

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Кафедра информационных систем

КУРСОВАЯ РАБОТА

Тема: Разработка веб-приложения «Генерация репа в стиле Славы КПСС»

Работу выполнил студент: Сливкин Артём Сергеевич **группы** М3315
(фамилия, имя, отчество) (номер группы)

Руководитель Андреев Н. В., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")
(фамилия, имя, отчество)

Работа защищена " _____ " _____ 2025 г. с оценкой _____

Подписи членов комиссии: _____

**САНКТ-ПЕТЕРБУРГ
2025**

УНИВЕРСИТЕТ ИТМО

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

Студент	Сливкин А.С
	(Фамилия, И., О.)
Факультет	Информационных технологий и программирования
Кафедра	Информационных систем
Направление (специальность)	Группа М3315
	Разработка программного обеспечения / Software Engineering
Руководитель	Андреев Н. В., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")
	(Фамилия, И.О., должность, ученое звание, степень)
Дисциплина	Web-программирование
Наименование темы	Разработка веб-приложения «Генерация репа в стиле Славы КПСС»

Задание	Разработать веб-приложение, проанализировать и смоделировать процессы, средства автоматизации, спроектировать архитектуру информационной системы и разработать пользовательский интерфейс
---------	---

Краткие методические указания	В ходе выполнения работы необходимо:
<p>1. Описать структуру разрабатываемого приложения в целом, выделить модули и конкретизировать задачи для каждого модуля (авторизация, работа с базой данных и т.д.), а также указать основные информационные объекты, которые используются в каждом модуле (модели данных, сервисы, и прочие подобные сущности фреймворка). Выбрать методологию и в соответствии с ее правилами сформировать набор диаграмм, дающих формальное описание. Сделать выводы о функциональных требованиях к средствам автоматизации со стороны смоделированных процессов. При наличии возможность описать нефункциональные требования</p>	
<p>2. Описать типовые функциональные возможности классов информационных систем, применяющихся для автоматизации определенных на предыдущем этапе процессов, обосновать выбор конкретного набора информационных систем, детально описать их функциональные возможности и сопоставить их с функциональными требованиями, полученными на предыдущем этапе.</p>	
<p>3. Представить функциональную и информационную архитектуры ИС, включающие все выбранные на предыдущем этапе программные средства автоматизации. Функциональная архитектура представляется как распределение операций смоделированных процессов по функциональным компонентам отдельных программных средств. В случае взаимосвязанных процессов или распределения операций одного процесса по нескольким средствам автоматизации указывается передача данных между функциональными компонентами соответствующих модулей. Информационная архитектура представляется в виде сопоставления информационных объектов, выделенных на первом этапе с информационными объектами, реализованными в выбранных средствах автоматизации. Описать интеграцию систем на уровне совместного использования преобразования данных информационных объектов, обеспечение целостности данных и синхронизации выполняемых над ними операций.</p>	

Содержание пояснительной записки

1. Определение основных понятий
 2. Анализ и моделирование процессов
 3. Анализ средств автоматизации процессов
 4. Проектирование архитектуры ИС
 5. Реализация пользовательского интерфейса
-

Рекомендуемая литература

1. Мартин Фаулер - Архитектура корпоративных программных приложений. Издательский дом "Вильямс". 2006 г.
 2. Флэнаган, Дэвид. JavaScript. Полное руководство, 7-е изд. : Пер. с англ. — СПб. : ООО "Диалектика", 2021. — 720 с .
 3. Янг А., Мек Б., Кантелон М. Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.
 4. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. 2-е издание. — СПб.: Питер, 2021. — 336 с.
 5. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 04.05.2021.
-

Руководитель

Подпись, дата

Студент

Подпись, дата

УНИВЕРСИТЕТ ИТМО

АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

Студент Сливкин А.С

(Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М3315

Направление (специальность) 09.03.02 «Разработка программного обеспечения / Software Engineering»

Руководитель Андреев Н. В., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")

(Фамилия, И.О., место работы, должность, ученое звание, степень)

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения «Генерация репа в стиле Славы КПСС»

ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

1. Цель и задачи работы

☐ Предложены студентом

☐ Сформулированы при участии студента

☐ Определены руководителем

Цель: Разработать веб-приложение.

Задачи:

1) Сформировать функциональные требования к веб-приложению.

2) Проанализировать функциональные возможности и выбрать набор средств автоматизации.

3) Сформировать функциональную и информационную архитектуру решения.

2. Характер работы

☐ Расчет

☐ Конструирование

☐ Моделирование

☐ Другое,

4. Содержание работы

1) Определение основных понятий

2) Анализ и моделирование процессов

3) Анализ средств автоматизации процессов

4) Проектирование архитектуры ИС

5) Реализация пользовательского интерфейса

5. Выводы

Студент _____

(подпись)

Руководитель _____

(подпись)

«_____» _____ 2025 г.

УНИВЕРСИТЕТ ИТМО

О Т З Ы В РУКОВОДИТЕЛЯ

о выполнении курсового проекта (работы)

Студент Сливкин А.С.

(Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М3315

Направление (специальность) 09.03.02 «Разработка программного обеспечения / Software Engineering»

Руководитель Андреев Н. В., Университет ИТМО, преподаватель (квалификационная категория "преподаватель практики")

(Фамилия, И.О., место работы, должность, ученое звание, степень)

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения «Генерация репа в стиле Славы КПСС»

ОЦЕНКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

№ п/п	Показатели	Оценка			
		5	4	3	0*
1.	Способность к работе с литературными источниками, справочной литературой, Интернет-ресурсами и т. п.				
2.	Использование иностранных источников				
3.	Способность к анализу и обобщению информационного материала				
4.	Владение базовыми знаниями в профессиональной области				
5.	Владение базовыми знаниями в смежных областях				
6.	Владение навыками решения технических задач				
7.	Способность применять знания на практике				
8.	Уровень и корректность использования в работе методов численного моделирования, инженерных расчетов и статистической обработки данных				
9.	Владение навыками использования современных пакетов компьютерных программ и технологий				
10.	Владение навыками оформления отчетных материалов с применением современных пакетов программ				
11.	Качество оформления пояснительной записки (общий уровень грамотности, стиль изложения, качество иллюстраций, корректность цитирования и пр.**)				
12.	Качество оформления презентации				
13.	Владение навыками публичного выступления и межперсональной коммуникации				
14.	Владение навыками планирования и управления временем при выполнении работы				
ИТОГОВАЯ ОЦЕНКА					

* - не оценивается (трудно оценить)

**Отмеченные
достоинства:** _____

**Отмеченные
недостатки:** _____

Заключение _____

Руководитель _____
(подпись)

Дата « _____ » _____ 2025 г.

Оглавление

Введение.....	8
Определения, обобщения и сокращения	9
Описание предметной области	10
Описание прикладного процесса.....	10
Формирование требований	10
Проектирование.....	12
Используемый стек технологий	12
Системная архитектура	12
Архитектура данных	13
Программная архитектура.....	15
Разработка	17
Реализация серверного API.....	17
Реализация пользовательского интерфейса	18
Заключение	24
Список использованной литературы	25

Введение

Объектом разработки является веб-приложение для генерации текстов в стиле рэп-исполнителя, реализованное на языке программирования Python с использованием фреймворка FastAPI — для серверной части приложения, и JavaScript — для клиентской части, без применения сторонних фреймворков. Выбранный фреймворк отличается высокой производительностью, лаконичностью и активной поддержкой сообщества. Целью работы является разработка веб-приложения и пользовательского интерфейса, анализ требования и моделирование процессов, средств автоматизации и архитектуры информационной системы.

В ходе работы были получены следующие результаты:

- Серверная часть системы, принимающая запросы.
- Клиентская часть системы, предоставляющая интерфейс пользователя.
- База данных для хранения информации о пользователях, файлах, а также служебной информации внутри системы.

Определения, обобщения и сокращения

Браузер – программное обеспечение для просмотра веб-страниц, обработки HTML, CSS и JavaScript. Используется конечным пользователем для взаимодействия с веб-приложением

Фреймворк – набор инструментов и библиотек, предоставляющий архитектурную основу для разработки программного обеспечения.

Роль «администратор» внутри приложения – просмотр всех пользователей.

Генеративная модель – языковая модель, способная создавать текст на основе заданного входа.

API (Application Programming Interface) – программный интерфейс приложения. Позволяет внешним приложениям взаимодействовать с функциональностью сервера.

FAISS (Facebook AI Similarity Search) — библиотека для эффективного поиска похожих векторов в больших объёмах данных. Используется для поиска похожих отрывков текста на основе эмбеддингов.

RAG (Retrieval-Augmented Generation) — подход генерации текста, при котором модель получает внешние текстовые фрагменты (документы) из базы знаний перед генерацией ответа, что повышает точность и релевантность результата.

Описание предметной области

Описание прикладного процесса

Разрабатываемое веб-приложение представляет собой интерактивную систему генерации текстов в стиле конкретного исполнителя, построенную на основе языковых моделей и RAG-системы. Приложение позволяет пользователю ввести начало текста, на основе которого система сгенерирует продолжение, стилизованное под выбранного исполнителя.

В отличие от классических систем управления контентом, где основной задачей является просмотр, редактирование и публикация информации на сайте, данное приложение ориентировано на генерацию уникального контента в диалоге с пользователем.

Формирование требований

В ходе анализа прикладного процесса был получен следующий список функциональных требований:

Модуль аутентификации пользователей должен иметь:

- Возможность регистрации новых пользователей с уникальным именем
- Возможность входа в систему с именем пользователя и пароля
- Возможность выхода из системы
- Система обновления токенов JWT

Модуль генерации текста должен иметь:

- Возможность генерации текста на основе пользовательского запроса
- Использование векторной базы данных для RAG-системы
- Интеграция с API OpenAI

Модуль работы с избраным должен иметь:

- Возможность сохранения сгенерированных текстов в раздел избранного
- Просмотр всех сохраненных текстов пользователя

Модуль веб-интерфейса должен иметь:

- Интуитивно понятный пользовательский интерфейс
- Отображение статуса процесса генерации в реальном времени
- Возможность переключения между страницами генератора и избранного

Нефункциональные требования

Разрабатываемая система не является публичной, поэтому основным требованием выступает её закрытость, путем обязательного прохождения процесса авторизации и аутентификации.

Так же должна быть возможность работы с приложением напрямую, через общие программные интерфейсы, описанные по спецификации OpenAPI версии не ниже 3.0

Проектирование

Используемый стек технологий

Решение создания именно веб-приложения обусловлено тем, что необходимо было обеспечить доступ к системе с любого устройства, в любое время. Веб-приложение решает этот вопрос, а также снимает вопрос обновлений на стороне клиента.

Проект использует стек стандартных технологий, характерный для большинства веб-приложений: HTML, CSS, Javascript.

В качестве веб-сервера используется FastAPI.

В качестве базы данных используется PostgreSQL, ввиду того, что данная СУБД является самой стабильной в данной связке. Для работы с данными была выбрана ORM SQLAlchemy.

В проекте используется свободная распределённая система управления версиями Git, хранилищем исходных кодов является крупнейший веб-сервис GitHub, а в качестве хостинга используется облачный сервис Render.com.

Системная архитектура

Архитектура системы основана на клиент-серверной модели. Клиентская часть представлена веб-браузером, который взаимодействует с сервером через протокол HTTPS. Сервер реализован на FastAPI и обрабатывает запросы пользователей, а также осуществляет аутентификацию и авторизацию.

Сервер взаимодействует с PostgreSQL через ODBC протокол для хранения информации о пользователях и их избранных текстах. Для генерации текстов используется интеграция с OpenAI API по протоколу HTTPS.

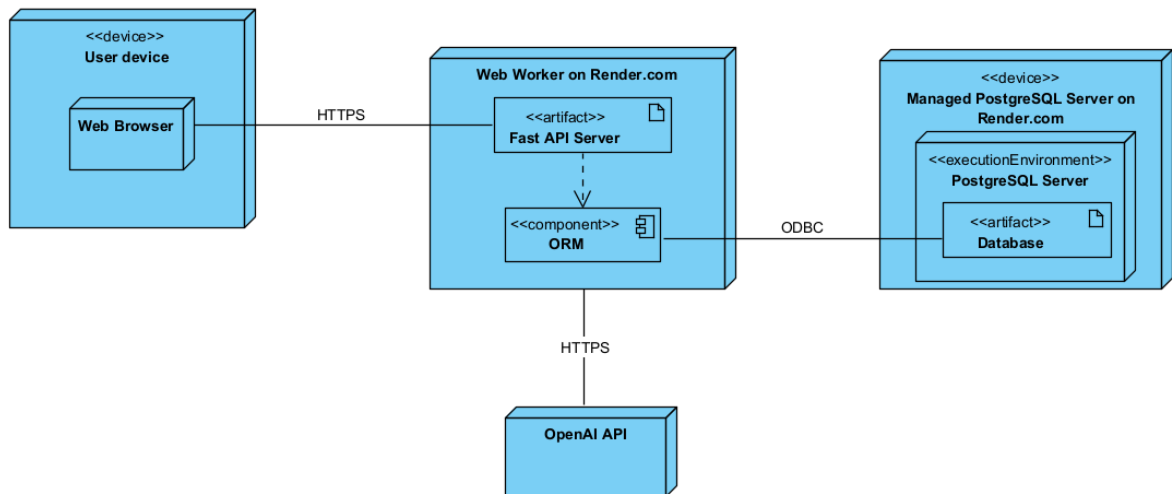


Рисунок 1. Системная архитектура приложения.

Архитектура данных

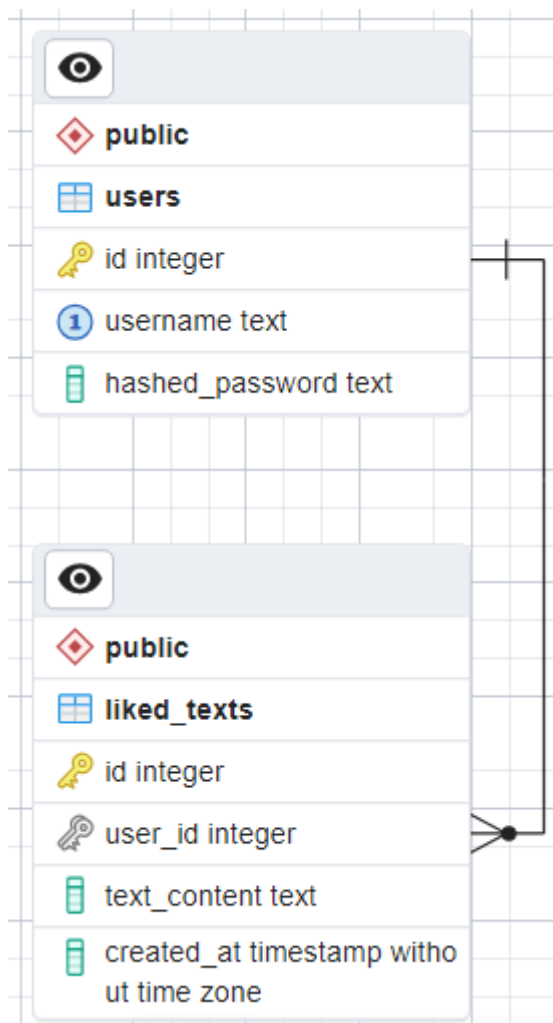


Рисунок 2. Схема таблиц базы данных

Users – Предоставляет информацию о пользователе

Liked_texts – Предоставляет информацию о понравившихся пользователю текстах

Программная архитектура

Таблица 1. Отношения модулей и классов

Название модуля	Функциональный компонент	Назначение компонента
Модуль аутентификации	register, login, logout	Регистрация, вход и выход пользователя в системе
Модуль генерации текста	generate	Генерация текста песни на основе пользовательского запроса
Модуль генерации текста	stream_generation	Потоковая генерация текста с отображением прогресса
Модуль избранного	like_text	Сохранение текста в избранное пользователя
Модуль веб-интерфейса	serve_frontend, generator_page, favorites_page	Отображение главной страницы приложения, страницы генератора и избранных текстов.

Таблица 2. Описание классов

Название компонента	Описание компонента
register	Эндпоинт для регистрации новых пользователей с проверкой на уникальность имени и хешированием паролей.
login	Эндпоинт для аутентификации пользователей

generate	Защищенный эндпоинт для генерации текста песни с помощью RAG и GPT-4. Требуется авторизация и визуализируется на странице генератора.
stream_generation	Эндпоинт для пошаговой генерации текста с отображением процесса в реальном времени с помощью SSE.
like_text	Эндпоинт для сохранения понравившихся текстов. Отображается в виде кнопки "Добавить в избранное" на странице генератора.

Разработка

Реализация серверного API

В качестве описания программного интерфейса был выбран инструмент, поддерживающий стандарт OAS 3.0 – Swagger. Далее представлена полученная документация API полученная автоматически по директивам, указанным в декораторах различных методов и структурах данных внутри разрабатываемой информационной системы.

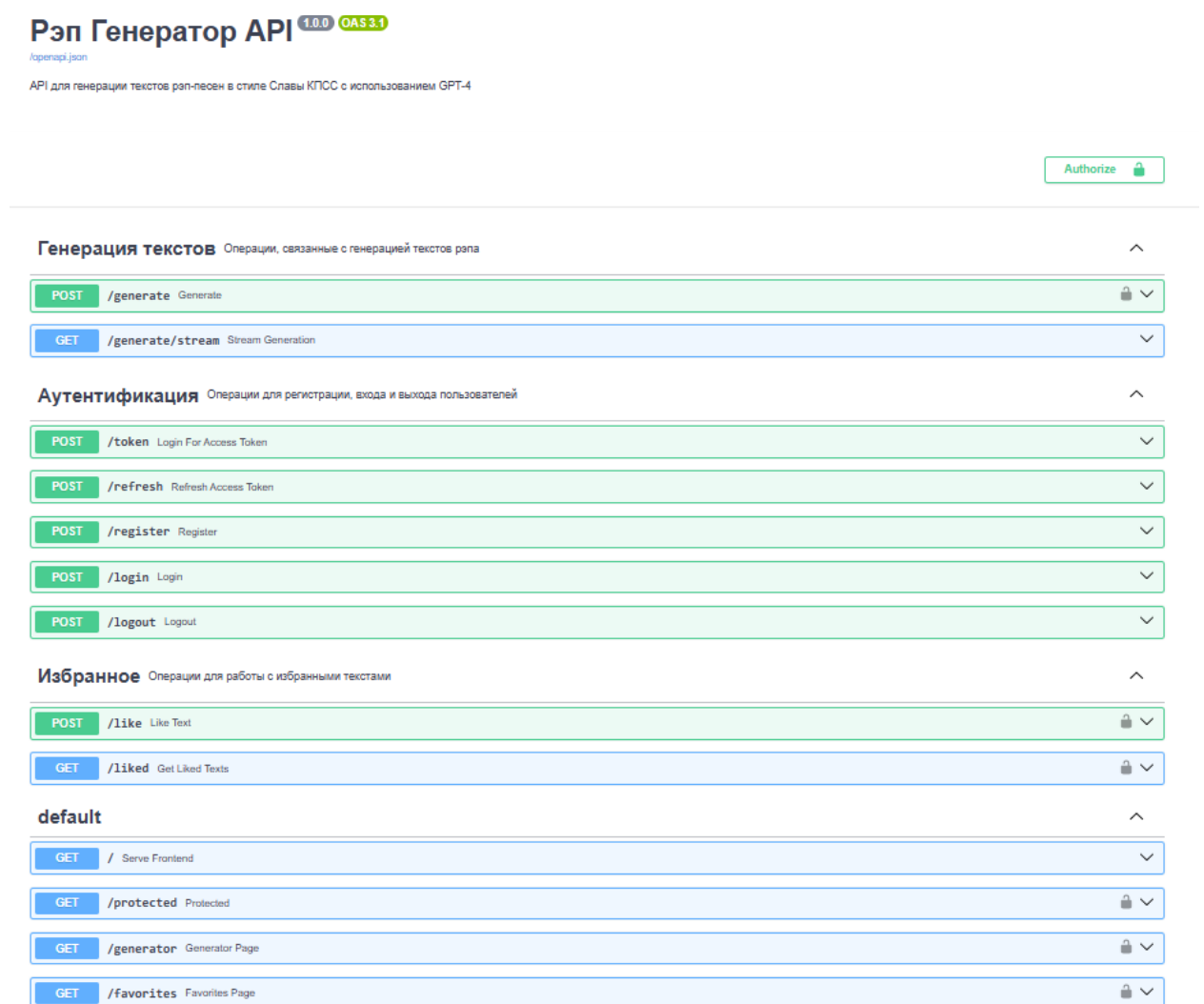
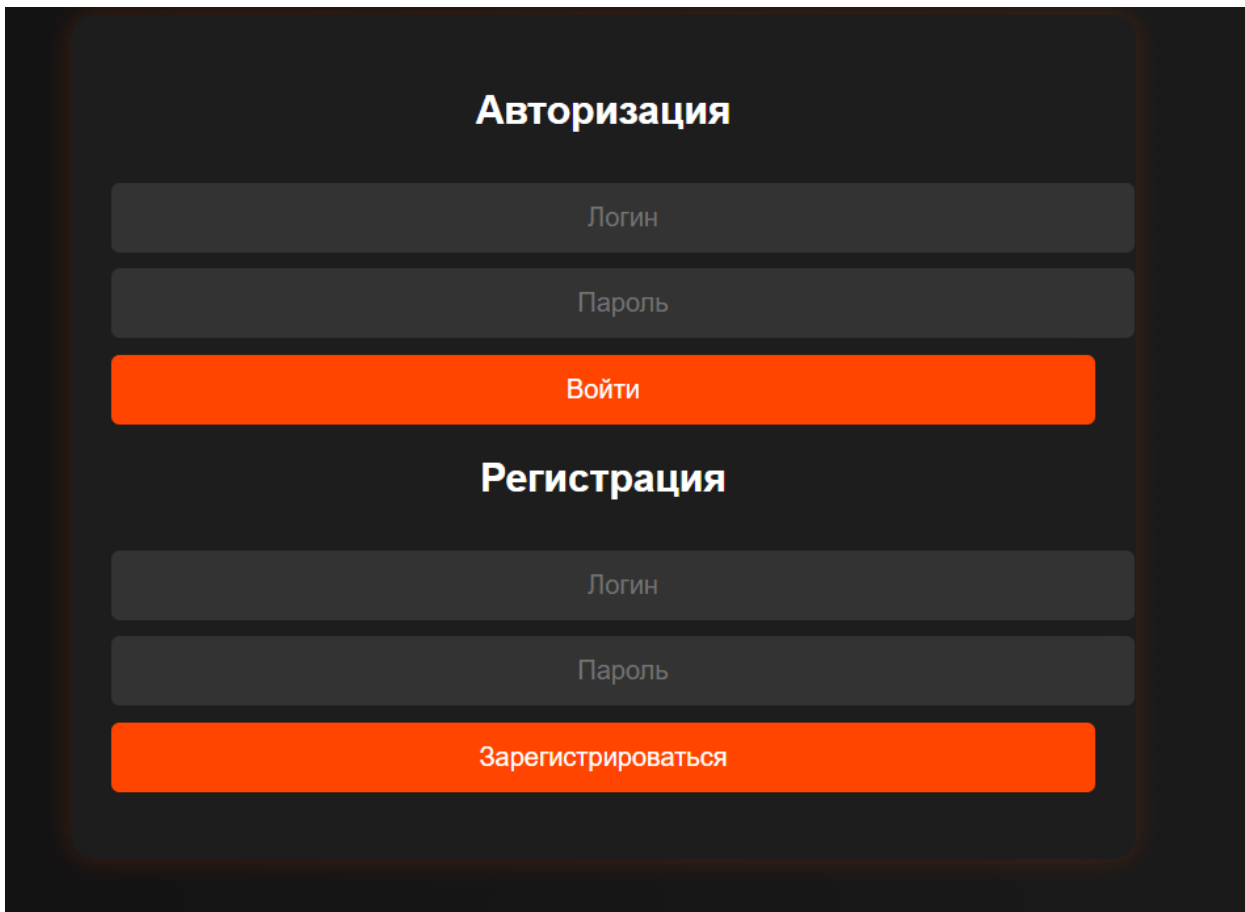


Рисунок 3. Программный интерфейс серверного API.

Реализация пользовательского интерфейса

Аутентификации пользователей:



The image shows a user interface for authentication and registration. It is divided into two main sections: 'Авторизация' (Authorization) and 'Регистрация' (Registration). Each section contains input fields for 'Логин' (Login) and 'Пароль' (Password), and a corresponding action button. The 'Авторизация' section has a 'Войти' (Login) button, while the 'Регистрация' section has a 'Зарегистрироваться' (Register) button. The interface is dark-themed with orange accents for the buttons.

Авторизация

Логин

Пароль

Войти

Регистрация

Логин

Пароль

Зарегистрироваться

Рисунок 4

Пользователь проходит регистрацию и далее может авторизоваться:

Авторизация

Логин

Пароль

Войти

Регистрация

15

..

Зарегистрироваться

✓ Регистрация успешна!

Рисунок 5

Выдается JWT токен

После авторизации пользователь попадает на страницу генерации и может ввести название песни или посмотреть ранее сохраненные тексты:

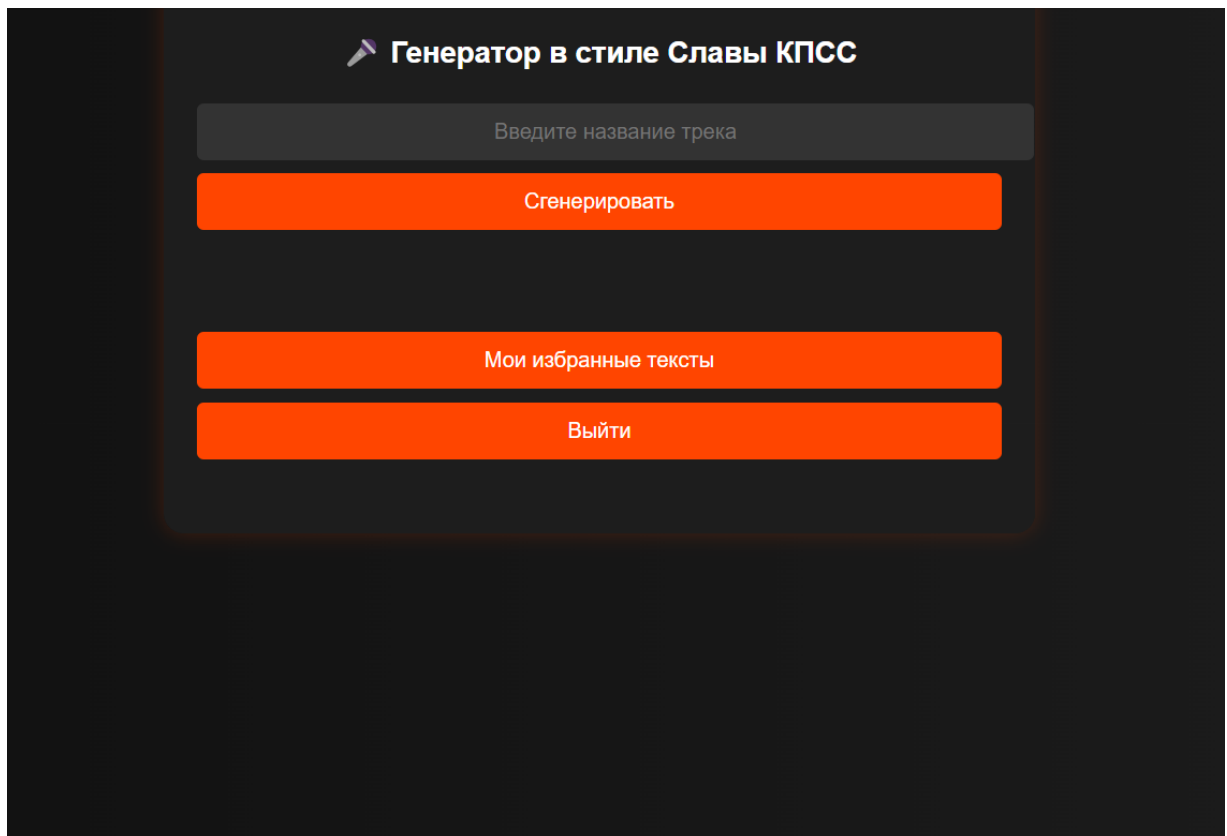


Рисунок 6

После того, как пользователь ввел строчку за который генератор должен продолжить текст, на сайте изображаются шаги генерации.

В этот момент через FAISS в векторной базе данных ищутся похожие текста из 1500 спаршенных песен исполнителя, на их основе генерируется текст, который видит пользователь

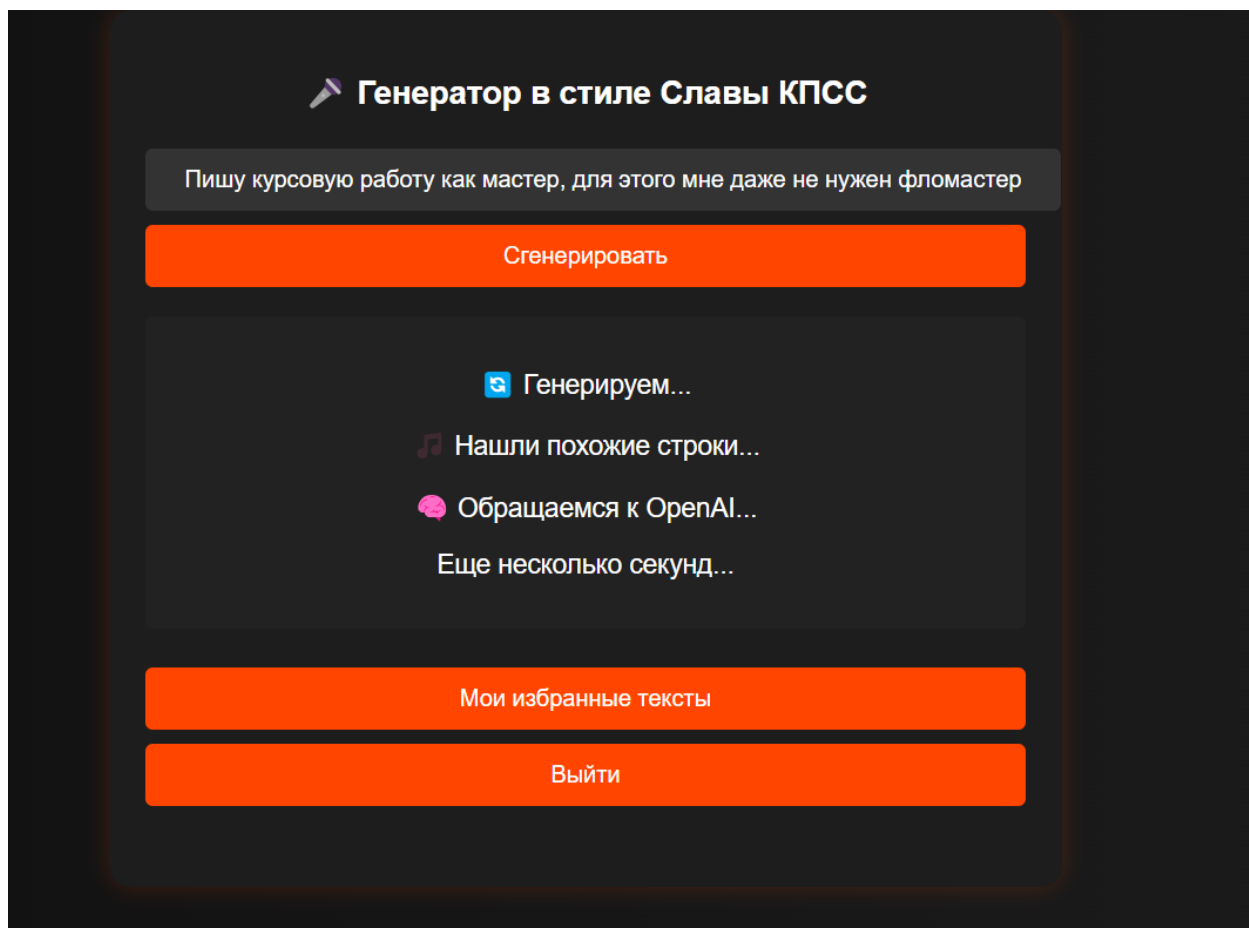


Рисунок 7

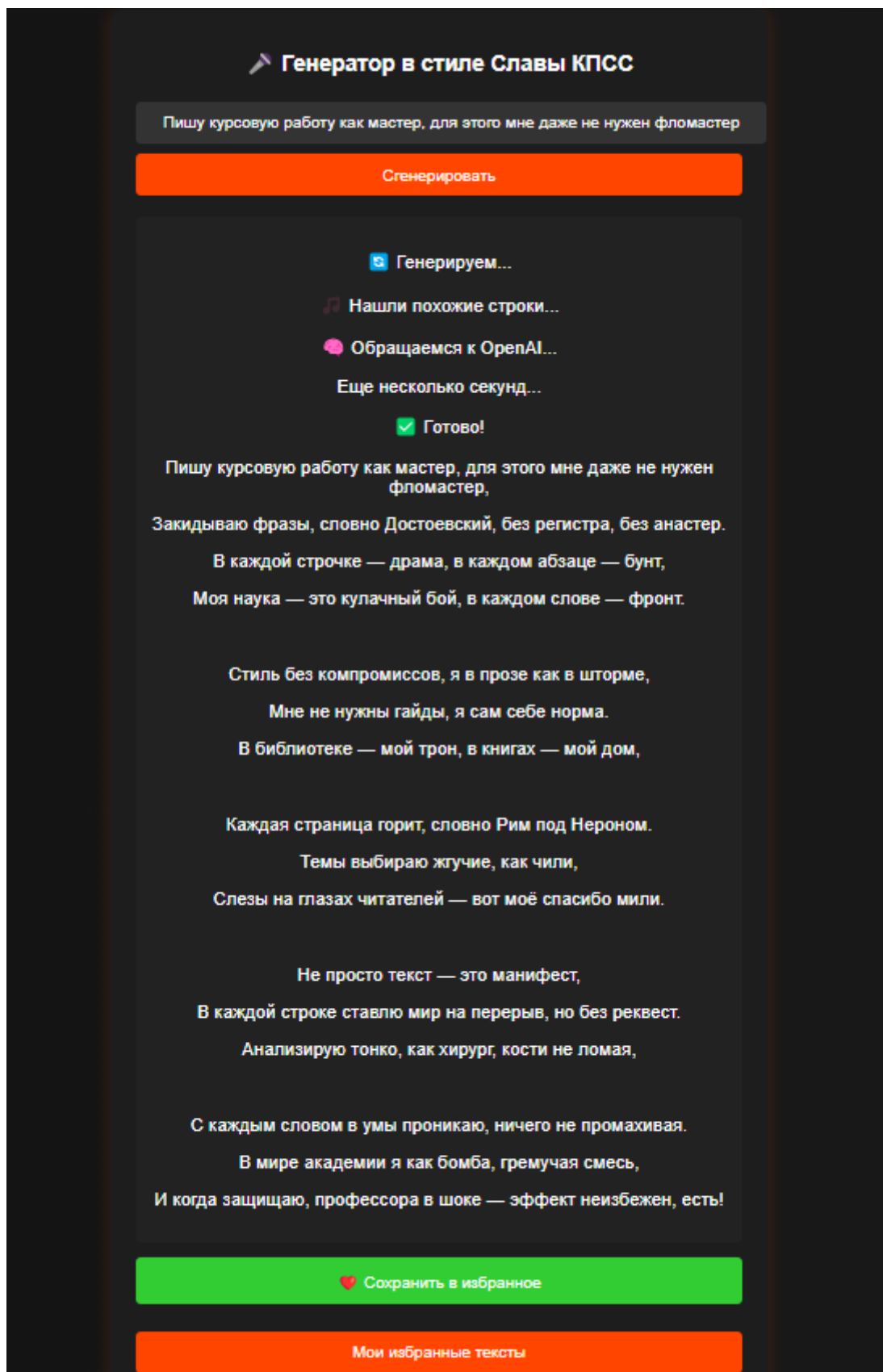


Рисунок 8

Далее пользователь видит текст и может нажать на кнопку «Сохранить в избранное»



Мои избранные тексты

← Вернуться к генератору

Выйти

Пишу курсовую работу как мастер, для этого мне даже не нужен
фломастер,
Закидываю фразы, словно Достоевский, без регистра, без анастер.
В каждой строчке — драма, в каждом абзаце — бунт,
Моя наука — это кулачный бой, в каждом слове — фронт.

Стиль без компромиссов, я в прозе как в шторме,
Мне не нужны гайды, я сам себе норма.
В библиотеке — мой трон, в книгах — мой дом,
Каждая страница горит, словно Рим под Нероном.

Темы выбираю жгучие, как чили,
Слезы на глазах читателей — вот моё спасибо мили.
Не просто текст — это манифест,
В каждой строке ставлю мир на перерыв, но без реквест.

Рисунок 9

Заключение

В ходе выполнения курсовой работы был проведён анализ работы классических систем управления сайтами, а также существующих моделей генерации текстов, исходя из которых были выявлены и сформированы требования к разрабатываемому веб-приложению.

Исходя из выбранной архитектуры и наложенных ограничений были сформированы требования к используемым технологиям внутри модулей. Была спроектирована архитектура данных, программная и системная архитектура в виде набора диаграмм в нотации UML.

Опираясь на выше изложенные требования и стек технологий было разработано веб-приложение и пользовательский интерфейс в рамках дисциплины «Web-программирование».

Таким образом, все поставленные ранее цели были выполнены.

Разработанное приложение является результатом данной курсовой работы.

Список использованной литературы

1. Мартин Фаулер - Архитектура корпоративных программных приложений. Издательский дом "Вильямс". 2006 г.
2. Флэнаган, Дэвид. JavaScript. Полное руководство, 7-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2021. — 720 с .
3. Янг А., Мек Б., Кантелон М. Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.
4. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. 2-е издание. — СПб.: Питер, 2021. — 336 с.
5. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 04.05.2021.