Оформление итогов работы	15%	05.05.2024	
Задача 4.1. Создание Git-репозитория с кодом проекта	3%	1.06.2024	
Задача 4.2. Деплой приложения на хостинг	7%	01.05.2024	
Задача 4.3. Оформление отчёта о проделанной работе	5%	27.05.2024	

Руководитель курсовой работы (проекта): преподаватель кафедры «Инфокогнитивные технологии»

«13» март 2024 г.

Дата выдачи задания

Дата сдачи выполненной работы (проекта)

Задание принял к исполнению

«15» март 2024 г.

(подпись)

А. С. Кружалов

«13» март 2024 г.

«17» июнь 2024 г.

С. В. Тютичкин

(И. О. Фамилия)

ВВЕДЕНИЕ

В современном информационном обществе эффективное управление товарными запасами является ключевым фактором успешной работы в различных сферах бизнеса. С ростом объемов продаж и увеличением ассортимента товаров необходимо иметь эффективные инструменты для учета, отслеживания и управления запасами. Однако, многие существующие решения не всегда полностью удовлетворяют потребности пользователей и не обеспечивают достаточного уровня удобства и функциональности.

В рамках данного курсового проекта ставится цель разработки веб-приложения для учета товаров в магазине техники. Основными задачами проекта являются создание удобного и интуитивно понятного интерфейса для добавления, редактирования, отслеживания запасов и списания товаров.

Актуальность проблемы подтверждается повседневной необходимостью эффективного управления товарными запасами как в розничной, так и в оптовой торговле. Множество существующих продуктов на рынке поддерживают базовый функционал учета товаров, однако существует потребность в более гибких и персонализированных решениях, способных адаптироваться под конкретные потребности и особенности работы магазинов техники.

В ходе работы будут рассмотрены существующие подходы и программные продукты в области управления товарными запасами; проанализированы требования пользователей и, в итоге, будет разработано веб-приложение, представляющее собой современное эффективное решение для учета товаров в магазинах техники.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор существующих программных продуктов по теме работы

Для проведения анализа необходимо учитывать следующие критерии:

- **Функциональность:** Оценка доступных функций управления товарными запасами, таких как добавление и редактирование товаров, отслеживание запасов и списание товаров.
- **Удобство использования:** Оценка интерфейса пользователя, его интуитивной понятности, удобства навигации и доступности основных функций.
- **Поддержка платформ:** исследование возможности использования продуктов на различных платформах (веб, мобильные устройства, десктопные приложения) и их совместимость с различными операционными системами.

Рассмотрим несколько существующих на рынке веб-приложений для учета товаров, используя эти критерии:

Zoho Inventory

- **Функциональность:** Zoho Inventory предоставляет полный набор инструментов для управления товарными запасами, включая добавление и редактирование товаров, отслеживание запасов в реальном времени, генерацию отчетов и интеграцию с различными платформами электронной коммерции.
- **Удобство использования:** Интерфейс Zoho Inventory интуитивно понятен, с удобной навигацией и доступом к основным функциям через главную панель управления.
- **Поддержка платформ:** Приложение доступно в веб-версии, а также имеет мобильные приложения для iOS и Android.

МойСклад

- **Функциональность:** МойСклад предлагает полный набор инструментов для управления запасами, включая добавление и редактирование товаров, отслеживание запасов, обработку заказов, интеграцию с системами продаж и генерацию отчетов.

- Удобство использования: Интерфейс МойСклад интуитивно понятен и удобен для навигации, с доступом ко всем основным функциям через главную панель управления.
- Поддержка платформ: МойСклад доступен как веб-приложение и имеет мобильные приложения для iOS и Android.

1С:Управление торговлей

- Функциональность: 1С:Управление торговлей предоставляет обширный набор инструментов для управления запасами, включая добавление и редактирование товаров, управление складскими операциями, интеграцию с системами бухгалтерского учета и аналитики.
- Удобство использования: Интерфейс 1С достаточно сложный, но мощный, с возможностью настройки под конкретные нужды пользователя.
- Поддержка платформ: 1С:Управление торговлей доступен как десктопное приложение и имеет веб-версию.

Исходя из проведенного анализа существующих веб-приложений для учета товаров, можно сделать вывод о том, что на рынке уже существует широкий выбор инструментов с различными функциональными возможностями и уровнями удобства использования. Эти приложения предоставляют базовый и расширенный набор функций, позволяющий пользователям эффективно управлять товарными запасами.

Однако, в процессе анализа также становится очевидным, что каждое из существующих решений имеет свои особенности и ограничения, которые могут не полностью соответствовать потребностям конкретного магазина техники. Некоторые приложения могут обладать более широким набором функций, в то время как другие могут предлагать более удобный интерфейс или интеграцию с другими сервисами.

1.2 Анализ программных инструментов разработки веб-приложений

Рассмотрим несколько инструментов, через которые можно реализовать веб-приложение:

HTML (HyperText Markup Language)

- Описание: HTML является стандартным языком разметки для создания веб-страниц. Он определяет структуру содержимого веб-страницы с помощью различных элементов и тегов.
- Преимущества: простота использования, широкая поддержка, возможность создания доступных и понятных структур веб-страниц.
- Недостатки: ограниченные возможности в создании динамических элементов, необходимость комбинировать с другими технологиями для создания полноценных веб-приложений.

CSS (Cascading Style Sheets)

- Описание: CSS используется для оформления и стилизации веб-страниц, определяя внешний вид элементов, их расположение, цвета, шрифты и т.д.
- Преимущества: возможность создания красивого и современного дизайна веб-страниц, легкая поддержка и изменение стилей, улучшение доступности и пользовательского опыта.
- Недостатки: некоторые сложности в организации стилей для крупных проектов, возможные конфликты и переопределения стилей.

JavaScript (JS)

- Описание: JavaScript — это язык программирования, который используется для добавления интерактивности и динамических возможностей на веб-страницах.
- Преимущества: широкие возможности в создании интерактивных элементов, обработке событий, анимации, валидации форм и других функций.

- Недостатки: возможность возникновения ошибок и проблем совместимости между браузерами, увеличение сложности кода при разработке крупных проектов.

Python

- Описание: Python — высокоуровневый язык программирования с акцентом на читаемость кода. Он широко используется для разработки веб-приложений, научных вычислений, автоматизации и других целей.

- Преимущества: простота и понятность синтаксиса, богатая стандартная библиотека, широкие возможности в разработке веб-приложений с использованием фреймворков.

- Недостатки: некоторые аспекты производительности могут быть не такими эффективными, как в некоторых других языках программирования, некоторые библиотеки могут быть менее развитыми или не поддерживаются.

Node.js

- Описание: Node — программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

- Преимущества: Node.js позволяет использовать JavaScript как на клиентской, так и на серверной стороне, обеспечивая единый язык программирования для всего приложения. Node.js обеспечивает высокую производительность благодаря асинхронному, неблокирующему вводу/выводу.

- Недостатки: Node.js может столкнуться с проблемами стабильности при работе с большими объемами данных или высокой нагрузкой.

Java

- Описание: Java — это объектно-ориентированный язык программирования, широко используемый для разработки корпоративных приложений и веб-приложений.

- Преимущества: высокая производительность, кросс-платформенность, богатая стандартная библиотека, большой выбор фреймворков для разработки веб-приложений (например, Spring).

- Недостатки: более сложный синтаксис по сравнению с некоторыми современными языками, более длительное время компиляции, относительно высокая требовательность к ресурсам.

Golang (Go)

- Описание: Go — это компилируемый язык программирования, разработанный компанией Google. Он предназначен для создания высокопроизводительных и масштабируемых веб-приложений.

- Преимущества: высокая производительность, простота синтаксиса, автоматическое управление памятью, встроенная поддержка параллелизма.

- Недостатки: ограниченная стандартная библиотека по сравнению с более зрелыми языками, более сложное управление пакетами и зависимостями.

Каждый из этих инструментов имеет свои преимущества и недостатки, и выбор зависит от конкретных требований проекта, потребностей разработчика и контекста использования. В контексте разработки веб-приложения для учета товаров в магазине техники, наилучшим выбором может стать язык программирования Go. Его высокая производительность и простота синтаксиса позволяют эффективно управлять ресурсами и обеспечивают отличную масштабируемость, что особенно важно для приложения с высокой нагрузкой и требующего быстрой обработки данных. Встроенная поддержка параллелизма в Go поможет оптимизировать работу с большими объемами данных и повысить общую производительность системы.

1.3 Формулировка цели и задач работы

Основная цель курсового проекта "Разработка веб-приложения для учета товаров в магазине техники" заключается в создании эффективного и удобного инструмента для управления товарами, который будет удовлетворять потребности пользователя в инвентаризации, отслеживании и управлении складскими запасами магазина техники.

Задачи проекта:

1. **Анализ предметной области.**
2. **Проектирование веб-приложения.**
3. **Разработка веб-приложения.**
4. **Тестирование и отладка.**
5. **Оформление итогов работы.**

2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ

2.1 Анализ целевой аудитории

Целевая аудитория проекта может включать следующие группы:

1. Интернет-магазины техники. Владельцы и администраторы онлайн-магазинов, которым необходимо отслеживать наличие товаров в реальном времени, управлять заказами и инвентаризацией.
2. Крупные торговые сети и дистрибьюторы техники. Логистические отделы и менеджеры по закупкам, которые требуют надежных решений для управления большими объемами данных о товарах и запасах.
3. Складские и логистические компании. Компании, предоставляющие услуги складирования и логистики для магазинов техники, которым важно иметь точную информацию о поступлениях и отгрузках товаров.
4. Технические специалисты и IT-компании. Разработчики программного обеспечения и системные интеграторы, которые занимаются внедрением и поддержкой учетных систем для магазинов и складов техники.

2.2 Описание функциональности приложения

2.2.1 Диаграмма вариантов использования

Диаграмма вариантов использования поможет визуализировать основные функции веб-приложения для учета товаров в магазине техники и взаимодействие между пользователями и системой.

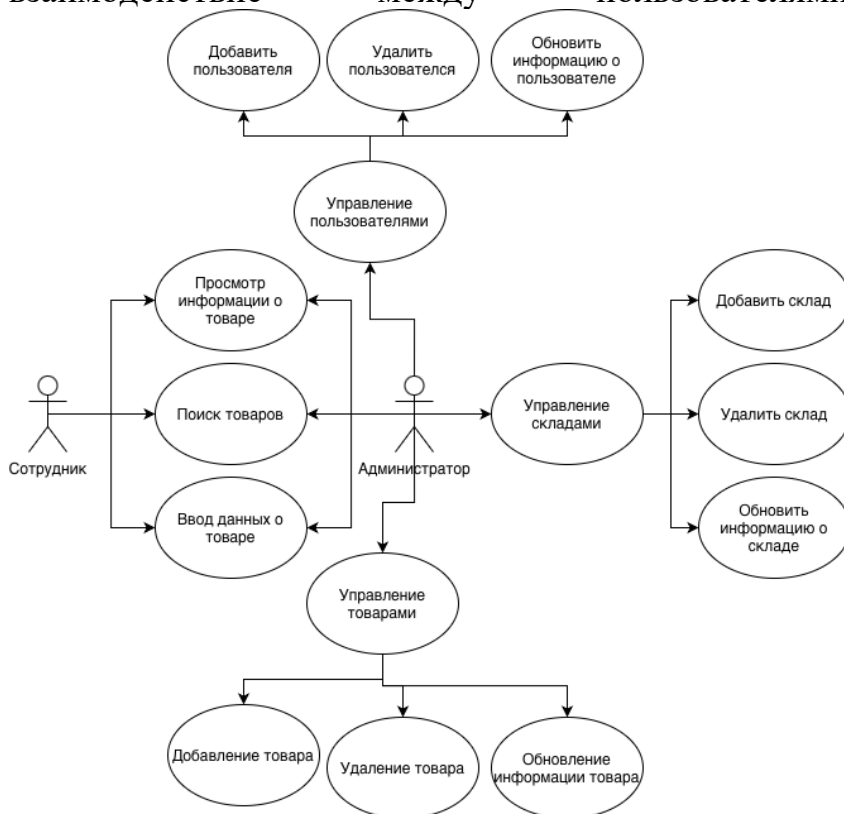


Рисунок 2.2.2.1 - Диаграмма вариантов использования 2.2.1.1

2.2.2 User Stories

1. Управление товарами

Как администратор, я хочу иметь возможность добавлять новые товары, редактировать существующие и удалять старые, чтобы поддерживать актуальность товарного ассортимента.

Критерии:

- Администратор может добавлять новую информацию о товаре, включая название, описание, цену и количество на складе.
- Администратор может импортировать и экспортировать товары с помощью файлов формата JSON
- Администратор может редактировать информацию о существующих товарах.
- Администратор может удалять товары из системы.

2. Ввод данных о товарах

Как сотрудник, я хочу иметь возможность вводить информацию о новых поступлениях товаров, чтобы актуализировать складские запасы.

Критерии:

- Сотрудник может вводить количество поступивших товаров.
- Система автоматически обновляет информацию о количестве товаров на складе.

3. Просмотр товаров

Как сотрудник, я хочу иметь возможность просматривать список всех товаров, чтобы иметь представление о текущем состоянии складских запасов.

Критерии:

- Сотрудник может просматривать информацию обо всех товарах в виде списка.
- Список товаров можно сортировать и фильтровать по различным критериям (кол-во товара на складе, цена).

4. Поиск товаров

Как сотрудник, я хочу иметь возможность искать товары по названию или другим параметрам, чтобы быстро находить нужные позиции.

Критерии:

- Сотрудник может использовать строку поиска для ввода названия товара или параметров (цена, кол-во).
- Система выводит результаты поиска в виде списка товаров, соответствующих заданным критериям.

5. Управление пользователями

Как администратор, я хочу иметь возможность добавлять новых сотрудников в систему.

Критерии:

- Администратор может создавать учетные записи для новых пользователей.
- Администратор может удалять учетные записи пользователей.

2.3 Проектирование модели данных

2.3.1 Сущности

В данной работе были выделены следующие сущности: пользователи и товары.

2.3.2 Атрибуты

Атрибуты, описанных выше сущностей, можно рассмотреть на рисунке 2.3.2.1.



Рисунок 2.3.2.1 - Диаграмма базы данных 2.3.2.1

2.3.3 Физическая модель базы данных

Физическую модель базы данных можно рассмотреть на рисунке 2.3.3.1. В качестве базы данных используется SQLite. Т.к. в SQLite отсутствует булевый тип, вместо него используется тип *integer*.

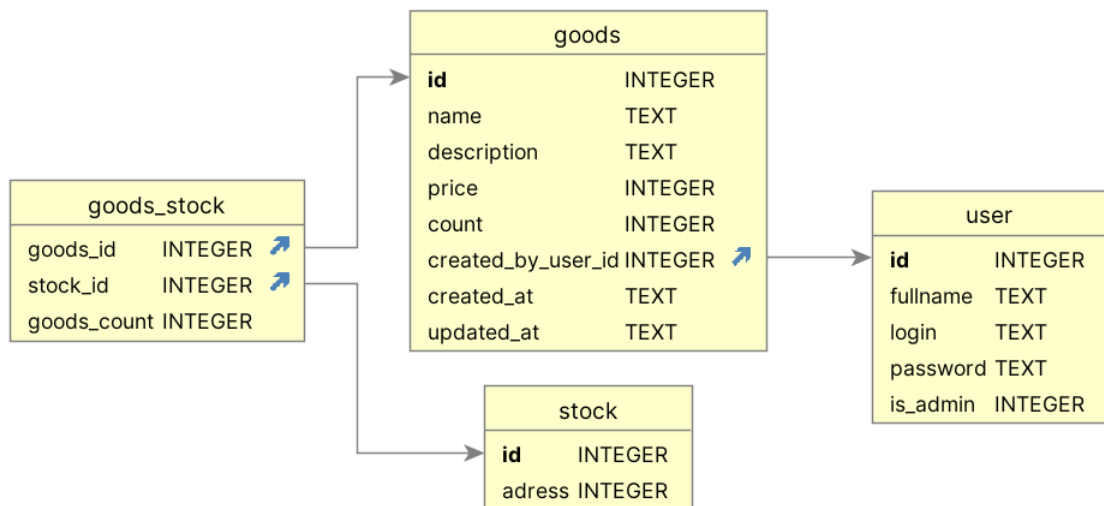


Рисунок 2.3.3.1 - Физическая модель базы данных 2.3.3.1

2.4 Разработка макетов страниц

Макет страницы авторизации представлен на рисунке 2.4.1

Login

Login

Password

Password

Frame

LOG IN

Рисунок 2.4.1 – Страница авторизации

Макет основной страницы сотрудника представлен на рисунке 2.4.2

ОСНОВНАЯ СТРАНИЦА

Кнопка сортировки

Кнопка редактирования

01 Input Field

01 Input Field

min Цена Placeholder

min Кол-во Placeholder

01 Input Field

01 Input Field

max Цена Placeholder

max Кол-во Placeholder

Frame

Фильтр по названию товара

ПОИСК

ID	Название	Описание	Номер склада	Кем добавлен	Цена	Кол-во	
1	name	desc			123	5	

Рисунок 2.4.2 – Основная страница сотрудника

Макет окна редактирования для сотрудника представлен на рисунке 2.4.3

Название товара

Кол-во товара

0 1 Input Field

Placeholder

0 0 Button

Применить

Рисунок 2.4.3 – Окно редактирования товара для сотрудника

Макет страница с товарами для администратора представлен на рисунке 2.4.4

СТРАНИЦА С ТОВАРАМИ

00 Button 00 Button 00 Button delete-button 1

Товары Пользователи Склады

01 Input Field 01 Input Field 01 Input Field 01 Input Field

min Цена Placeholder min Кол-во Placeholder

max Цена Placeholder max Кол-во Placeholder

Добавить новый товар

Добавить товар на склад

Изменить кол-во товара на складе

Импортировать товар из файла JSON

Фильтр по названию товара

поиск

ID	Название	Описание	Номер склада	Кем добавлен	Цена	Кол-во	
1	name	desc			123	5	✎ 🗑️ 📄

Кнопка редактирования

Экспорт товара в формате JSON

Кнопка удаления

Кнопка сортировки

Рисунок 2.4.4 – Страница товаров для администратора

Макет страницы с пользователями для администратора представлен на рисунке 2.4.5

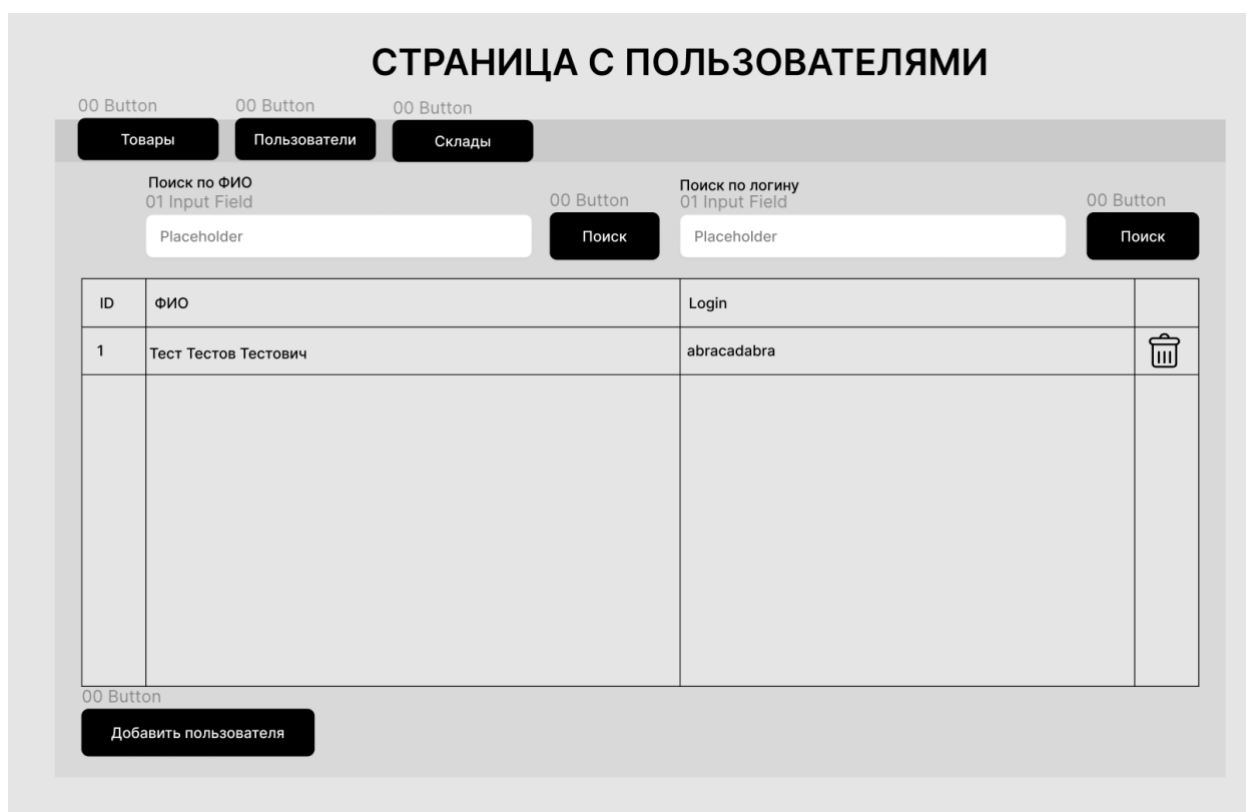


Рисунок 2.4.5 – Страница с пользователями для администратора

Макет окна редактирования для администратора представлен на рисунке 2.4.6

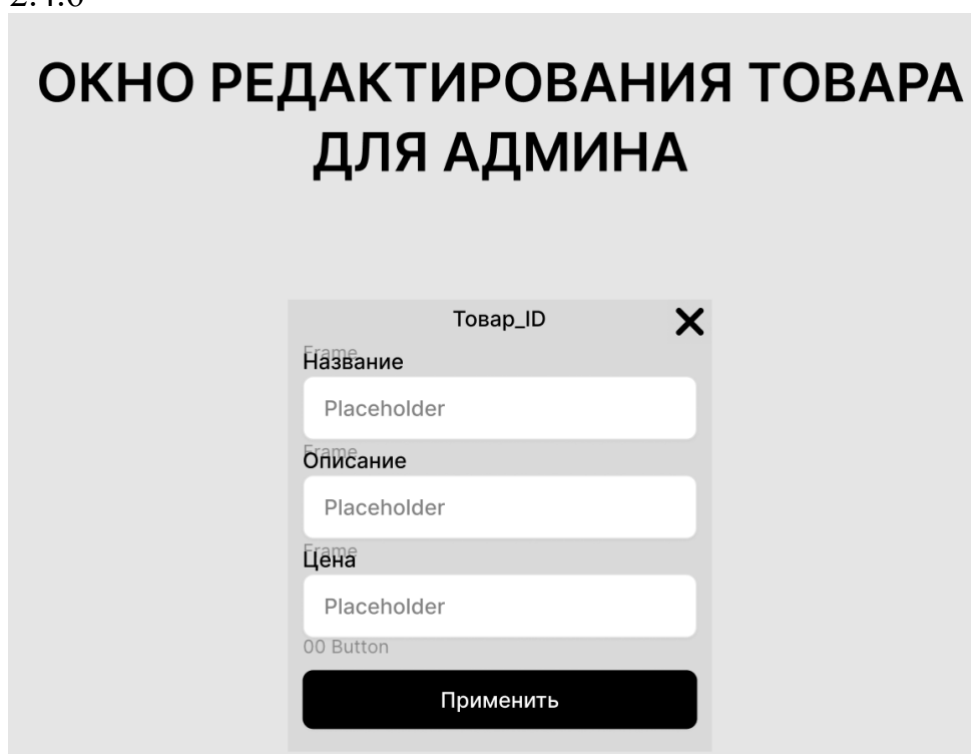


Рисунок 2.4.6 – Окно редактирования для администратора

Макет окна добавления товара для администратора представлен на рисунке 2.4.7.

Рисунок 2.4.7 – Окно добавления товара для администратора

Макет окна добавления товара на склад представлен на рисунке 2.4.8

Рисунок 2.4.8 – Окно добавления товара на склад для администратора

Макет окна редактирования кол-ва товара на складе представлен на рисунке 2.4.9

The mockup shows a modal window with a close button (X) in the top right corner. It contains three input fields: 'ID_товара' with a placeholder and a downward arrow, 'ID_склада' with a placeholder and a downward arrow, and 'Кол-во' with a placeholder. An arrow labeled 'РАСКРЫТЬ СПИСОК' points to the downward arrow of the 'ID_товара' field. At the bottom is a black button labeled 'Изменить'.

Рисунок 2.4.9 – Окно редактирования кол-ва товара на складе для администратора

Макет страницы с складами для администратора представлен на рисунке 2.4.10.

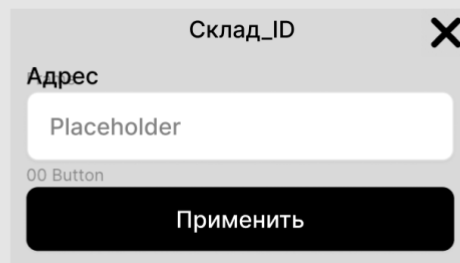
The mockup is titled 'СТРАНИЦА С СКЛАДАМИ'. It features three tabs: 'Товары', 'Пользователи', and 'Склады', with 'Склады' being the active tab. Below the tabs is a search section with the text 'Поиск по адресу', an input field with a placeholder, and a 'Поиск' button. A table with three columns is shown: 'ID', 'Адрес', and an action column. The first row contains the value '1' in the ID column and 'Тест Тестов Тестович' in the address column. The action column contains edit and delete icons. At the bottom left is a 'Добавить склад' button.

ID	Адрес	
1	Тест Тестов Тестович	

Рисунок 2.4.10 – Страница с складами для администратора

Макеты окна редактирования и добавления для администратора представлены на рисунках 2.4.11 - 2.4.12.

ОКНО РЕДАКТИРОВАНИЯ ДЛЯ АДМИНА



Склад_ID

Адрес

Placeholder

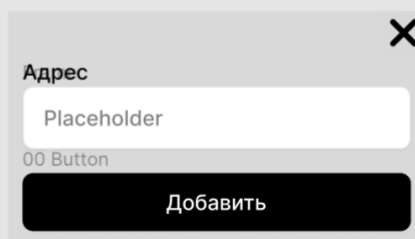
00 Button

Применить

This is a screenshot of a software window titled 'Склад_ID' with a close button (X) in the top right corner. Inside the window, there is a label 'Адрес' above a text input field containing the placeholder text 'Placeholder'. Below the input field, the text '00 Button' is visible. At the bottom of the window is a large black button with the white text 'Применить'.

Рисунок 2.4.11 – Окно добавления для администратора

ОКНО ДОБАВЛЕНИЯ ДЛЯ АДМИНА



Адрес

Placeholder

00 Button

Добавить

This is a screenshot of a software window titled 'Адрес' with a close button (X) in the top right corner. Inside the window, there is a label 'Адрес' above a text input field containing the placeholder text 'Placeholder'. Below the input field, the text '00 Button' is visible. At the bottom of the window is a large black button with the white text 'Добавить'.

Рисунок 2.4.12 – Окно добавления для администратора

3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

3.1 Выбор языка

В качестве инструмента для разработки веб-приложения был выбран язык Golang с использованием фреймворка Gin. Для frontend части были использованы следующие языки: «JS», «CSS» и «HTML».

3.2 Функционал веб-приложения

3.2.1 Поиск товаров

В данном веб-приложении реализован поиск товаров по следующим фильтрам: «минимальная цена», «максимальная цена», «минимальное кол-во», «максимальное кол-во» и «название товара». Фильтрация происходит по нескольким параметрам одновременно. Также присутствует возможность отсортировать значения по цене и кол-во. Цена находится в приоритете. По умолчанию выставлена фильтрация по возрастанию.

Листинг 1 – Сервисная часть поиска товаров

```
9 func (s *Service) SearchGoods(c *gin.Context) { 1 usage
10     var (
11         searchGoodRequest models.SearchGoodRequest
12     )
13     if err := c.BindJSON(&searchGoodRequest); err != nil {
14         c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
15         return
16     }
17     goods, err := s.repo.SearchGoods(c, searchGoodRequest)
18     if err != nil {
19         c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
20         return
21     }
22     c.JSON(http.StatusOK, gin.H{"data": goods})
23 }
24
```

Листинг 2 – Репозиторий поиска товаров (часть 1)

```

func (r *Repository) SearchGoods(ctx context.Context, searchRequest models.SearchGoodRequest) (goods []models.Good, err error) {
    var query = `
        SELECT g.id,
               name,
               description,
               price,
               coalesce(u.fullname, ''),
               coalesce(gs.stock_id, 0),
               coalesce(gs.goods_count, 0)
        FROM main.goods g
        LEFT JOIN main.user u ON u.id = g.created_by_user_id
        LEFT JOIN main.goods_stock gs on gs.goods_id = g.id
        WHERE ($1 = '' OR (name like concat_ws('%', $1, '%')))
        AND ($2 = 0 OR (price >= $2))
        AND ($3 = 0 OR (price <= $3))
        AND ($4 = 0 OR (gs.goods_count >= $4))
        AND ($5 = 0 OR (gs.goods_count <= $5))
        AND ($6 = 0 OR (g.id = $6))

    `

    orderPriceStr := ""
    orderCountStr := ""
    if searchRequest.IsCountDesc == true {
        orderCountStr = "DESC"
    }
    if searchRequest.IsPriceDesc == true {
        orderPriceStr = "DESC"
    }

    query = fmt.Sprintf(format: "%v\nORDER BY price %v, gs.goods_count %v", query, orderPriceStr, orderCountStr)
}

```

Листинг 3 – Репозиторий поиска товаров (часть 2)

```

36     query = fmt.Sprintf(format: "%v\nORDER BY price %v, gs.goods_count %v", query, orderPriceStr, orderCountStr)
37     rows, err := r.db.Query(query,
38         searchRequest.Name,
39         searchRequest.MinPrice,
40         searchRequest.MaxPrice,
41         searchRequest.MinCount,
42         searchRequest.MaxCount,
43         searchRequest.ID,
44     )
45     if err != nil {
46         return goods: nil, err
47     }
48
49     for rows.Next() {
50         var good models.Good
51         err := rows.Scan(
52             &good.ID,
53             &good.Name,
54             &good.Description,
55             &good.Price,
56             &good.CreatedByUserFullName,
57             &good.StockID,
58             &good.Count,
59         )
60         if err != nil {
61             return goods: nil, err
62         }
63         goods = append(goods, good)
64     }
65     return goods, err: nil
66 }

```

3.2.2 Добавление товара

При использовании ручки “добавления товара” из контекста берется информация “является ли пользователь администратором”, если пользователь

таковым не является, то ему будет отказано в использовании этой ручки. Также из контекста берется информация о логине пользователя, которая дальше используется для заполнения информации “кем был добавлен товар”.

Листинг 4 – Сервисный метод добавления товара

```
9  func (s *Service) AddGood(c *gin.Context) { 1 usage
10      isAdmin, _ := c.Get(key: "isAdmin")
11      if isAdmin == false {
12          c.JSON(http.StatusForbidden, gin.H{"error": "Access is denied"})
13          return
14      }
15      var (
16          good models.Good
17      )
18      if err := c.BindJSON(&good); err != nil {
19          c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
20          return
21      }
22
23      err := s.setCreatedByUserID(c, &good)
24      if err != nil {
25          c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
26      }
27
28      err = s.repo.AddGood(c, good)
29      if err != nil {
30          switch {
31          default: c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
32          }
33      }
34      return
35  }
36      c.Status(http.StatusOK)
37  }
38
39  func (s *Service) setCreatedByUserID(c *gin.Context, good *models.Good) (err error) { 1 usage
40      login, _ := c.Get(key: "login")
41      user, err := s.repo.GetUserByLogin(c, login.(string))
42      if err != nil {
43          return err
```

Листинг 5 – Метод репозитория “добавить товар”

```
9 func (r *Repository) AddGood(ctx context.Context, good models.Good) (err error) { 2 usages
10     var query = `
11         INSERT INTO main.goods(name, description, price, created_by_user_id, created_at, updated_at)
12         VALUES ($1, $2, $3, $4, $5, $6)
13     `
14     _, err = r.db.Exec(query,
15         good.Name,
16         good.Description,
17         good.Price,
18         good.CreatedByUserID,
19         time.Now(),
20         time.Now(),
21     )
22     if err != nil { return err }
25     return err, nil
26 }
```

Листинг 6 – Метод репозитория “получить пользователя по логину”

```
8 func (r *Repository) GetUserByLogin(ctx context.Context, login string) (user models.User, err error) {
9     var query = `
10         SELECT id,
11             fullname,
12             login
13         FROM main.user
14         WHERE login = $1;
15     `
16     err = r.db.QueryRow(query, login).Scan(
17         &user.ID,
18         &user.FullName,
19         &user.Login)
20     if err != nil {
21         return models.User{}, err
22     }
23     return user, err, nil
24 }
25
```

3.2.3 Удаление товара

При удалении товара также проверяется наличие флага “является ли пользователь администратором”, если он таковым не является, то ему будет отказано в доступе.

Листинг 7 – Метод сервиса “удалить товар”

```
10 func (s *Service) DeleteGood(c *gin.Context) { 1 usage
11     isAdmin, _ := c.Get(key: "isAdmin")
12     if isAdmin == false {
13         c.JSON(http.StatusForbidden, gin.H{"error": "Access is denied"})
14         return
15     }
16     id, err := strconv.Atoi(c.Param(key: "id"))
17     > if err != nil { c.JSON(http.StatusBadRequest, gin.H{"error": "Incorrect id"}) }
20     err = s.repo.DeleteGoodByID(c, models.Good{ID: id})
21     if err != nil {
22         switch {
23         default:
24             c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
25         }
26         return
27     }
28     c.Status(http.StatusOK)
29 }
```

Листинг 8 – Метод репозитория “удалить товар”

```
8 ⑧ func (r *Repository) DeleteGoodByID(ctx context.Context, good models.Good) (err error) { 1 usag
9     var query = `
10         DELETE FROM main.goods
11         WHERE id = $1;
12
13         DELETE FROM main.goods_stock
14         WHERE goods_id = $1;
15     `
16     _, err = r.db.Exec(query, good.ID)
17     > if err != nil { return err }
20     return err: nil
21 }
```

3.2.4 Изменить товар

При изменении товара также, как и в пунктах 3.2.2-3.2.3, выполняется проверка роли пользователя на администратора.

Листинг 9 – Метод сервис “изменить товар”

```
9 func (s *Service) EditGood(c *gin.Context) { 1 usage
10     isAdmin, _ := c.Get(key: "isAdmin")
11     if isAdmin == false {
12         c.JSON(http.StatusForbidden, gin.H{"error": "Access is denied"})
13         return
14     }
15     var (
16         good models.Good
17     )
18     if err := c.BindJSON(&good); err != nil {
19         c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
20         return
21     }
22     err := s.repo.EditGood(c, good)
23     if err != nil {
24         switch {
25         default: c.JSON(http.StatusInternalServerError, gin.H{"error": err.Error()})
26         }
27     }
28     return
29 }
30 c.Status(http.StatusOK)
31 }
```

Листинг 10 – Метод репозитория “изменить товар”

```
8 func (r *Repository) EditGood(ctx context.Context, good models.Good) (err error) { 1 usage
9     var query = `
10         UPDATE main.goods SET
11             name = ?,
12             description = ?,
13             price = ?
14         WHERE id = ?;
15     `
16     _, err = r.db.Exec(query, good.Name, good.Description, good.Price, good.ID)
17     if err != nil { return err }
18     return err: nil
19 }
20
21 }
```

3.1 Финальный проект

Исходные файлы проекта находятся по ссылке:
https://github.com/Tyutichkin/tyutichkin_web_cur_2024

ЗАКЛЮЧЕНИЕ

Разработка веб-приложения для учета товаров в магазине техники является важной задачей, обусловленной необходимостью современного и эффективного управления товарными запасами. В ходе данного проекта были проведены всесторонний анализ предметной области и существующих решений, что позволило определить основные потребности и требования пользователей.

Созданное веб-приложение предоставляет удобный и интуитивно понятный интерфейс для ввода, редактирования и поиска товаров, а также функциональные возможности для управления заказами и пользователями. Реализация приложения обеспечивает гибкость и адаптируемость системы к различным условиям работы, что позволяет оптимизировать складские процессы и улучшить общую эффективность управления запасами.

Таким образом, разработанное веб-приложение не только удовлетворяет текущие потребности пользователей, но и закладывает основу для дальнейшего расширения, что делает его важным шагом в направлении модернизации управления товарными запасами в магазинах техники.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Документация Go: <https://go.dev/doc/>
2. Документация SQLite: <https://www.sqlite.org/docs.html>
3. Документация JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
4. Stack Overflow. (n.d.). <https://stackoverflow.com/>
5. Курс «Основы веб-приложений (СИПИ)»: <https://c1447.c.3072.ru/course/view.php?id=13080>
6. Курс «Разработка веб-приложений (СИПИ)»: <https://c1447.c.3072.ru/course/view.php?id=13940>