

2. Поддержка нескольких баз данных: Возможность подключения к нескольким базам данных одновременно делает сервис гибким и универсальным для различных систем.

3. Простая конфигурация: Настройка сервиса с помощью yaml файла обеспечивает легкость и удобство конфигурирования, что упрощает процесс развертывания и управления.

4. Высокая производительность: Благодаря использованию бинарных логов и эффективной системе их обработки, сервис обеспечивает высокую производительность и надежность.

5. Поддержка Kubernetes: Внедрение в K8s позволяет использовать все преимущества оркестрации контейнеров, включая масштабируемость, надежность и простоту управления.

6. Безопасность и устойчивость: Сервис разработан с учетом современных требований безопасности и устойчивости, что обеспечивает защиту данных и непрерывность работы.

ЛИТЕРАТУРА

1. Документация Golang [Электронный ресурс] – Режим доступа: <https://go.dev/> – Дата доступа: 10.05.2024

2. Документация WAL в PSQl [Электронный ресурс] – Режим доступа: <https://habr.com/ru/companies/postgrespro/articles/459250/> – Дата доступа: 10.05.2024

3. Документация RabbitMQ [Электронный ресурс] – Режим доступа: <https://www.rabbitmq.com/docs> – Дата доступа: 10.05.2024

4. Документация K8s [Электронный ресурс] – Режим доступа: <https://kubernetes.io/docs/home/> – Дата доступа: 10.05.2024

УДК 004.4

Студ. В.В. Гарельский

Науч. рук. ассист. Д.В. Сазонова

(кафедра информационных систем и технологий, БГТУ)

МУЗЫКАЛЬНЫЙ СТРИМИНГОВЫЙ СЕРВИС «REMIX»

В современном мире, где музыка стала неотъемлемой частью нашей повседневной жизни, использование информационных систем для управления музыкальными стриминговыми сервисами становятся все более актуальными. С учетом огромного количества доступных треков и альбомов, а также разнообразия музыкальных предпочтений каждого слушателя, эффективное управление музыкальными стриминговыми сервисами позволяет пользователям легко находить и воспроизводить любимую музыку.

Целью разработки являлось реализация интуитивно понятного и функционального музыкального стримингового сервиса, который предоставляет пользователям доступ к широкому спектру музыкального контента, включая информацию о треках, альбомах и исполнителях. Задача состояла в создании платформы, которая не только удовлетворяет потребности пользователей в удобном доступе к музыке, но и способствует развитию их музыкальных вкусов и интересов через персонализированный контент.

Разработанный музыкальный стриминговый сервис, как и его аналоги может предоставлять детальную информацию о треках, альбомах и исполнителях, включая жанры, рейтинги, отзывы пользователей и даже историю прослушивания. Это позволяет пользователям принимать более осознанные решения о том, что слушать дальше.

Также, данный сервис предлагает рекомендации по трекам и альбомам, основываясь на предыдущем прослушивании, предпочтениях и даже настроении пользователя. Сервис автоматически обновляет плейлисты и рекомендации на основе текущих музыкальных интересов пользователя, учитывая, что ему может понравиться.

Помимо всего прочего, он оптимизирует процесс поиска и воспроизведения музыки, чтобы сэкономить время и усилия пользователей, предлагая персонализированные музыкальные предложения и упрощая навигацию по огромному каталогу музыки.

Приложение разрабатывалось на языке Python с помощью фреймворка Django для реализации серверной части приложения, а также с использованием библиотеки React.js для написания клиентского приложения.

ЛИТЕРАТУРА

1. Документация фреймворка Django [Электронный ресурс] – Режим доступа: <https://docs.djangoproject.com/en/5.0/> – Дата доступа: 22.04.2024
2. Документация библиотеки React [Электронный ресурс] – Режим доступа: <https://ru.reactjs.org/> – Дата доступа: 22.04.2024
3. Документация PostgreSQL [Электронный ресурс] – Режим доступа: <https://www.postgresql.org/docs/> – Дата доступа: 22.04.2024