**Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Защита информации (ИУ10)

ПРИЛОЖЕНИЕ

***К ТЗ ДЛЯ SCPC***

*Описание Используемых Библиотек*

ПО КУРСУ: «Языки Программирования»

Студент ИУ10-44 С.Р. Катлюшкин (Группа) (Подпись, дата) (И.О. Фамилия)

Студент ИУ10-44 М.А. Янченко (Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель К.А. Буркацкий (Подпись, дата) (И.О. Фамилия)

Москва, 2025

**Содержание**

Оглавление

[1. Введение 3](#_Toc192633175)

[2. Описание библиотек 4](#_Toc192633176)

[2.1. ​Библиотеки для работы с HTTP и сетевыми запросами 4](#_Toc192633177)

[2.1.1. Библиотека Requests 4](#_Toc192633178)

[2.1.2. Библиотека AIOHTTP 4](#_Toc192633179)

[2.2. ​Библиотеки для работы с базами данных 4](#_Toc192633180)

[2.2.1. Библиотека SQLAlchemy 4](#_Toc192633181)

[2.2.2. Библиотека Alembic 4](#_Toc192633182)

[2.3. ​Библиотеки для анализа данных и NLP 5](#_Toc192633183)

[2.3.1. Библиотека spaCy 5](#_Toc192633184)

[2.3.2. Библиотека Gensim 5](#_Toc192633185)

[2.4. ​Библиотеки для машинного обучения 5](#_Toc192633186)

[2.4.1. Библиотека Scikit-learn 5](#_Toc192633187)

[2.5. Библиотеки для аутентификации и шифрования 5](#_Toc192633188)

[2.5.1. Библиотека PyJWT 5](#_Toc192633189)

[2.6. ​Библиотеки для валидации данных 6](#_Toc192633190)

[2.6.1. Библиотека Pydantic 6](#_Toc192633191)

[2.7. ​Библиотеки для клиентской части 6](#_Toc192633192)

[2.7.1. Библиотека threading 6](#_Toc192633193)

[2.7.2. Библиотека PyQt5 6](#_Toc192633194)

[3. Будущие добавления 6](#_Toc192633195)

# Введение

Настоящий документ представляет собой описание библиотек, использованных при разработке клиент-серверного приложения и веб-сайта, функционирующего в сети Интернет. Документ составлен в соответствии с требованиями к технической документации и предназначен для ознакомления разработчиков, тестировщиков и других заинтересованных лиц с используемыми программными компонентами.

Цель документа — предоставить полную информацию о выборе, назначении и функциональных возможностях библиотек, а также описать возможные направления для их расширения и модификации в будущем. Документ содержит описание каждой библиотеки, включая ее версию, назначение, причины выбора и примеры использования.

Данная документация является частью проектной документации и может быть использована для поддержки, развития и модернизации программного обеспечения.

# Описание библиотек

## ​Библиотеки для работы с HTTP и сетевыми запросами

### Библиотека Requests

* Назначение:  
  Выполнение HTTP-запросов к внешним API (Genius API, Lyrics.ovh API) для сбора текстов песен. Используется в клиентской и серверной частях для получения данных.
* Причины выбора:
* Простота интеграции с RESTful API.
* Поддержка сессий, прокси и обработки ошибок.
* Широкая документация и активное сообщество.
* Особенности:
* Используется для периодического обновления базы данных текстов песен.
* Обработка кодов ответов (200, 404 и т.д.) для стабильности системы.

### Библиотека AIOHTTP

* Назначение:  
  Асинхронная обработка HTTP-запросов на сервере для обеспечения многопоточности (≥10 одновременных запросов).
* Причины выбора:
* Высокая производительность при асинхронном выполнении.
* Совместимость с FastAPI для работы с эндпоинтами /get\_lyrics, /find\_similar, /generate\_report.
* Особенности:
* Используется в серверной части для минимизации времени отклика (≤500 мс).
* Поддержка WebSocket не требуется, так как веб-интерфейс исключен из ТЗ.

## ​Библиотеки для работы с базами данных

### Библиотека SQLAlchemy

* Назначение:  
  ORM (Object-Relational Mapping) для взаимодействия с SQLite. Управление таблицей lyrics, хранение текстов песен, результатов анализа и метаданных.
* Причины выбора:
* Поддержка схемы БД из ТЗ: поля id, product\_id, user\_id, review\_text, sentiment, keywords, created\_at.
* Гибкость в работе с SQL-запросами через объекты Python.
* Особенности:
* Интеграция с Alembic для миграций.
* Использование пула соединений для оптимизации производительности.

### Библиотека Alembic

* Назначение:  
  Управление миграциями базы данных. Автоматическое применение изменений в структуре БД (добавление столбцов, индексов и т.д.).
* Причины выбора:
* Совместимость с SQLAlchemy.
* Упрощение процесса обновления схемы БД без ручного вмешательства.
* Особенности:
* Миграции выполняются при каждом обновлении серверной части.
* Резервное копирование данных перед применением миграций.

## ​Библиотеки для анализа данных и NLP

### Библиотека spaCy

* Назначение:  
  Обработка естественного языка (NLP): токенизация, лемматизация, извлечение ключевых слов из текстов песен.
* Причины выбора:
* Готовые предобученные модели для английского языка (например, en\_core\_web\_lg).
* Высокая скорость обработки больших объемов текста.
* Особенности:
* Используется для подготовки данных перед векторизацией.
* Интеграция с Gensim для семантического анализа.

### Библиотека Gensim

* Назначение:  
  Построение векторных представлений текстов (Doc2Vec, Word2Vec) для вычисления семантической схожести между песнями.
* Причины выбора:
* Эффективность в задачах анализа текстовой схожести.
* Поддержка распределенного обучения на больших корпусах данных.
* Особенности:
* Обучение моделей на собранных текстах песен.
* Экспорт векторов для использования в алгоритмах машинного обучения.

## ​Библиотеки для машинного обучения

### Библиотека Scikit-learn

* Назначение:  
  Реализация алгоритмов машинного обучения для поиска схожих треков: расчет косинусной схожести на основе TF-IDF.
* Причины выбора:
* Широкий набор инструментов для кластеризации и классификации.
* Интеграция с NLP-пайплайнами (spaCy, Gensim).
* Особенности:
* Использование TfidfVectorizer для преобразования текстов в числовые векторы.
* Оптимизация производительности за счет разреженных матриц.

## Библиотеки для аутентификации и шифрования

### Библиотека PyJWT

* Назначение:  
  Генерация и проверка JWT-токенов для аутентификации пользователей и защиты API.
* Причины выбора:
* Соответствие требованию безопасности из ТЗ (JWT-аутентификация).
* Простота интеграции с FastAPI.
* Особенности:
* Хранение секретного ключа в защищенном конфигурационном файле.
* Валидация токенов при каждом запросе к /find\_similar и /generate\_report.
  + 1. **Библиотека cryptography**
* Назначение:  
  Шифрование данных при передаче (TLS 1.3) и хранении (хеширование паролей с SHA-256).
* Причины выбора:
* Поддержка современных криптографических стандартов.
* Интеграция с FastAPI для HTTPS.
* Особенности:
* Использование симметричного шифрования для резервных копий БД.
* Генерация SSL-сертификатов для защиты API.

## ​Библиотеки для валидации данных

### Библиотека Pydantic

* Назначение:  
  Валидация входных данных API (например, параметров запроса к /get\_lyrics).
* Причины выбора:
* Автоматическая генерация ошибок при неверных данных.
* Совместимость с FastAPI для аннотации типов.
* Особенности:
* Использование моделей данных для строгой типизации.
* Поддержка пользовательских валидаторов (например, проверка формата product\_id).

## ​Библиотеки для клиентской части

### Библиотека threading

* Назначение:  
  Многопоточная обработка запросов в клиентском приложении для предотвращения зависания интерфейса.
* Причины выбора:
* Простота использования в сочетании с PyQt5.
* Поддержка асинхронных операций без блокировки основного потока.
* Особенности:
* Использование QThreadPool для управления потоками.
* Обработка ошибок в дочерних потоках через сигналы PyQt5.

### Библиотека PyQt5

* Назначение:  
  Создание десктопного приложения с графическим интерфейсом для взаимодействия с сервером.
* Причины выбора:
* Кроссплатформенность (поддержка Windows, Linux, macOS).
* Возможность реализации многопоточности через QThread.
* Особенности:
* Интерфейс включает поля для ввода запроса, таблицу результатов и кнопку экспорта в CSV/JSON.
* Интеграция с requests для отправки API-запросов.

# Будущие добавления

Планируется расширение функционала с использованием:

* **Celery** — для асинхронной обработки задач (например, периодический сбор текстов).
* **Docker** — контейнеризация серверной части для упрощения развертывания.
* **Elasticsearch** — улучшение скорости и точности поиска схожих треков.
* **NLTK** — дополнительные методы NLP (например, анализ тональности текста).