Государственное бюджетное профессиональное образовательное учреждение

Ростовской области

|  |
| --- |
| «Таганрогский колледж морского приборостроения» |

|  |
| --- |
| «Магазин по продаже спортивных товаров» |
|  |
| Курсовой проект |
| Пояснительная записка |
| ТКМП.МДК.09.02.07.17 ПЗ |

Проверил Выполнил

Руководитель Студент

2024

**Содержание**

[Сокращение 3](#_Toc138779628)

[Аннотация 4](#_Toc138779629)

[Введение 5](#_Toc138779630)

[1. Общая часть 6](#_Toc138779631)

[1.1. Постановка задачи 6](#_Toc138779632)

[1.2. Анализ и исследование задачи, построение модели системы 8](#_Toc138779633)

[1.3. Разработка архитектуры решения 10](#_Toc138779634)

[1.4 Обоснование и выбор средств разработки решения 12](#_Toc138779635)

[1.5. Формализация расчетов 13](#_Toc138779636)

[1.5.1. Проектирование базы данных 14](#_Toc138779637)

[1.5.2. Проектирование интерфейса 17](#_Toc138779638)

[2. Описание программы 18](#_Toc138779639)

[3. Тестирование программы 20](#_Toc138779640)

[4 Программирование 22](#_Toc138779641)

[5 Отладка, оптимизация и тестирование программы 25](#_Toc138779642)

[Заключение 28](#_Toc138779643)

[Список используемых источников 29](#_Toc138779644)

[Приложение А. Результаты выполнения программы 30](#_Toc138779645)

**Сокращение**

БД – База данных

ИС - Информационная система

ПК – Персональный компьютер

ПО – Программное обеспечение

**Аннотация**

Курсовой проект представляет собой разработку информационной системы для магазина спортивных товаров при помощи Flask, Python. Система будет включать в себя функционал для онлайн-покупок и управления товарными запасами. Пользователи смогут осуществлять заказ товаров, отслеживать их наличие и получать информацию о статусе своих покупок. Для реализации проекта будут использованы современные технологии веб-разработки, включая Flask для серверной части и Python для написания бизнес-логики. База данных будет использоваться для хранения информации о товарах, заказах и других операциях в магазине спортивных товаров.

# **Введение**

В современном мире, где оперативность и точность играют ключевую роль, эффективная информационная система становится неотъемлемой частью успешного бизнеса. В контексте моего курсового проекта "Магазин по продаже спортивных товаров", я представляю информационную систему, которая поможет оптимизировать процессы управления товарами и повысить эффективность работы магазина.

Цель моего проекта - создать информационную систему, которая позволит автоматизировать работу магазина спортивных товаров. Решение направлено на оптимизацию управления товарными запасами, сокращение времени на выполнение заказов и учет остатков товаров.

Моя информационная система предлагает широкий спектр функциональных возможностей, включая онлайн-покупки, управление товарными запасами, отслеживание статуса заказов и многое другое. Мой подход основан на использовании современных технологий веб-разработки, а также применении языка программирования Python и фреймворка Flask для создания мощной и гибкой системы.

Основная особенность моей информационной системы заключается в её разработке на Python с использованием фреймворка Flask. Python является одним из самых популярных языков программирования, особенно в области веб-разработки, благодаря своей гибкости, простоте и мощности. Использование Python и Flask позволяет мне создавать динамические веб-страницы, взаимодействовать с базами данных и разрабатывать функциональные модули для обработки заказов, управления запасами и многого другого.

1. **Общая часть**
   1. **Постановка задачи**

Целью данного курсового проекта является разработка веб-ресурса под названием «Магазин спортивных товаров» - системы автоматизации работы магазина спортивных товаров, используя язык программирования Python и фреймворк Flask. Веб-ресурс «Магазин спортивных товаров» предназначен для автоматизации работы магазина спортивных товаров.

Для достижения данной цели необходимо решить следующие задачи:

1) разработка пользовательского интерфейса:

Создать интуитивно понятный и удобный пользовательский интерфейс, который позволит пользователям легко взаимодействовать с системой. Интерфейс должен быть отзывчивым и иметь интуитивно понятные элементы управления для выполнения различных задач, таких как выбор товаров, оформление заказа, отслеживание статуса заказа.

2) учет и отслеживание товаров:

Реализовать функционал, который позволит пользователю выбирать товары, покупать товары.

3) отслеживание товарными запасами:

Создать механизм, позволяющий автоматически оповещать пользователя о низких уровнях запасов и предлагать возможности для пополнения запасов. Система должна предлагать оптимальные рекомендации по дозаказу товаров, основываясь на уровне спроса и заданных параметрах запасов.

4) оптимизация распределения товаров:

Реализовать функционал, который поможет оптимизировать распределение товаров по различным зонам склада, учитывая их характеристики, требования к хранению и скорость продажи. Это позволит эффективно использовать пространство склада и снизить затраты на хранение.

* 1. **Анализ и исследование задачи, построение модели системы**

Для начала необходимо проанализировать основные требования к системе «Магазин спортивных товаров». Основной целью данного веб-ресурса должно быть обеспечение удобства и эффективности работы с продукцией.

1) Основные задачи, которые должны быть решены с помощью системы, включают:

- Учет и контроль за поступлением и продажей товаров;

- Мониторинг остатков и движения товаров на складе;

- Поддержание актуальности информации о товарах и их наличии;

- Управление заказами покупателей.

Исследование технологий:

Для создания веб-ресурса системы «Магазин спортивных товаров» могут быть использованы различные языки программирования и фреймворки. Одним из наиболее подходящих вариантов является использование Python с фреймворком Flask. Этот инструмент предоставляет широкие возможности для создания динамических веб-сайтов с использованием баз данных SQL или NoSQL.

Построение модели системы:

Для построения модели системы необходимо определить ее структуру и функциональные возможности. Разработка системы «Магазин спортивных товаров» должна включать следующие компоненты:

- База данных для управления информацией о товарах, покупателях и остатках;

- Интерфейс пользователя для взаимодействия с системой;

- Механизмы обработки заказов покупателей;

- Механизмы мониторинга остатков и движения товаров на складе.

База данных должна содержать информацию о каждом товаре, включая его наименование, количество, цену и наличие на складе. Кроме того, база данных должна содержать информацию о покупателях, их заказах и остатках на складе.

Интерфейс пользователя должен предоставлять возможность просмотра информации о товарах, оформления заказов, а также мониторинга остатков товаров на складе.

Таким образом, создание веб-ресурса системы «Магазин спортивных товаров» с использованием Python и Flask позволит обеспечить эффективное управление товарами на складе, повысить его эффективность и сохранить актуальную информацию о товарах.

## **Разработка архитектуры решения**

Данный раздел пояснительной записки посвящен разработке архитектуры программного решения для интернет-магазина спортивных товаров. В качестве основного инструмента разработки используется веб-фреймворк Flask, а в качестве языка программирования – Python.

Архитектура решения включает в себя следующие компоненты:

1) Веб-интерфейс, реализованный с помощью HTML/CSS/JavaScript. Он предоставляет пользователям возможность просмотра каталога товаров, добавления товаров в каталог, редактирование описания товаров, покупка товаров.

2) Серверная часть, написанная на Python с использованием фреймворка Flask. Она отвечает за обработку запросов от пользователей, взаимодействие с базой данных и генерацию ответов.

3) База данных, которая хранит информацию о товарах, пользователях и заказах. Для работы с базой данных будет использоваться ORM (Object-Relational Mapping), предоставляемая Flask.

4) Система авторизации и аутентификации пользователей.

Структурная диаграмма

На структурной диаграмме изображены основные компоненты архитектуры решения. Здесь можно увидеть взаимосвязь между различными частями системы, включая веб-интерфейс, серверную часть, базу данных и другие модули.

Для определения основных функциональных частей приложения, была разработана структурная схема. Информационная система состоит из 3 подсистем:

1. подсистема работы с сотрудниками:

* регистрация пользователя;
* авторизация пользователя;

1. подсистема добавления товаров в каталог:

* внесение названия товара;
* внесение цены товара;
* внесение описания товара;

1. подсистема редактирования товара из каталога:

* изменение названия товара;
* изменение цены товара;
* изменение описания товара;

Структурная схема приложения, представлена в соответствии с рисунком 1.1.

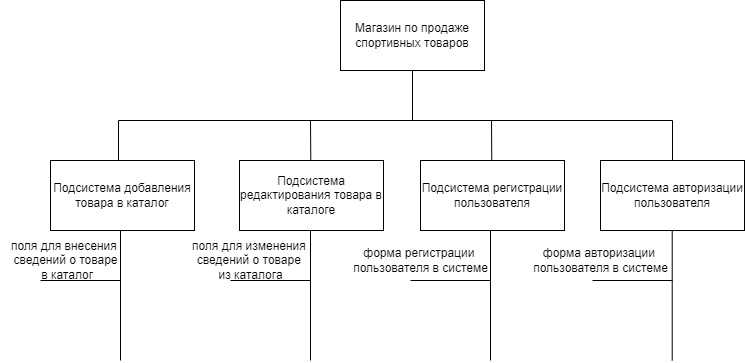


Рисунок 1.1 – Структурная схема веб-приложения

Диаграмма деятельности

На диаграмме деятельности показаны взаимодействия между различными элементами системы при выполнении типичного сценария использования, например, добавление товара в каталог. В ходе анализа задания выделены следующие роли:

– Администратор;

– Менеджер;

– Авторизованный клиент;

– Неавторизованный пользователь;

Роль «администратора» позволяет выполнять следующие действия:

– Редактирование товара;

– Удаление товара;

– Добавление товара;

Роль «менеджер» позволяет выполнять следующие действия:

– просмотр каталога;

Роль «авторизованный клиент» позволяет выполнять следующие действия:

– просмотр каталога;

– сортировка товара;

– поиск товара;

– добавление фильтров к товарам;

Разработанная архитектура решения обеспечивает гибкость и масштабируемость системы, позволяет легко добавлять новые функциональные возможности и интегрироваться со сторонними сервисами. Использование современных технологий и инструментов разработки делает систему надежной и безопасной.

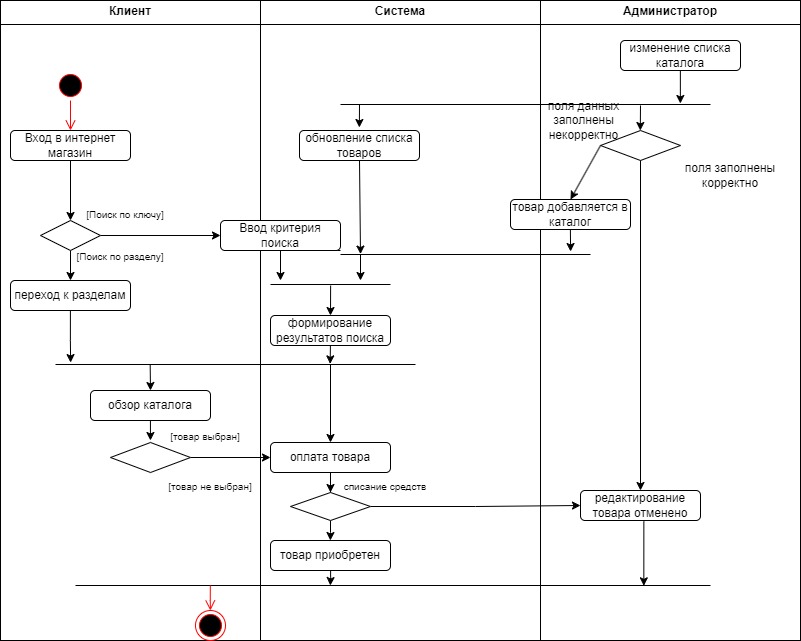


Рисунок – диаграмма деятельности

## **1.4 Обоснование и выбор средств разработки решения**

Выбранное средство разработки решения для данного курсового проекта на тему "Магазин по продаже спортивных товаров" - веб-ресурс для учета товаров нашего сайта "Магазин спортивных товаров" - состоит из трех основных компонентов: HTML, CSS и JavaScript. Наиболее подходящим и распространенным инструментом, который обеспечивает все эти возможности является Python с использованием фреймворка Flask. Выбор средства разработки веб-ресурса для данного проекта сделан, исходя из следующих причин:

1) HTML — это стандартный язык разметки веб-страниц, который является основным языком для создания всей структуры и функциональности веб-сайта.

2) CSS — это язык таблиц стилей, который позволяет определить стиль веб-страницы, изменять ее внешний вид и улучшать пользовательский опыт.

3) JavaScript — это язык программирования, который позволяет создавать интерактивные элементы веб-страниц, улучшать навигацию по сайту и предоставлять пользователю средства для взаимодействия со страницей.

4) Python — это высокоуровневый язык программирования общего назначения, который часто используется для веб-разработки благодаря своей читаемости, гибкости и большому количеству библиотек. Flask — это микрофреймворк для Python, который предоставляет минимальный набор инструментов для быстрого создания веб-приложений.

5) База данных сделана через SQLite. Это легковесная система управления базами данных, которая отлично подходит для небольших и средних проектов. Она не требует отдельного сервера и может быть использована непосредственно из файловой системы.

6) Python с Flask является мощным инструментом для создания веб-приложений, поддерживает работу с различными базами данных, включая SQLite, MySQL, PostgreSQL и другие. В результате, выбор использования CSS, Python и JavaScript оправдан, так как они позволяют создавать качественный и функциональный веб-ресурс для работы с товарами на сайте. Кроме того, использование SQLite позволит обеспечить простоту и удобство работы с базой данных и приложением в целом.

## **Формализация расчетов**

1) Определить ожидаемые результаты работы подсистемы:

- Возможность просмотра списка товаров: Пользователи должны иметь возможность просматривать доступные товары в магазине, включая их описание, цену, наличие и другую информацию.

- Возможность приобретения товаров: Пользователи должны иметь возможность добавлять товары в корзину и оформлять заказ.

2) Определить методы передачи данных:

Использование базы данных через SQLite: Для хранения информации о товарах и заказах будет использоваться легковесная система управления базами данных SQLite.

-Использование языков программирования;

-HTML: Для создания структуры веб-страниц;

-CSS: Для стилизации элементов интерфейса;

-Python: Для написания обработки запросов;

-JavaScript: Для реализации интерактивности;

3) Определить логику работы подсистемы:

А) Клиенты могут просматривать список товаров: Пользователи должны иметь возможность просматривать все доступные товары в магазине, фильтровать их по категориям, цене, бренду и другим параметрам.

Б) Клиенты могут приобретать товары: Пользователи должны иметь возможность добавлять товары в корзину, оформлять заказ, выбирать способ оплаты и доставки.

4) Определить использование инструментов программирования:

А) Для создания интерфейса можно использовать языки программирования HTML и CSS.

Б) Для обработки запросов и отображения данных необходимо использовать сценарии на языке программирования Python.

В) Для реализации динамической интерактивности можно использовать библиотеку JavaScript.

5) Определить требования к базе данных:

А) Необходимо определить таблицы хранения информации о товарах, включая их название, описание, цену, наличие, категории и другие параметры.

Б) Также необходимо определить таблицы для хранения информации о заказах, включая данные о покупателе, выбранных товарах, способе оплаты и доставки.

6) Определить и описать процессы и взаимодействия между компонентами системы:

А) Клиенты могут просматривать список товаров: Пользователи могут просматривать все доступные товары в магазине, используя веб-интерфейс, который будет генерироваться на основе данных из базы данных.

Б) Клиенты могут приобретать товары: Пользователи могут добавлять товары в корзину, оформлять заказ, выбирать способ оплаты и доставки, после чего информация о заказе будет сохраняться в базе данных.

### **1.5.1 Проектирование базы данных**

Для проектирования базы данных для веб-ресурса «Магазин по продаже спортивных товаров» необходимо следовать определенной структуре. При этом можно использовать реляционную модель. Следующие таблицы могут быть использованы в данной базе данных:

1) Таблица "users":

- ID пользователя (уникальный идентификатор);

- Логин пользователя;

- Адрес электронной почты;

- ФИО пользователя;

- Пароль пользователя;

- Баланс пользователя.

2) Таблица "item":

- ID товара (уникальный идентификатор);

- Название производителя;

- Количество товара на складе;

- Описание товара.

Проектирование базы данных для магазина по продаже спортивных товаров на Flask включает в себя создание таблиц, которые будут хранить информацию о товарах, заказах, клиентах и других аспектах бизнеса. Ниже приведен примерный план проектирования базы данных для такого магазина.

### Для понимания того, как таблицы связаны в системе была разработана ER-диаграмма, на которой можно рассмотреть типы данных, соответствующие каждой колонке в таблице. ER-диаграмма выполнена в соответствии с рисунком 1.4.

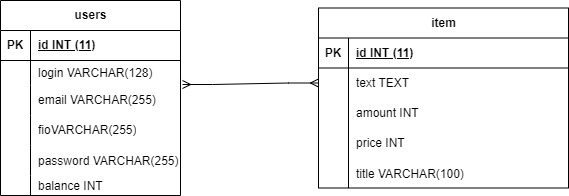


Рисунок 1.4 – ER-диаграмма

### **1.5.2 Проектирование интерфейса**

Для успешного проектирования интерфейса магазина по продаже спортивных товаров на Flask необходимо учесть следующие моменты:

1) Навигация по сайту должна быть интуитивно понятной и легкой в использовании. Это можно обеспечить с помощью главного меню, которое будет содержать ссылки на все разделы сайта.

2) Главная страница должна представлять собой витрину магазина, где будут отображаться популярные товары, акции и специальные предложения.

3) Страницы товаров должны содержать детальное описание, фотографии, цену и возможность добавления в корзину.

4) Корзина покупок должна быть легко доступна и позволять покупателям изменять количество товаров, удалять их или оформлять заказ.

5) Форма оформления заказа должна быть простой и понятной, требуя минимального количества информации от покупателя.

6) Весь интерфейс должен быть адаптирован под различные устройства и разрешения экрана, чтобы обеспечить комфортное использование сайта на любом устройстве.

7) Дизайн сайта должен соответствовать общим стандартам и требованиям, включая использование читаемых шрифтов, гармоничной цветовой гаммы и логотипа.

Проектирование макета интерфейса пользователя выполнен в соответствии с рисунками 1.5–1.9.

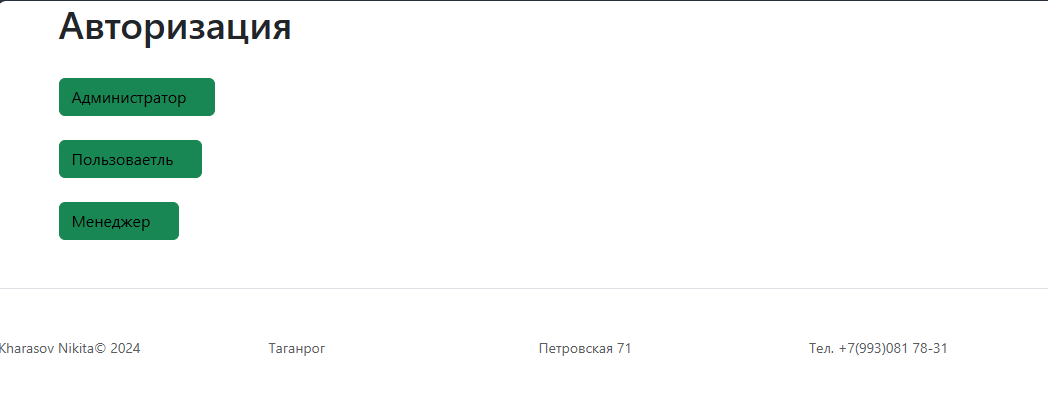


Рисунок 1.5 – Выбор роли пользователя

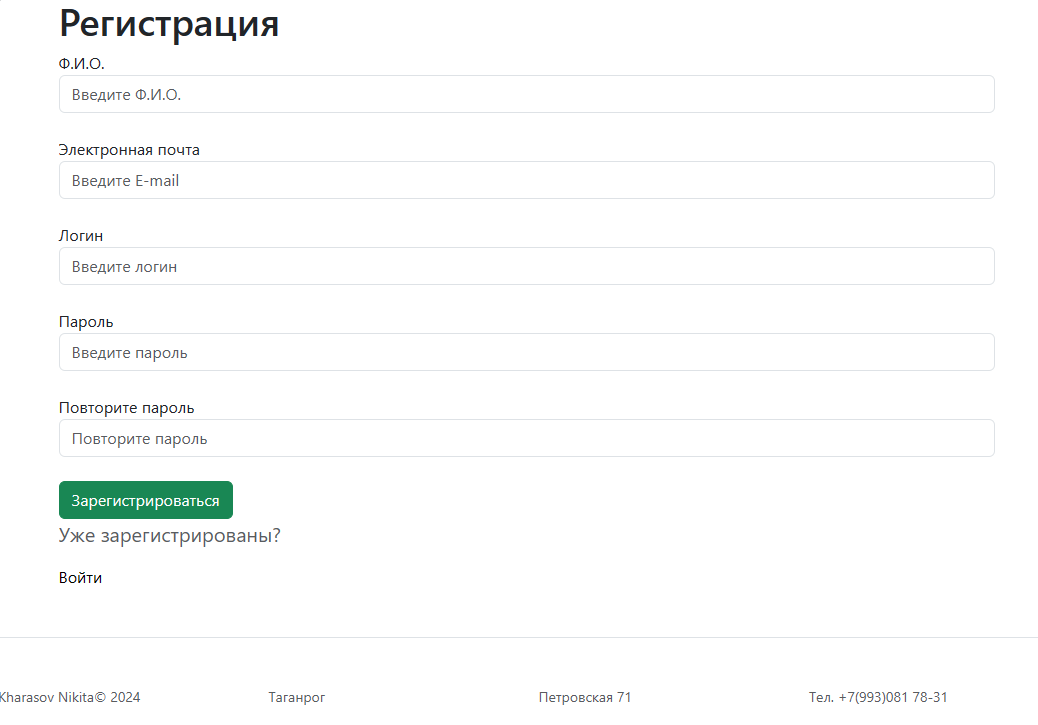


Рисунок 1.6 – Регистрация пользователя

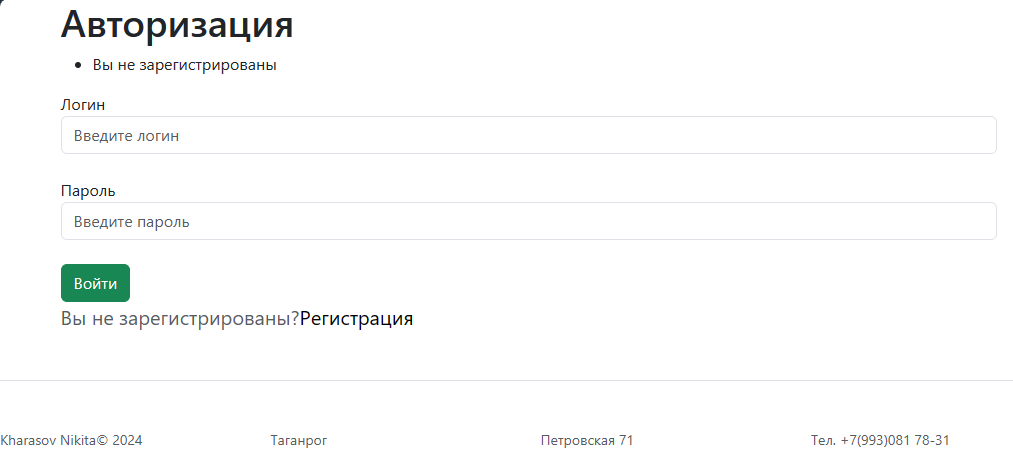


Рисунок 1.7 – Авторизация пользователя

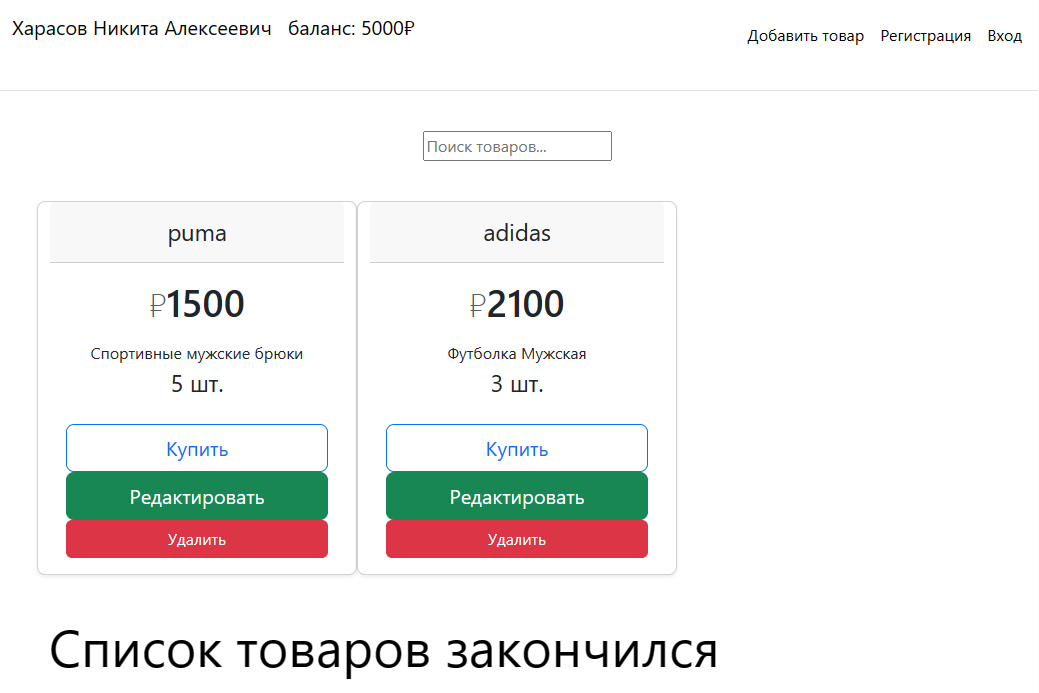


Рисунок 1.8 – Главная страница приложения

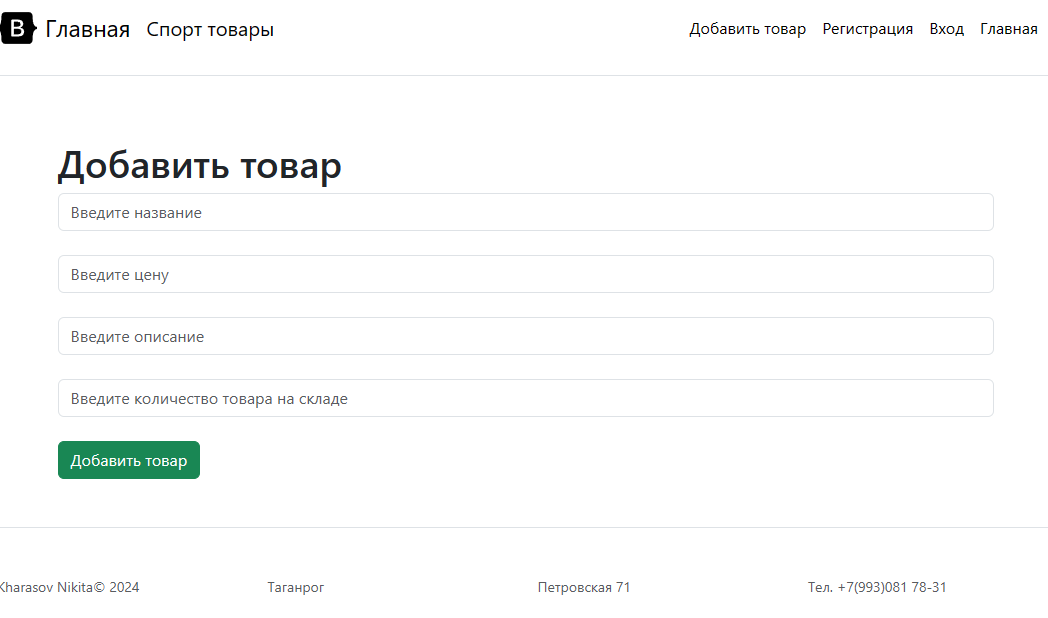


Рисунок 1.9 - Добавление товара в каталог

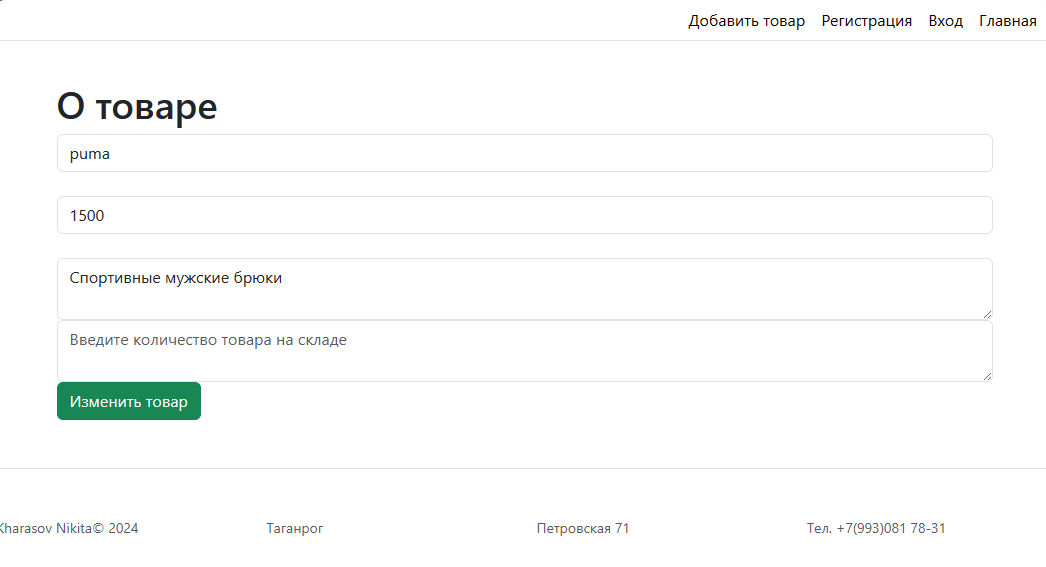


Рисунок 1.10 – Редактирование полей товара

# **Описание программы**

# Программа, разработанная для автоматизации деятельности магазина по продаже спортивных товаров, представляет собой веб-приложение, написанное на Flask (Python), которое позволяет управлять товарными запасами и складскими операциями.

main.py – основной файл приложения, который содержит точку входа для выполнения программы.

Диаграмма компонентов разработанного приложения представлена в соответствии с рисунком 2.1.

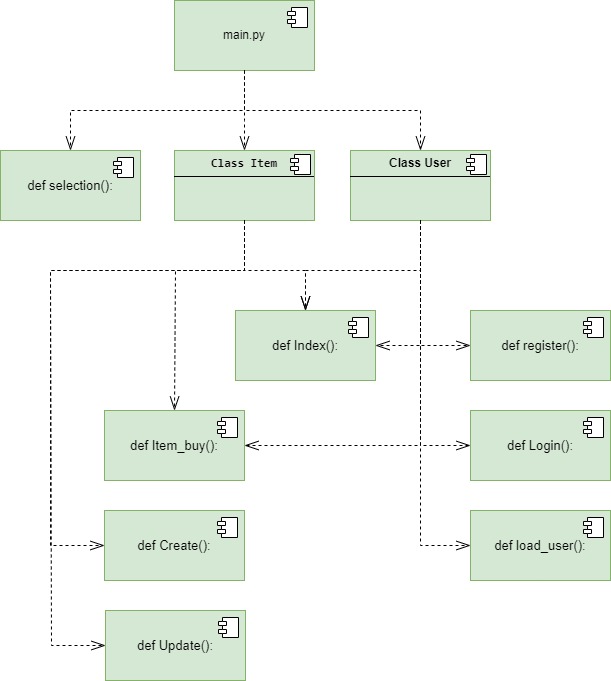


Рисунок 2.1 – Диаграмма компонентов

# Основное назначение этой программы заключается в обеспечении оперативного и эффективного учета, распределения и управления запасами спортивных товаров. Система включает в себя широкий спектр функциональных возможностей, предназначенных для облегчения работы персонала магазина и оптимизации процессов управления товарными запасами.

Основные возможности системы включают:

1) просмотр списка товаров и их наличия на складах, представленного в удобном для пользователя. Пользователи имеют возможность фильтрации по категориям и просмотра информации о количестве конкретного товара.

2) осуществление добавление, редактирование, а также удаление товаров в каталоге магазина.

Для разработки веб-приложения использовался Flask (Python), что позволило создать простой в использовании и функциональный ресурс, который значительно упрощает и автоматизирует процессы управления товарными запасами в магазине спортивных товаров.

# **Тестирование программы**

**3.1 Разработка плана тестирования**

1) План проведения тестирования:

2) Проверить работоспособности авторизации;

3) Определить входные данные

4) Проверить работоспособность добавления товара в каталог;

5) Проверить работоспособность функции редактирования товара из каталога.

Для проверки авторизации был введен логин и пароль в соответствующие поля и нажата кнопка «Войти». Пример авторизации выполнен в соответствии с рисунком 3.1.

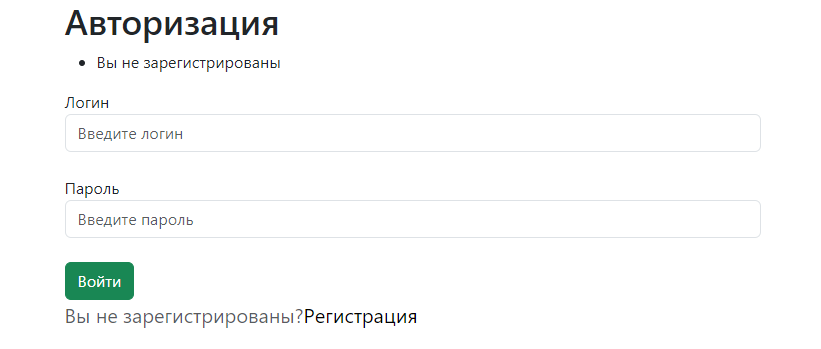
****

Рисунок 3.1 Авторизация

Для проверки работоспособности функции создания товара были введены данные товара в соответствующие поля и нажата кнопка «Добавить товар». Пример создания товара выполнен в соответствии с рисунком 3.2.

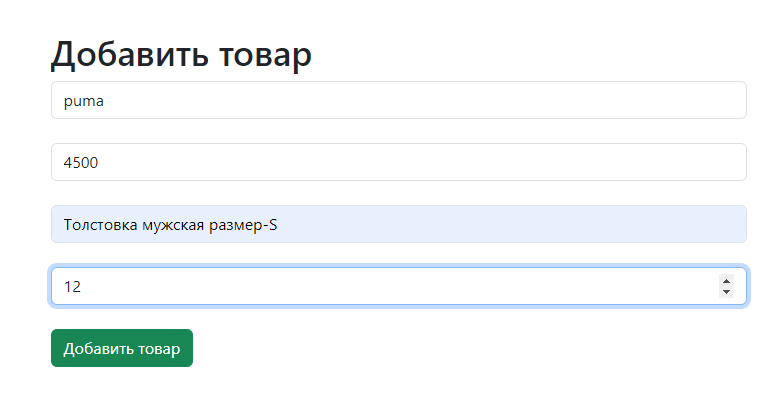


Рисунок 3.2 Создание товара

Для проверки работоспособности функции изменение данных о товаре, была нажата кнопка «Редактировать», после чего введены данные в соответствующие поля и выполнено подтверждение изменений нажатием кнопкой «Изменить товар», после чего нас перенесут на страницу «Авторизации».

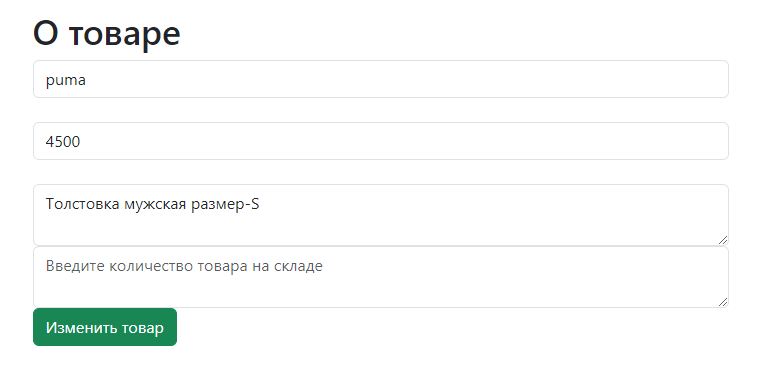
****

Рисунок 3.3 Редактирование товара

* 1. **Оценка результатов проведения тестирования**

«Тестирование программного продукта проводилось согласно плану, представленному в пункте 3.1 данной пояснительной записки. В ходе тестирования были проверены следующие функциональные возможности:

1) Проверка авторизации. Пользователь вводит корректные данные (логин и пароль), система проверяет их и предоставляет доступ к функционалу программы. Рисунок 3.1 иллюстрирует процесс авторизации.

2) Проверка создания товара. Система позволяет пользователю создать новый товар, заполнив необходимые поля. Процесс создания товара отображён на рисунке 3.2

3) Проверка функции редактирования товара. Пользователь выбирает существующий товар и изменяет его характеристики. После сохранения обновлённая информация отображается в каталоге товаров.

Все тестовые сценарии были успешно выполнены без ошибок. Таким образом, можно сделать вывод о том, что разработанный программный продукт соответствует требованиям технического задания и готов к эксплуатации».

# **Заключение**

В ходе работы над проектом были изучены и применены передовые технологии веб-разработки, включая HTML, CSS и Python, а также базы данных с использованием SQLite. Программа предоставляет широкий спектр функциональных возможностей, позволяющих управлять процессами поступления, хранения и продажи спортивных товаров, а также обеспечивает генерацию необходимых отчетов для оптимизации складских операций.

В результате работы над проектом были выявлены некоторые недостатки, однако система остается весьма перспективной и может быть доработана и улучшена в дальнейшем. В целом, разработанная информационная система "Магазин спортивных товаров" является значимым шагом в направлении современной автоматизации складских операций спортивного магазина, обеспечивая более эффективное управление запасами спортивных товаров и повышение качества обслуживания клиентов.

# **Список используемых источников**

* 1. https://metanit.com/
  2. С.Куликов Работа с MySQL, MS SQL Server и Oracle в примерах
  3. https://svyatoslav.biz/database\_book\_download/
  4. Васильев, Алексей. Программирование на Python в примерах и задачах / Алексей
  5. Васильев. — Москва: Эксмо, 2021. — 616 с. — (Российский компьютерный бестселлер)
  6. <https://www.w3schools.com/python/python_mysql_getstarted.asp>