**Информационная система для магазина аниме-атрибутики**

**Руководство программиста**

**СОДЕРЖАНИЕ**

[1 Введение 2](#_Toc181481376)

[1.1 Наименование системы 2](#_Toc181481377)

[1.2 Область применения 2](#_Toc181481378)

[2 Требования к окружению 3](#_Toc181481379)

[3 Настройка окружения разработки 4](#_Toc181481380)

[4 Установка и запуск приложения 5](#_Toc181481381)

[5 Внутреняя структура системы 6](#_Toc181481382)

[5.1 Используемый технологический стек 6](#_Toc181481383)

[5.2 Функции и их описание 6](#_Toc181481384)

[5.3 Структура данных 8](#_Toc181481385)

# Введение

## Наименование системы

Информационная система для магазина аниме атрибутики (далее ИС).

## Область применения

ИС для магазина аниме атрибутики включает в себя инструменты для маркетинга и анализа данных. Эта система предоставляет разнообразные функциональные возможности, помогающие владельцам и сотрудникам магазина управлять ассортиментом, заказами, складскими запасами.

# Требования к окружению

Минимальные требования к окружению:

1. Операционная система: Windows 7 и выше;
2. Оперативная память (ОЗУ): 4 ГБ;
3. Жесткий диск: 5 ГБ;
4. Процессор: Intel Core i3 или аналогичный;
5. Разрешение экрана: минимум 1024x768 пикселей для удобного отображения интерфейса;
6. Сетевое подключение: стабильное соединение с высокой скоростью передачи данных для корректного подключения и работы базы данных.
7. Python: Версия 3.8 и выше;
8. Библиотеки Python:
   * PyQt5 для создания графического интерфейса;
   * «mysql-connector-python» для работы с базой данных MariaDB.
9. СУБД: MariaDB версии 8.0 и выше.

# **Настройка окружения разработки**

Для настройки окружения выполните следующие действия:

1. Установите [Python](https://www.python.org/downloads/) версии 3.8 или выше;
2. Для удобства можно установить IDE PyCharm;
3. Установите необходимые библиотеки с помощью команды «pip install PyQt5 mysql-connector-python»;
4. Проверьте соединение с базой данных. Параметры конфигурации представлены на рисунке 1.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 - конфигурация БД

# Установка и запуск приложения

Для установки и запуска приложения необходимо выполнить следующие действия:

1. Клонируйте репозиторий с GitHub:

* «git clone <https://github.com/ALEKCZY/IS-Anime-Store.git>»

1. Перейдите в директорию:

* «cd <название\_директории>»

1. Запустите программу для менеджеров:

* «python AS.py»

1. Если хотите запустить приложение для администратора:

* «python PU.py»

# Внутреняя структура системы

## Используемый технологический стек

Для разработки системы использовался следующий технологический стек:

1. Язык программирования: Python — основной язык разработки, который обеспечивает гибкость и богатый выбор библиотек для работы с базой данных, интерфейсом и валидацией данных.
2. Фреймворк для пользовательского интерфейса: PyQt5 — библиотека для создания графического интерфейса пользователя (GUI) на Python. PyQt5 предоставляет широкий набор компонентов для построения интерфейсов, таких как окна, кнопки, таблицы и формы, что позволяет создавать удобные и интуитивно понятные приложения.
3. База данных: MariaDB — это популярная система управления базами данных с открытым исходным кодом, которая была создана как ответвление от MySQL после того, как компанию MySQL AB приобрела Oracle. Основные особенности MariaDB включают высокую производительность, надежность и совместимость с MySQL
4. Библиотека для работы с базой данных: mysql.connector — библиотека Python для подключения и работы с MySQL. Она обеспечивает удобный интерфейс для выполнения SQL-запросов, включая операции добавления, обновления, удаления и выборки данных.

## Функции и их описание

Программа для управления и учета товаров в магазине аниме товаров предоставляет следующие функции, организованные по ролям пользователей:

Функции менеджера:

1. Просмотр каталога товаров:

* Менеджер может просматривать весь каталог доступных товаров, включая детальную информацию;
* Цель функции — обеспечить менеджеру быстрый доступ к полному перечню товаров.

1. Фильтрация товаров по категориям:

* Менеджер может отфильтровать товары по определенным категориям;
* Это помогает сузить результаты поиска и улучшить навигацию по каталогу.

1. Поиск товара по названию:

* Менеджер может искать конкретный товар по названию, что позволяет быстро находить нужные позиции, особенно при большом ассортименте;
* При вводе части названия программа отображает все соответствующие позиции.

1. Оформление заказов:

* Менеджер может оформить заказ на выбранные товары, добавив их в корзину и указав количеств;
* После подтверждения заказа система фиксирует его и обновляет количество товаров на складе;
* Эта функция автоматизирует процесс оформления заказа, делая его быстрым и удобным.

Функции администратора:

Администратор имеет доступ ко всем функциям менеджера и дополнительным административным функциям:

1. Добавление нового товара:

* Администратор может добавить новый товар в базу данных, указав его название, описание, цену и категорию;
* Функция предназначена для пополнения ассортимента и актуализации каталога товаров.

1. Изменение существующего товара:

* Администратор может редактировать информацию о любом товаре;
* Это позволяет актуализировать данные о товарах, обеспечивая правильную информацию для менеджеров.

1. Удаление товара:

* Администратор может удалить товар из каталога, если он больше не доступен;
* Функция поддерживает актуальность ассортимента и удаляет устаревшие позиции.

1. Добавление новой категории:

* Администратор может добавлять новые категории товаров;
* Это улучшает навигацию по каталогу и позволяет пользователям проще находить товары по новым категориям.

1. Изменение категории:

* Администратор может редактировать информацию о категории, например, изменять её название;
* Это помогает поддерживать структуру категорий в актуальном состоянии.

1. Удаление категории:

* Администратор может удалить категорию, если она больше не актуальна;
* Эта функция обеспечивает упрощение каталога, убирая ненужные категории.

## Структура данных

Все данные берутся из БД, в которой существует несколько таблиц. В таблице Категории хранится следующая информация:

* Код категории – уникальный код категории, который автоматически генерируется при ее создании;
* Название – название категории.

В таблице Товары имеются следующие столбцы:

* Код товара – уникальный код товара, который автоматически генерируется при создании;
* Название – название товара;
* Код категории – код категории, который берется из таблицы Категории;
* Цена – цена одного товара;
* Количество – количество определенного товара на складе.

В таблице Заказы имеются следующие столбцы:

* Код заказа – уникальный код заказа, который автоматически генерируется при оформлении заказа;
* Код товара – код товара, который берется из таблицы Товары;
* Дата – дата оформления заказа;
* Статус – статус заказа, администратор может выбрать один из 3 статусов (Оформлен, отправлен и получен);
* Сумма – общая сумма заказа.

Логическая ER диаграмма показана на рисунке 2.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 2 - Логическая ER диаграмма

Данная модель имеет три сущности: Категории, Товары и Заказы.

В базе данных были определены следующие связи:

1. «код\_категории» в таблице «Товары» связана с «код\_категории» в таблице «Категории» - один ко многим, один товар имеет только одну категорию, но у одной категории может быть несколько товаров;
2. «код\_товара» в таблице «Заказы» связана с «код\_товара» в таблице «Товары» - многие ко многим, один заказ может иметь несколько товаров и один товар может участвовать в нескольких заказах;

Физическая ER диаграмма представлена на рисунке 3.

Изображение выглядит как снимок экрана, текст, Шрифт, черный

Автоматически созданное описание

Рисунок 3 - Физическая ER диаграмма