Техническое задание на разработку «Сервиса для совместного просмотра видео»

* 1. **Введение**
  2. Цель документа

Этот документ описывает требования к разработке «Сервиса для совместного просмотра видео» для возможности людям, находящимся далеко друг от друга проводить вместе досуг.

1.2 Актуальность и назначение проекта

Часто людей разделяет большое расстояние, и чтобы как-то преодолеть его можно за совместным просмотром видео-роликов.

* 1. Основные пользователи системы

Система предназначена для использования всеми интернет пользователями

* 1. **Цели и задачи проекта**
  2. Основные цели

Разработка сервиса для синхронного просмотра видео

* 1. Конкретные задачи
* реализовать плеер для воспроизведения видео;
* реализовать функции чата для общения во время просмотра;
* обеспечить синхронный просмотр видео для нескольких пользователей;
* предоставить возможность добавления видео из различных источников;
* обеспечить анонимный доступ;
* обеспечить простоту и удобство использования интерфейса.
  1. **Требования к системе**

3.1 Функциональные требования

* Реализация плеера: возможность просмотра видео на странице;
* Реализация просмотра со сторонних сервисов;
* Добавление чата: возможность обмениваться сообщениями между участниками одной комнат;
* Реализация синхронного просмотра: возможность у участников одной комнаты синхронно смотреть один видео-ролик.
  1. Нефункциональные требования

Производительность: отсутствие асинхронного просмотра.

* 1. **Требования к пользовательскому интерфейсу**
  2. Основные экраны
* Главная страница: отображает основную ознакомительную информацию о сервисе;
* Страница комнат: отображает уже активные комнаты, возможность создания новой комнаты;
* Страница просмотра: отображает плеер для просмотра и чат участников;
  1. Навигация

Система должна включать меню с разделами: «Главная», «Комнаты».

* 1. Юзабилити

Простой и понятный интерфейс для пользователя.

* 1. **Требования к технической реализации**
  2. Языки и технологии
* Backend: Python, Django.
* Frontend: HTML, CSS.
* База данных: PostgreSQL, Redis(для хранения кэша).
  1. Архитектура системы
* Использование REST API и WebSocket для поддержания чата.
* Хранение данных в реляционной базе данных.
* Хранение данных в нереляционной базе данных(кэш).
  1. **Требования к безопасности**

6.1 Безопасность соединения

* Шифрование данных: использование HTTPS для защиты данных, передаваемых между клиентом и сервером.
* WebSocket безопасность: если вы используете WebSocket для чата, убедитесь, что соединение также защищено с помощью WSS (WebSocket Secure).
  1. **Ограничения и допущения**

7.1 Сроки выполнения

Полная реализация проекта – 1 месяц с начала разработки

* 1. **Требования к тестированию и приемке**
  2. Типы тестирования
* Функциональное тестирование: проверка всех реализованных функций.
* Нагрузочное тестирование: тестирование на корректную работу у 4-х пользователей.

9. **Требования к документации**

9.1 Пользовательская документация

* Руководство пользователя с инструкциями по созданию комнаты и просмотра видео.

9.2 Техническая документация

* Описание архитектуры, структура базы данных, API.

10. **План реализации**

10.1 Этапы разработки

1. Анализ и проектирование — 1 неделя.

2. Разработка прототипа — 4 дня.

3. Полноценная разработка и тестирование — 2 недели

10.2 Сроки выполнения этапов

* Дата начала: 20 ноября 2024 года. (1732136272)
* Дата завершения: 19 декабря 2024 года. (1734623844)

10.3 Ответственные лица

* Ведущий разработчик: Фаттахова Л.Р.