Текст к презентации.

1 слайд.

Добрый день уважаемые присутствующие, рад вам представить свою дипломную работу с названием «музыкальная социальная сеть».

2 слайд.

Для того, чтобы наиболее информативно презентовать вам суть своего проекта, я разбил презентацию на 3 основных раздела, каждый из которых состоит из подпунктов, являющимися ключевыми особенностями проекта.

3 слайд.

Всем нам известно, что музыка объединяет миллионы человек по всему миру. Масштаб индустрии настолько внушителен, что превосходит любые другие направления искусства. Помимо тех тысяч музыкальных произведений, ставшими классикой, ежегодно появляется множество новых исполнений, обладающих уникальным набором жанровых оттенков. И каждый из них находит своего слушателя.

Идея моего проекта основывается на том, чтобы предоставить площадку, где центром внимания становятся персональные музыкальные собрания, представленные в виде плейлистов. Место, где каждый сможет не только собрать свою любимую коллекцию, но и представить ее миру, поучаствовать в дискуссиях, найти людей с похожими музыкальными вкусами, и еще шире открыть для себя музыкальный мир.

4 слайд.

Исходя из идеи сформировалось техническое задание к проекту, который должен представляет из себя веб-сервис и предлагает пользователям спектр следующих функциональных возможностей. Все они представлены на слайде.

5 слайд.

С целью разработать проект были использованы следующие технологии:

Python – как язык наш программирования, на котором осуществлялась полная разработка всех алгоритмов приложения и на базе которого использовались вспомогательные библиотеки.

Django – веб – фреймворк, предоставляющий множество решений для готовых задач, связанных с веб-сервисами.

MySQL – реляционная база данных для для хранения и управления данными.

HTML/CSS – с помощью которых были созданы и оформлены веб-страницы

Docker – для упаковки приложения в изолированную виртуальную среду.

6 слайд.

Собрав комплект требований и инструментов, я приступил к разработке приложения. Стоит отметить, что для заполнения музыкальным контентом нашего веб-приложения использовался API стороннего сервиса Musixmatch, который содержит современную библиотеку музыкальных произведений.

7 слайд.

Работа над проектом началась с разработки структуры реляционной базы данных. Она представлена на слайде в виде ERR таблиц и связей между ними. Из основных сущностей я выделил следующие: пользователь, музыкальный трек, альбом, исполнитель, жанр, плейлист, комментарий.

8 слайд.

Содержание корневого пути нашего проекта выстроилось подобным образом. Фреймворк Django предоставил основную архитектуру веб сервиса инициализировав основные пакеты и модули и выполнив настройку сервера на определенный порт. Здесь хочется отметить модули views.py и urls.py. Это модули, в которых велась основная часть разработки.

9 слайд.

Клиентская часть представлена в виде HTML кода браузерных веб-страниц, выдающихся в результате HTTP запросов по объявленным URL маршрутам и содержащих различные ссылки для переадресаций, форм для отправки данных на сервер и проверок.

На слайде представлены основные блоки контента. Боковой блок и заголовок являются статичными и представлены на всех маршрутах приложения. Основной блок является динамическим и представляет результаты запросов.

10 слайд.

Серверная часть содержит в себе функции и классы обработчики HTTP запросов. Принадлежность URL маршрутов к их обработчикам отражается на слайде. Основными задачами обработчиков являются управление данными и идентификация пользователей.

Здесь хочется отметить разработанный класс миксин, имеющий вспомогательное значение для представлений top-tracks и search, которые первым этапом своего алгоритма обращаются с помощью HTTP запроса методом GET к API сервису и получают музыкальный контент. Класс миксин является прослойкой между данными, полученными от musixmatch и нашим локальным хранилищем. Класс в качестве аргумента получает ответ веб-сервиса/6 обрабатывает его, загружая в положительном случае данные в локальное хранилище и выдает обратно обработчику отформатированный ответ.

11 слайд.

Разработанное приложение на локальной машине в дальнейшем хочется разместить в открытом доступе. Для этого необходимо переместить все содержание проекта на удаленную машин. Для того, чтобы сделать это быстро, удобно и безопасно я использовал возможности докера и собрал все в единый контейнер. Мой контейнер под управлением докер композ содержал в себе два образа, один с базой данных, второй с нашим приложением, собранным в образ по алгортиму докерфайла. Это позволило создать единую изолированную виртуальную сеть.

12 слайд. (видео)

В результате проделанной работы было разработано приложение, отвечающее требованиям к основным функциональным возможностям. Получилось предоставить площадку, где пользователи могут осуществлять поиск любимых музыкальных произведений и взаимодействует друг с другом на основе своих предпочтений.

13 слайд.

Работа над проектом мне очень понравилась и есть множество деталей, которые я не успел проделать. Веб-сервис обладает интересным потенциалом для развития. В частности, разработка тестирования, модификации существующих представлений с целью оптимизации и усовершенствования, внедрение дополнительных возможностей, указанных в ТЗ проекта. Визуальное улучшение клиентской части. Решение таких задач позволит приложению выходить на более высокие уровни.