МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Старооскольский технологический институт

им. А.А. УГАРОВА

(филиал) федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский университет МИСИС»

ОСКОЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

**ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

по специальности:

Информационные системы и программирование

Студент И. И. Иванов

Специальность 09.02.07

Группа ИСП-24-Х

Старый Оскол, 2025

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА** (филиал) федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский технологический университет «МИСИС»

**ОСКОЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

**ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

по специальности:

Информационные системы и программирование

по МДК 09.01. Проектирование и разработка веб-приложений

МДК 09.02. Оптимизация веб-приложений

Тема: Разработка веб-приложения музыкального магазина

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. И. Иванов

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. М. Коренев

Старый Оскол, 2025

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА** (филиал) федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский технологический университет «МИСИС»

**ОСКОЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ**

**Задание на ИнДИВИДУАЛЬНЫЙ ПРОЕКТ**

|  |
| --- |
| **Студент: Иванов Иван Иванович** |
| **Специальность: 09.02.07 Информационные системы и программирование** |
| **Тема работы: Разработка веб-приложения музыкального магазина** |

В индивидуальный проект должны входить следующие пункты:

**Титульный лист**

**Задание на индивидуальный проект**

**Информативный реферат (Abstract)**

**Содержание**

**Введение**

1. **Теоретическая часть** 
   1. **Анализ предметной области** 
      * Описание проекта
      * Анализ пользователей продукта
      * Анализ конкурентов
   2. **Проектирование разрабатываемой системы**
      * Постановка цели
2. **Практическая часть**

**2.1. Соадминистрирование базы данных и серверов**

* Проектирование ER-диаграммы
* Разработка базы данных, объектов баз данных

**2.2. Разработка и интеграция программных модулей**

* Разработка API
* Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения
* Сопровождение информационной системы

**2.3. Реализация информационной безопасности**

**2.4. Разработка веб-модуля для взаимодействия с API**

* Разработка требований к функциям сайта
* Разработка прототипов пользовательских интерфейсов системы

**Заключение**

**Список использованных источников**

**ПРИЛОЖЕНИЕ А. Учетные записи программного продукта**

**ПРИЛОЖЕНИЕ Б. Листинг программного продукта**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc213430752)

[1.1. Анализ предметной области 7](#_Toc213430753)

[1.2. Проектирование разрабатываемой системы 12](#_Toc213430754)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 13](#_Toc213430755)

[2.1. Соадминистрирование базы данных и серверов 13](#_Toc213430756)

[2.2. Разработка и интеграция программных модулей 16](#_Toc213430757)

[2.3. Реализация информационной безопасности 24](#_Toc213430758)

[2.4. Разработка веб-модуля для взаимодействия с API 25](#_Toc213430759)

[ЗАКЛЮЧЕНИЕ 33](#_Toc213430760)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc213430761)

[ПРИЛОЖЕНИЕ А 38](#_Toc213430762)

[ПРИЛОЖЕНИЕ Б 39](#_Toc213430763)

[ПРИЛОЖЕНИЕ В 40](#_Toc213430764)

# **ВВЕДЕНИЕ**

В современном мире цифровые технологии играют важную роль в обеспечении эффективной работы предприятий розничной торговли и их взаимодействия с клиентами. Для успешного ведения бизнеса, увеличения продаж и охвата новой аудитории необходимы современные и удобные онлайн-решения. Одной из ключевых задач таких решений является создание интуитивно понятного и функционального веб-приложения, представляющего собой основной цифровой канал для презентации товаров, совершения покупок и построения коммуникации с потребителем.

Актуальность разработки обусловлена стремлением современных компаний повысить эффективность продаж, оптимизировать бизнес-процессы за счет автоматизации и выйти на новый уровень обслуживания клиентов в цифровой среде.

Цель данного индивидуального проекта заключается в проектировании веб-приложения музыкального магазина, которое будет обеспечивать удобный и надежный способ покупки музыкальных инструментов и оборудования, облегчая клиентам поиск товаров и доступ к актуальной информации, а владельцам — управление ассортиментом и заказами.

Проектируемая система будет включать основные функции по каталогизации товаров, управлению заказами и взаимодействию с пользователями, что позволит минимизировать типичные ошибки ручного управления магазином и повысить удовлетворенность клиентов.

Предметом исследования данного проекта является разрабатываемое веб-приложение музыкального магазина.

Объектом исследования является магазин музыкальных инструментов.

Конкурентоспособность предлагаемого проекта обеспечивается его ориентацией на потребности современных покупателей, удобным пользовательским интерфейсом, а также возможностью масштабирования для адаптации под растущие потребности бизнеса.

Практическая значимость работы заключается в том, что внедрение такого веб-приложения позволит компании значительно увеличить продажи, расширить целевую аудиторию и оптимизировать внутренние бизнес-процессы, связанные с торговлей

**1.** **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1.1. Анализ предметной области**

Описание проекта

Музыкальная индустрия, несмотря на цифровизацию, продолжает процветать. Потребность в покупке физических носителей (дисков, виниловых пластинок) сохраняется, а покупка с гарантией самих инструментов продолжает расти.

В то же время, онлайн-продажа музыкальных товаров становится всё более популярной. Интернет-магазины предлагают широкий выбор продукции, удобные способы оплаты и доставки, а также дополнительные услуги, такие как музыкальные рекомендации, инструкции к инструментам, онлайн гарантию и тех. поддержку.

Разработка веб-приложения для музыкального магазина актуальна по следующим причинам:

* + Увеличение онлайн-продаж: Интернет-магазины стали ключевым каналом продаж для музыкальных товаров, предоставляя доступ к широкому ассортименту и удобным сервисам.
  + Конкуренция: Рынок онлайн-продаж музыки конкурентный, поэтому важно создавать приложения, предлагающие уникальные функции и выгодные предложения.
  + Повышение удовлетворенности клиентов: Веб-приложение может улучшить взаимодействие с клиентами, предлагая удобные функции поиска, сравнения товаров, оформления заказов, а также возможность получения актуальной информации о новостях и акциях, проверку гарантии и получение помощи у тех. поддержки.

Логотип был создан в программе «Figma» с использованием таких инструментов как хост, линии, текст. В логотип был добавлен так же рисунок. Последующие редактирование были произведены в программе «Paint» – удаление лишних объектов, обрезка. В конце в онлайн программе «Remove BG» был убран фон у логотипа.



Рисунок 1 – Логотип

Анализ пользователей продукта

Целевая аудитория:

* + - Музыкальные энтузиасты: Люди, которые увлечены музыкой и регулярно покупают музыкальные товары.
    - Коллекционеры: Люди, которые собирают музыкальные записи (винил, CD, кассеты) и ценят редкие издания.
    - Поклонники артистов: Люди, которые хотят приобрести товары любимых исполнителей (альбомы, мерч, билеты на концерты, инструменты).

Основные характеристики:

* + - Возраст: 9–35 лет.
    - Интересы: Музыка разных жанров, коллекционирование, концерты, фестивали, обучение музыкальным инструментом.
    - Уровень дохода: Средний и выше среднего.
    - География: Россия.

Анализ конкурентов

1. Музторг (https://www.muztorg.ru/)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Автоматически созданное описание

Рисунок 2 — Главная страница «Музторг»

Описание: Крупнейший интернет-магазин музыкальных инструментов и оборудования в России. Предлагает широкий ассортимент товаров от бюджетных до премиальных, а также услуги по ремонту и настройке инструментов.

Достоинства:

* + - Большой выбор товаров.
    - Наличие собственных сервисных центров.
    - Удобная система поиска и фильтрации товаров.
    - Возможность покупки в рассрочку.

Недостатки:

* + - Сайт может быть перегружен информацией.
    - Доставка в некоторые регионы может быть дорогой.
    - Не всегда удаётся найти подробные описания товаров.

1. Мир музыки (https://mirm.ru/)

Изображение выглядит как текст, снимок экрана, Веб-сайт, программное обеспечение

Автоматически созданное описание

Рисунок 3 — Главная страница «Мир музыки»

Описание: Интернет-магазин музыкальных инструментов и оборудования с большим выбором товаров от разных производителей. Предлагает услуги по подбору инструментов и аксессуаров, а также консультации специалистов.

Достоинства:

* Удобная система онлайн-оплаты.
* Наличие онлайн-консультации.
* Широкий выбор брендов.

Недостатки:

* Не много устаревший дизайн.
* Отсутствует возможность покупки в рассрочку.

1. Muzikroom (https://muzikroom.ru/)

Изображение выглядит как музыкальный инструмент, текст, снимок экрана, струнный инструмент

Автоматически созданное описание

Рисунок 4 — Главная страница «Muzikroom»

Описание: Интернет-магазин, специализирующийся на продаже музыкальных инструментов и оборудования для профессиональных музыкантов. Предлагает широкий выбор брендовых товаров, а также услуги по настройке и ремонту.

Достоинства:

* Высокое качество товаров.
* Наличие собственных мастерских.
* Профессиональный подход к обслуживанию клиентов.
* Возможность покупки в рассрочку.

Недостатки:

* Цены выше, чем у конкурентов.
* Ограниченный выбор товаров для начинающих музыкантов.

# **1.2.****Проектирование разрабатываемой системы**

Постановка цели

Цели проекта:

* 1. Создание платформы для онлайн-продажи музыкальных товаров.
  2. Обеспечение удобного и интуитивно понятного интерфейса для пользователей.
  3. Предложение широкого ассортимента продукции (диски, виниловые пластинки, аксессуары, мерч, музыкальные инструменты).
  4. Внедрение системы гарантии.
  5. Обеспечение безопасных и удобных способов гарантии, технической поддержки.

Задачи проекта

* + 1. Провести анализ конкурентной среды.
    2. Определить целевую аудиторию.
    3. Разработать структуру и дизайн веб-приложения.
    4. Создать функциональные модули (каталог товаров, корзина, оформление заказа, личный кабинет).
  1. Внедрить систему поиска и фильтрации товаров.
  2. Интегрировать систему оплаты и доставки.
  3. Провести тестирование и отладку приложения.
  4. Запустить веб-приложение в сети Интернет (присвоить домен, опубликовать на хостинг).
  5. Адаптировать приложение под различные платформы.
  6. Продемонстрировать сайт.

# **2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

## **2.1. Соадминистрирование базы данных и серверов**

Проектирование ER-диаграммы

ER-диаграмма (схема «сущность-связь») — графическая схема, которая показывает, какие данные есть в системе и как они связаны. ER-диаграмму строят при помощи графических элементов, выделяют объекты и их характеристики, определяют типы связей между объектами.

На рисунке 5 можно увидеть ER-диаграмму.

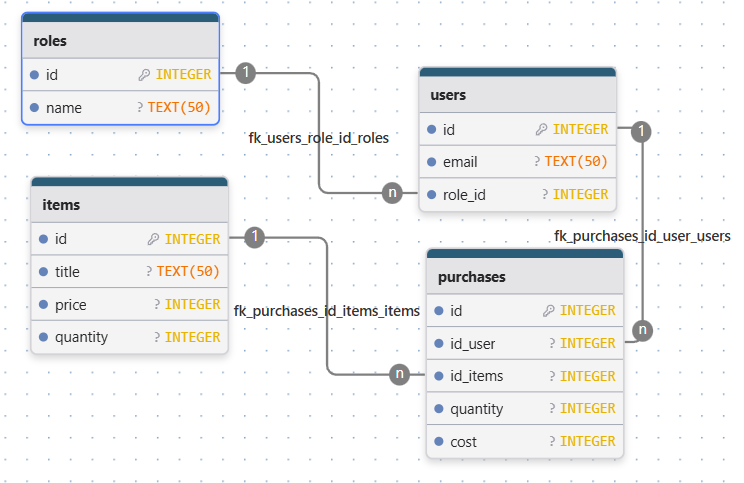


Рисунок 5 –Диаграмма «Сущность-связь»

Разработка базы данных, объектов баз данных

MySQL:

* Плюсы: Бесплатная и открытая, хорошо документирована, поддерживается множество платформ и языков программирования.
* Минусы: Меньше функциональности в сравнении с некоторыми другими СУБД, такими как Oracle или SQL Server.

PostgreSQL:

* Плюсы: Мощная и расширяемая, поддерживает множество функций, включая JSON, XML, и геоданные. Открытая и бесплатная.
* Минусы: Некоторые пользователи находят ее сложной в настройке и управлении.

Oracle:

* Плюсы: Масштабируемая и надежная, обладает множеством расширенных функций и возможностей для управления данными.
* Минусы: Высокая стоимость лицензий и поддержки, что делает ее менее доступной для небольших компаний.

SQL Server (Microsoft):

* Плюсы: Интегрирована с другими продуктами Microsoft, такими как Windows Server и Azure. Поддерживает масштабирование и высокую доступность.
* Минусы: Лицензионная стоимость, ограничения по платформам (в основном для Windows).

MongoDB (NoSQL):

* Плюсы: Гибкая схема данных, горизонтальное масштабирование, поддержка для больших объемов неструктурированных данных.
* Минусы: Не подходит для всех типов данных и запросов, менее строгая консистентность данных в сравнении с реляционными СУБД.

База данных включает в себя 4 таблицы, каждая из которых отвечает за хранение определенного типа данных:

* + Пользователи: Хранение информации о пользователях, включая данные об учетной записи в хешированном виде, ФИО и принадлежность к определенной роли;
  + Роли: Хранение информации о ролях, включающее номер записи и название роли для общего понятия их нахождения там;
  + Товары: Хранение информации о товарах в магазине
  + Покупки: Хранение информации совершенных покупках пользователями.

# **2.2. Разработка и интеграция программных модулей**

Разработка API

Разработка API музыкального магазина "Музыкальный рай" была реализована с использованием фреймворка Django 5.1.3 на языке программирования Python. Backend отвечает за обработку данных, взаимодействие с базой данных и реализацию бизнес-логики. Процесс создания серверной части сайта был разделен на несколько этапов:

**Этап 1: Настройка проекта Django**

1. Создание проекта:

Сначала был инициализирован новый Django-проект с названием MDK\_Musical\_Paradise с использованием команды:



Рисунок 6 – Команда создание проекта

В корневой директории проекта автоматически создались файлы конфигурации, включая settings.py, отвечающий за настройки проекта.

1. Настройка базы данных MySQL:

В файле settings.py был указан драйвер и параметры подключения к базе данных MySQL:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 7 – Подключение базы данных

После этого была создана база данных и выполнена миграция для применения стандартных таблиц Django:



Рисунок 8 – Миграция базы данных

**Этап 2: Создание приложений**

Для разделения функциональности сайта проект был разбит на несколько приложений, каждое из которых решает конкретную задачу:

* authorization – отвечает за регистрацию, авторизацию и аутентификацию пользователей.
* main – обрабатывает главную страницу и статический контент.
* messengers – реализует личные сообщения между пользователями.
* myprofile – позволяет пользователям управлять своим профилем, редактировать личные данные.
* shop – отвечает за каталог товаров, их отображение, фильтрацию и работу корзины.

Каждое приложение создавалось с использованием команды:



Рисунок 9 – Создание приложения

После этого приложения добавлялись в файл settings.py:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 10 – Добавление приложений в настройки

**Этап 3: Разработка моделей и миграции**

На основе требований сайта были созданы модели для базы данных. Например, в приложении myprofile реализованы модели для отзывов:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Модель отзывов

**Этап 4: Создание представлений и маршрутов**

Представления (views):

В каждом приложении были созданы функции и классы для обработки запросов. Например, для отображения списка товаров:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 12 – Функция обработки ссылке «каталог»

Маршруты (urls):

Для каждой функции создавались маршруты в файле urls.py приложения.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 13 – Маршруты приложения «shop»

**Этап 6: Адаптация под мобильные устройства и планшеты**

Одним из ключевых этапов разработки стало обеспечение адаптивности сайта для корректного отображения на различных устройствах: мобильных телефонах, планшетах и настольных компьютерах.

Для создания адаптивного дизайна были использованы CSS-медиазапросы, которые позволяют изменять стили в зависимости от размеров экрана:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 14 – Процесс адаптации

Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения

В рамках выполнения проекта было произведено тестирование функционала регистрации, авторизации и присвоение роли пользователя методом черного ящика.

Тестирование чёрного ящика или поведенческое тестирование – стратегия (метод) тестирования функционального поведения объекта (программы, системы) с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве (коде) тестируемого объекта. В таблице 1 представлены тестовые сценарии для проведения тестирования процесса регистрации в приложении.

Таблица 1 – Тестирование методом черного ящика метода регистрации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № Теста | Входные данные | Ожидаемый результат | Фактический результат | Примечание | Статус |
| 1 | userName = "",  password = "" | Должно появиться сообщение: "Пожалуйста, введите логин и пароль." | Сообщение: "Пожалуйста, введите логин и пароль." | Проверяется обработка  пустых  значений  логина и  пароля | Успешно завершен |
| 2 | userName = "a",  password = "password123" | Должно появиться сообщение: "Логин слишком короткий. Минимальная длина логина - 4 символа." | Сообщение: "Логин слишком короткий. Минимальная длина логина - 4 символа." | Проверяется минимальная  длина  логина | Успешно завершен |
| 3 | userName = "averylongusernameex  ceedingthelimit",  password = "password123" | Должно появиться сообщение: "Логин слишком длинный. Максимальная длина логина - 30 символов." | Сообщение: "Логин слишком длинный. Максимальная длина логина - 30 символов." | Проверяется максимальная  длина  логина | Успешно завершен |
| 4 | userName = "user123", password = "pass" | Должно появиться сообщение: "Пароль слишком короткий. Минимальная длина пароля - 8 символов." | Сообщение: "Пароль слишком короткий. Минимальная длина пароля - 8 символов." | Проверяется минимальная  длина  пароля | Успешно завершен |
| 5 | userName = "user123", password = "averylongpasswordexc  eedingthecharacterlim  itfortestingpurposes123" | Должно появиться сообщение: "Пароль слишком длинный. Максимальная длина пароля - 50 символов." | Сообщение: "Пароль слишком длинный. Максимальная длина пароля - 50 символов." | Проверяется максимальная  длина  пароля | Успешно завершен |
| 6 | userName =  "user123",  password =  "password123" | Должно появиться сообщение: "Пожалуйста, введите фамилию, имя и отчество." | Сообщение: "Пожалуйста, введите  фамилию,  имя и отчество." | Проверяется ввод  неполных данных  пользователя | Успешно завершен |
| 7 | userName = "user123", password = "password123", surname = "Иванов", name = "", patronymic = "Иванович" | Должно появиться сообщение: "Пожалуйста, введите фамилию, имя и отчество." | Сообщение: "Пожалуйста, введите  фамилию,  имя и отчество." | Проверяется ввод  пустого  имени | Успешно завершен |
| 8 | userName = "user123", password = "password123", surname = "Ив@нов", name = "Иван", patronymic = "Иванович" | Должно появиться сообщение: "Недопустимые символы в фамилии, имени или отчестве." | Сообщение: "Недопустимые  символы  в фамилии,  имени или  отчестве." | Проверяется валидация  на недопустимые  символы | Успешно завершен |
| 9 | userName = "newuser", password = "password123", surname = "Иванов", name = "Иван", patronymic = "Иванович" | Должно появиться сообщение: "Регистрация прошла успешно! Пожалуйста, подождите, пока администратор присвоит вам роль." | Сообщение:  "Регистрация  прошла успешно! Пожалуйста,  подождите, пока администратор  присвоит вам роль." | Проверяется успешное  создание нового  пользователя | Успешно завершен |

В таблице 2 представлены тестовые сценарии для проведения тестирования процесса авторизации в приложении.

Таблица 2 – Тестирование методом черного ящика метода авторизации

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № Теста | Входные данные | Ожидаемый результат | | Фактический результат | | Примечание | Статус |
| 1 | username = "", password = "" | Должно появиться сообщение: "Пожалуйста, введите логин и пароль." | | Сообщение: "Пожалуйста, введите логин и пароль." | | Проверяется обработка пустых значений логина и пароля | Успешно завершен |
| 2 | username = "user123", password = "" | Должно появиться сообщение: "Пожалуйста, введите логин и пароль." | | Сообщение: "Пожалуйста, введите логин и пароль." | | Проверяется, что пароль не может быть пустым | Успешно завершен |
| 3 | username = "", password = "password123" | Должно появиться сообщение: "Пожалуйста, введите логин и пароль." | | Сообщение: "Пожалуйста, введите логин и пароль." | | Проверяется, что логин не может быть пустым | Успешно завершен |
| 4 | username = "invalidUser", password = "password123" | Должно появиться сообщение: "Ошибка: Неправильный логин или пароль." | Сообщение: "Ошибка: Неправильный логин или пароль." | | Проверяется обработка неверного логина | | Успешно завершен |
| 5 | username = "user123", password = "wrongPassword" | Должно появиться сообщение: "Ошибка: Неправильный логин или пароль." | Сообщение: "Ошибка: Неправильный логин или пароль." | | Проверяется обработка неверного пароля | | Успешно завершен |
| 6 | username = "user123", password = "password123" (без присвоенной роли) | Должно появиться сообщение: "Ошибка: Учетная запись не имеет присвоенной роли." | Сообщение: "Ошибка: Учетная запись не имеет присвоенной роли." | | Проверяется ситуация отсутствия роли у пользователя | | Успешно завершен |

UNIT-тесты (или модульные тесты) являются частью практики разработки программного обеспечения, называемой тестированием программного обеспечения. Они представляют собой специальные фрагменты кода, которые проверяют отдельные компоненты программы, называемые "юнитами", на соответствие ожидаемому поведению. Они нужны для таких вещей, как:

* Обеспечение качества кода: UNIT-тесты помогают выявлять ошибки в коде на ранних этапах разработки, что позволяет их исправить до того, как они станут серьезными проблемами в реальной эксплуатации.
* Документация кода: Юнит-тесты могут служить в качестве спецификации того, как код должен работать. Путем изучения тестов другие разработчики могут быстро понять, как должен вести себя код.
* Сопровождение кода: Юнит-тесты делают код более устойчивым к изменениям. Если в процессе рефакторинга или добавления новой функциональности что-то ломается, UNIT-тесты быстро покажут, что именно пошло не так.
* Доверие к коду: Еще одним важным аспектом юнит-тестов является то, что они помогают строить доверие к коду. Если каждый компонент имеет набор тестов, подтверждающих его правильную работу, разработчики и пользователи могут быть уверены в его надежности.

Были реализованы три UNIT-теста, проверяющие работоспособность конструкций условий исключения в коде регистрации

После, необходимо запустить обозреватель тестов и запустить нужные нам методы проверок, процесс, который представлен на рисунке 15.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 15 – Результат выполненного тестирования

* RegisterUser\_ValidCredentials\_ReturnsTrue: Проверяет, что метод возвращает true, если логин и пароль валидны.
* RegisterUser\_NullUserName\_ReturnsFalse: Проверяет, что метод возвращает false, если логин равен null.
* RegisterUser\_NullPassword\_ReturnsFalse: Проверяет, что метод возвращает false, если пароль равен null.
* RegisterUser\_UserNameTooLong\_ReturnsFalse: Проверяет, что метод возвращает false, если логин слишком длинный.
* RegisterUser\_UserNameTooShort\_ReturnsFalse: Проверяет, что метод возвращает false, если логин слишком короткий.
* RegisterUser\_PasswordTooLong\_ReturnsFalse: Проверяет, что метод возвращает false, если пароль слишком длинный.
* RegisterUser\_PasswordTooShort\_ReturnsFalse: Проверяет, что метод возвращает false, если пароль слишком короткий.

Сопровождение информационной системы

Обучающая документация для пользователей – это набор информационных материалов, созданных для помощи пользователям в освоении и использовании вашего приложения. Она предназначена для того, чтобы облегчить процесс ознакомления с программным обеспечением, упростить его использование и максимально эффективно использовать его функционал.

Здесь должны быть описаны этапы по запуску приложения, первичным действиям в нем и по рекомендациям, как для администраторов, так и других пользователей.

## **2.3. Реализация информационной безопасности**

В данном разделе рассматриваются основные меры, принятые для обеспечения информационной безопасности в программном продукте, включая хеширование паролей и т.д.

Хеширование паролей обеспечивает защиту от несанкционированного доступа к учетным записям пользователей. Даже в случае компрометации базы данных злоумышленнику будет сложно восстановить исходные пароли. В приложении было реализовано хеширование.

## **Разработка веб-модуля для взаимодействия с API**

Разработка требований к функциям сайта

На сайте музыкального магазина реализованы следующие функции:

* Строка поиска: Удобный инструмент для быстрого нахождения товаров по названию или категории.
* Корзина: Система управления заказами, позволяющая добавлять товары, изменять их количество и удалять позиции.
* Форма регистрации и авторизации: Позволяет пользователям создавать личные кабинеты, сохранять заказы и отслеживать их статус.
* Форма обратной связи: Возможность для пользователей задать вопрос или оставить отзыв о товаре.
* Фильтры и сортировка товаров: Упрощают выбор подходящих товаров по характеристикам (цена, бренд, тип инструмента).

Эти функции обеспечивают удобство работы с сайтом, упрощают процесс покупки и делают взаимодействие пользователя с платформой максимально комфортным.

Разработка прототипов пользовательских интерфейсов системы

Для разработки музыкального магазина "Музыкальный рай" был создан макет в Figma. На этапе создания макета определялась структура всех страниц, включая расположение блоков, меню, элементов навигации и основных компонентов сайта. Макет позволил визуализировать общий вид веб-приложения до начала разработки, учесть удобство взаимодействия пользователей и проработать адаптивность для различных устройств.

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 16 – Макет страницы «Мессенджер»

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 17 –Макет страницы «Профиль»

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 18 –Макет страницы «Каталог»

Изображение выглядит как текст, снимок экрана, Параллельный, Шрифт

Автоматически созданное описание

Рисунок 19 - Макет страницы «Товар»

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 20 - Рисунок 14 — Макет страницы «Главная»

После создания макета был разработан дизайн страниц, в котором учитывались цветовая палитра, типографика и стилистика, соответствующие тематике музыкального магазина. В дизайне проработаны такие элементы, как кнопки, карточки товаров, фильтры, и другие пользовательские элементы. Figma предоставила удобные инструменты для выбора шрифтов, подбора цветов и создания интерактивных прототипов, что помогло добиться современного и эстетически привлекательного вида сайта.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 21 –Дизайн страницы «Профиль»

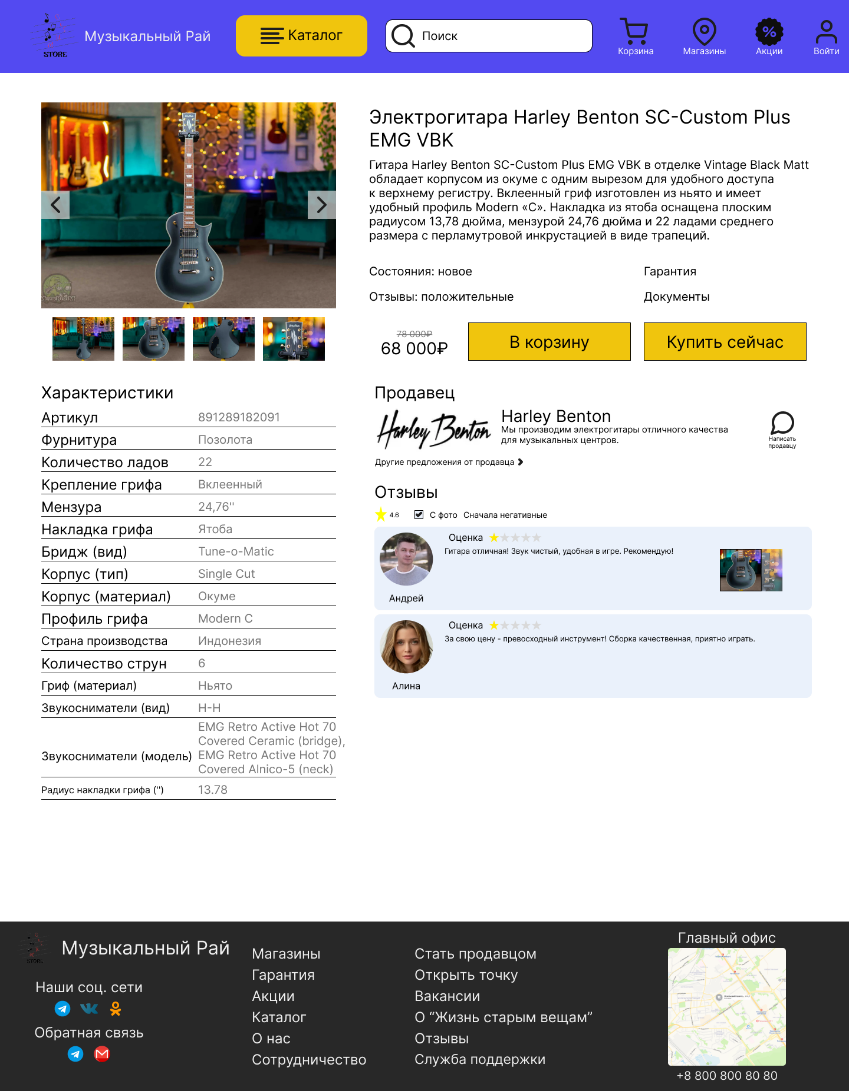


Рисунок 22 – Дизайн страницы «Товар»

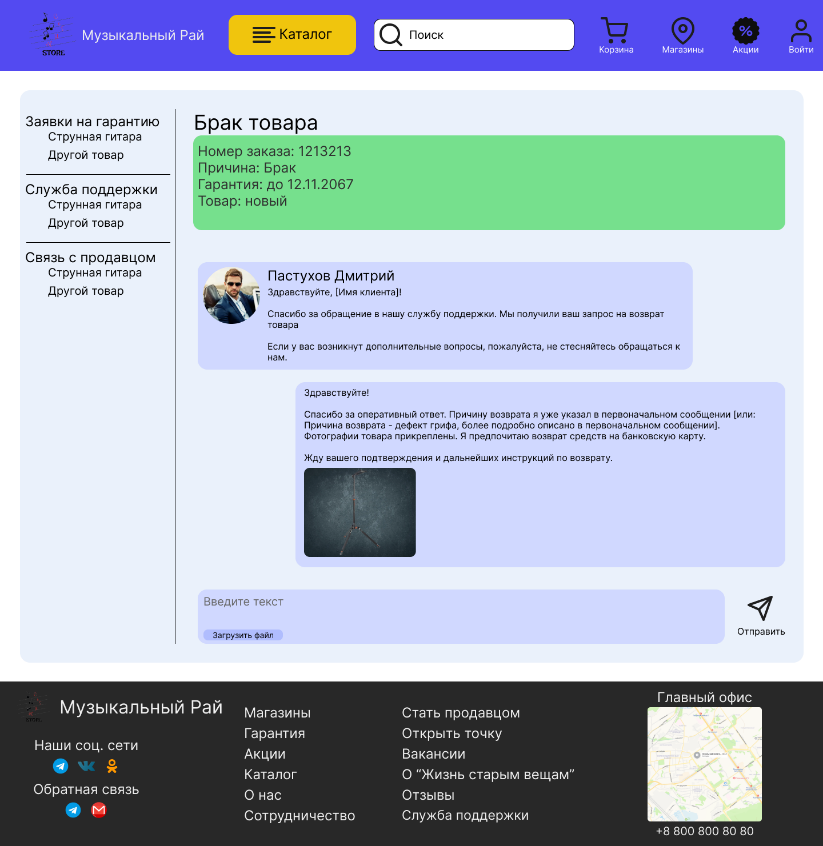


Рисунок 23 – Дизайн страницы «Мессенджер»

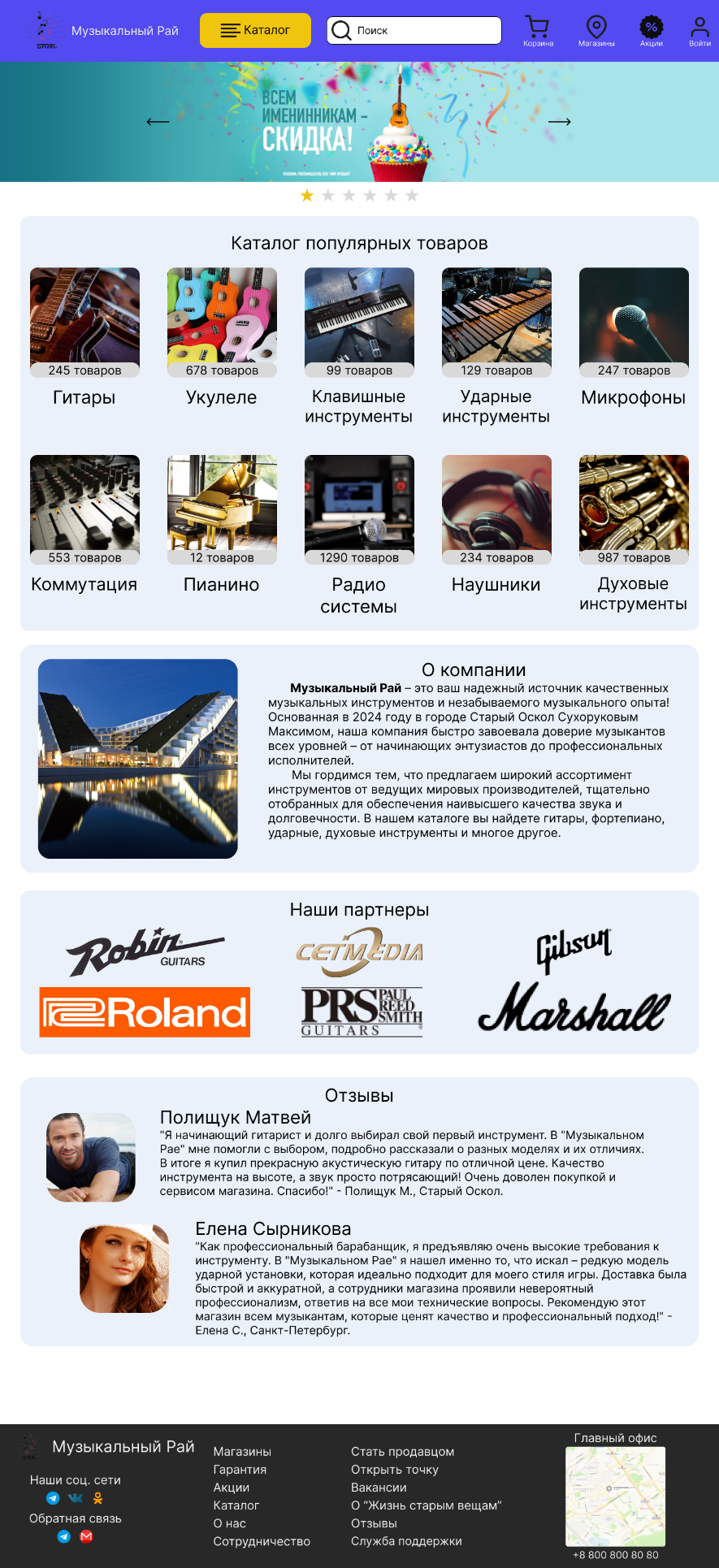


Рисунок 24 – Дизайн страницы «Главная»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Веб-сайт

Автоматически созданное описание

Рисунок 25 – Дизайн страницы «Каталог»

## **ЗАКЛЮЧЕНИЕ**

Актуальность разработки заключалась в стремлении современных компаний повысить эффективность продаж, расширить клиентскую базу за счет выхода в онлайн-пространство, а также оптимизировать внутренние бизнес-процессы.

Целью данного проекта являлось проектирование и разработка веб-приложения музыкального магазина, которое обеспечивает удобный и надежный способ покупки музыкальных инструментов и оборудования, облегчая клиентам поиск товаров и оформление заказов.

Проектируемая система включает основные функции по каталогизации товаров, управлению заказами, регистрации пользователей и взаимодействию с клиентами, что позволило минимизировать типичные ошибки ручного управления магазином и повысить удовлетворенность клиентов.

Предметом исследования данного проекта являлось разработанное веб-приложение музыкального магазина.

Объектом исследования являлся процесс онлайн-продаж и управления ассортиментом в розничной торговле.

Конкурентоспособность проекта обеспечивается его ориентацией на потребности современных покупателей, удобным пользовательским интерфейсом, а также возможностью масштабирования для адаптации под растущие потребности бизнеса.

Практическая значимость проекта заключается в том, что внедрение такого веб-приложения позволяет компании значительно увеличить продажи, расширить целевую аудиторию и оптимизировать внутренние процессы, связанные с управлением товарооборотом и клиентским сервисом.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Артюхина Д.Д., Коренькова Т.Н., Назарова О.И., Порядок подготовки, оформления и защиты курсовых и дипломных работ: учебно-методическое пособие – Старый Оскол: СТИ НИТУ «МИСиС», 2022. – 59 с

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | | Ресурс | | QR-код | |
| 2 | | Васильков, А. В. Безопасность и управление доступом в информационных системах | | Изображение выглядит как шаблон, прямоугольный, Симметрия, искусство  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 3 | | Гвоздева, В. А. Основы построения автоматизированных информационных систем | | Изображение выглядит как шаблон, прямоугольный, пиксель, кроссворд  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 4 | | Гришина, Н. В. Основы информационной безопасности предприятия | | Изображение выглядит как шаблон, прямоугольный, Симметрия, пиксель  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 5 | | Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности | | Изображение выглядит как шаблон, прямоугольный, пиксель  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 6 | | Черников, Б. В. Управление качеством программного обеспечения | | Изображение выглядит как шаблон, прямоугольный, пиксель  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 7 | | Мартишин, С. А. Базы данных. Практическое применение СУБД SQL- и NoSQL-типа для проектирования информационных систем | | Изображение выглядит как шаблон, прямоугольный, Симметрия, пиксель  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 8 | | Фиайли, К. SQL. Руководство для использования с любыми SQL СУБД | | Изображение выглядит как шаблон, прямоугольный, пиксель  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 9 | | Баланов, А. Н. Комплексная информационная безопасность | | Изображение выглядит как шаблон, прямоугольный, искусство, Прямоугольник  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 10 | Лубенцов, А. В. Комплексные системы безопасности | | Изображение выглядит как шаблон, прямоугольный, пиксель, дизайн  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 11 | Metanit | | Изображение выглядит как шаблон, прямоугольный, искусство, Прямоугольник  Контент, сгенерированный ИИ, может содержать ошибки. | |
| 12 | Форум программистов и сисадминов Киберфорум | | Изображение выглядит как шаблон, прямоугольный, искусство, Симметрия  Контент, сгенерированный ИИ, может содержать ошибки. | |

# 

# **ПРИЛОЖЕНИЕ А**

**Учетные записи для доступа к программному решению**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Наименование роли** | **Логин** | **Пароль** |
| 1 | Администратор | VV1310 | TrustMe14\_55 |
| 2 | Пользователь | BYNikKis52 | Sovetchick-S0chBlt |

# **ПРИЛОЖЕНИЕ Б**

**Листинг программного решения**

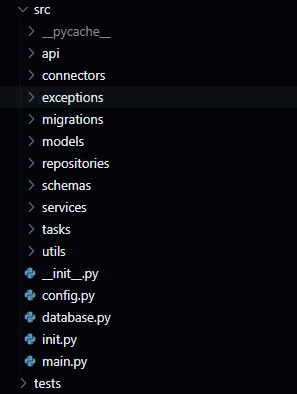
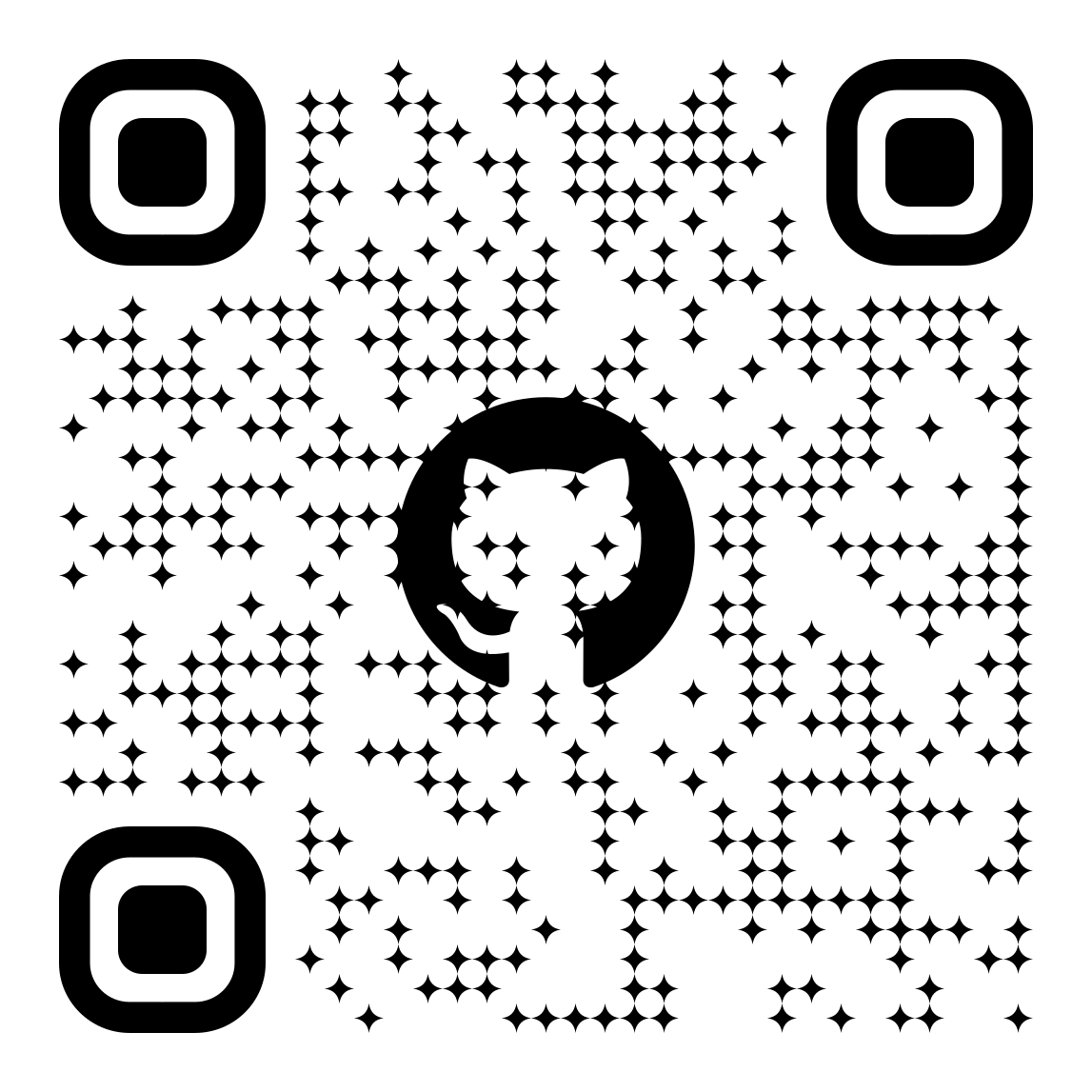


Рисунок 26 – Обозреватель проекта

**QR код для доступа к репозиторию программного решения**



# **ПРИЛОЖЕНИЕ В**

**АННОТАЦИЯ**

Проектирование веб-приложения для интернет-магазина музыкальных инструментов.

**Авторы:**

**Иванов И.И.**

**Руководитель:**

**Коренев А.М.**

**ОПК СТИ НИТУ «МИСИС»**

**Введение**

С ростом популярности онлайн-покупок и увеличением конкуренции на рынке музыкальных инструментов возникает потребность в современных веб-решениях, способных обеспечить удобный процесс оформления заказов и эффективное управление каталогом товаров. Автоматизация процессов онлайн-продаж становится ключевым шагом для повышения конкурентоспособности магазина, улучшения взаимодействия с клиентами и увеличения объема продаж.

Внедрение веб-приложения для музыкального магазина позволяет упростить процесс выбора товаров, оформления заказов и управления ассортиментом. Такое приложение систематизирует данные о товарах и заказах, предоставляет удобный доступ к информации для клиентов и администраторов. Функции, такие как управление каталогом товаров, обработка заказов в реальном времени, интеграция с системами оплаты и доставки, способствуют более эффективному ведению бизнеса и повышению уровня удовлетворенности клиентов.

**Цели и задачи системы**

Для поставленной цели необходимо выполнить следующие задачи:

* спроектировать веб-приложение на основе анализа предметной области и исходных данных;
* разработать ER-диаграмму базы данных;
* реализовать информационную безопасность разрабатываемой системы;
* разработать обучающую документацию для пользователей разрабатываемой системы.

Система должна состоять из нескольких модулей, доступ к которым будет определяться типом учетной записи (клиент, администратор).

Общее назначение разрабатываемых модулей:

* просмотр каталога товаров с фильтрацией и поиском;
* добавление/удаление/редактирование данных о товарах;
* управление процессом оформления и обработки заказов;
* ведение базы клиентов и истории заказов.

Объектом исследования является система автоматизации онлайн-продаж музыкальных инструментов, охватывающая аспекты управления каталогом товаров и обработки заказов.

Предметом исследования выступает веб-приложение, предназначенное для автоматизированного управления процессами онлайн-торговли, включая функционал для работы с каталогом товаров, обработки заказов и взаимодействия с клиентами.

Целью исследования является разработка эффективного веб-приложения, которое позволит музыкальному магазину автоматизировать процессы онлайн-продаж, улучшить взаимодействие с клиентами и повысить качество обслуживания. Результатом курсовой работы будет являться готовый программный продукт и пояснительная записка.

Результатом проекта будет являться готовое веб-приложение и пояснительная записка.

**Архитектура системы**

Информационная система реализована на основе клиент-серверной архитектуры. Клиентская часть представлена веб-интерфейсом, обеспечивающим удобный доступ для клиентов и администраторов. Серверная часть отвечает за обработку запросов, управление базой данных, обеспечение безопасности.

**Компоненты системы:**

1. **Интерфейс пользователя**: должен быть интуитивно понятным и доступным для всех категорий пользователей. Включает каталог товаров, корзину покупок, личный кабинет клиента.
2. **Административная панель**: для сотрудников магазина, где они могут управлять каталогом товаров, обрабатывать заказы и анализировать продажи.
3. **База данных**: хранит информацию о товарах, заказах, клиентах и категориях. Используется реляционная база данных для обеспечения целостности данных и возможности выполнения сложных запросов.
4. **Механизмы уведомлений:** система отправляет уведомления клиентам о статусе их заказов, а также напоминания администраторам о новых заказах.

**Реализация и тестирование**

Процесс разработки системы можно разбить на несколько этапов:

1. **Сбор требований**: взаимодействие с владельцами магазина и потенциальными клиентами для определения функционала системы.
2. **Проектирование**: Создание архитектуры системы и проектирование пользовательских интерфейсов.
3. **Разработка**: программирование компонентов системы с использованием современных веб-технологий и языков программирования.
4. **Тестирование**: проведение функционального тестирования системы для выявления и исправления ошибок, а также тестирование на удобство использования.
5. **Внедрение**: развертывание системы на хостинге, обучение администраторов и запуск в эксплуатацию, сбор обратной связи.

**Заключение**

Разработанное веб-приложение позволит оптимизировать процессы онлайн-продаж, повысить качество обслуживания клиентов, улучшить эффективность работы музыкального магазина и обеспечить конкурентоспособность на рынке.