

# Project Report

**Name:** Sankalpa Sarkar

**Email:** [22f3003030@ds.study.iitm.ac.in](mailto:22f3003030@ds.study.iitm.ac.in)

## **Introduction**

Greetings! I am Sankalpa Sarkar, and I come from Siliguri, West Bengal. Currently, I'm in the diploma level of this course. Every term there's a new challenge to look forward to, this term was no different, this project was an incredible way to test the skills I've gained from the MAD I course. I was able to showcase a lot of my web development skills on this project, I hope you won't be disappointed.

## **Project Description**

In this project, our task was to create a music app. The application should allow users to view songs, also they are allowed to perform any kind of CRUD operations on the songs, playlist, albums depending on their role. Admins are allowed to view app information, like song performance, blacklist/whitelist creator, remove songs to name a few. So, it's a multi-user app, where depending on the role, accesses are granted.

## **Technologies Used**

**Flask:** To perform all kinds of backend tasks (Controller), create different routes, as well as deciding on what to display on the view.

**Flask\_sqlalchemy:** To create all the models and perform CRUD operations on those models.

**Flask\_restful:** To create the API.

**SQLITE3:** To store the data.

**Werkzeug:** To perform validations for the API.

**Matplotlib:** To create graphs for the data.

**Jinja2:** To display the data into the page.

**HTML, CSS:** To create all the templates and style them.

## **Database Schema**

To understand the app database schema, please follow the link provided below.

Link: [Database Schema](#)

## **App Architecture and Features**

The App Architecture is as follows: the root folder contains a folder music, and a python file run.py (To run the application). Inside the music folder, we have 3 folders (database, static, templates), 5 python files and a YAML file. The database folder contains the app database, the static folder contains 3 folder: css (to store the app styles), images (To store the app assets), songcollection which has the song data (Cover Image and mp3 files) and the templates folder stores the HTML files.

Moving on to the features of this application, they are as follows:

1. Users can view and like songs, view albums, they are also allowed to follow creators, create, update, delete, view playlists, search for songs and albums (based on keywords, song name, creator name), update their profile, they can also register themselves as creators, which would enable them to the below mentioned features.

2. Creators can check their performance like total songs, albums, followers, likes on each song. They can perform on their songs and albums. Also, they are allowed to update their profile and check whether they are blacklisted or not.
3. Admins can check the app performance like total songs, albums, playlists, users, creators etc. They can remove songs, albums based on company policy as well as blacklist/whitelist creators.

The app, has validations for forms, before entering the data into the database and it has proper styling and aesthetics.