MUSIC STREAMING APPLICATION
PROJECT REPORT

NAME: MOHIT MAHUR
ROLLNO.: 22f1000344
EMAIL: 22f1000344@ds.study.iitm.ac.in

DESCRIPTION:

"This platform is a collaborative online music streaming service, listening to music. Developed with a Vue.js frontend and Flask backend, it boasts a seamless user experience.

In addition to core functionalities, it offers advanced features such as scheduled tasks and daily reminders powered by Redis and Celery. User authentication is fortified with Flask Security's token-based system. The development process followed a structured approach, starting with database schema design, followed by robust login mechanisms, API development, and user interface refinement. Finally, the implementation of Celery jobs adds efficiency to background processes."

I began by laying the foundation with a comprehensive database schema. Next, I established a secure and user-friendly login system. Then, I meticulously crafted the APIs, logic, and user interface to ensure a seamless experience. Finally, I integrated Celery to handle background tasks, completing the framework for a robust and efficient application."

Technologies Used:

Vue.js: Powering the frontend/client-side of the application, Vue.js ensures a dynamic and responsive user interface.
Flask: Serving as the backbone of the application, Flask handles the server-side/backend operations, providing a robust and scalable infrastructure.
Redis and Celery: Leveraged for executing scheduled jobs and sending daily reminders through Google MailHog, ensuring timely notifications and task management.
Flask-Restful: Facilitating the management of API calls, Flask-Restful streamlines communication between the frontend and backend components of the application.
Flask-Security: Providing token-based authentication, Flask-Security strengthens the security measures of the application, safeguarding user data and interactions.
Smtplib and MIMEMultipart: Enabling the transmission of multipart messages via Simple Mail Transfer Protocol, Smtplib and MIMEMultipart ensure effective communication functionalities.
SQLite3 and Flask-SQLAlchemy: Managing the relational database of the application, SQLite3 and Flask-SQLAlchemy ensure efficient data storage and retrieval, maintaining data integrity and consistency.

The database architecture is structured using Flask-SQLAlchemy, enabling the creation of database models and tables seamlessly.
Seven tables constitute the database schema:

ser: Stores user information, distinguishing them based on their roles.
Role: Defines the roles available within the application.
RolesUsers: Facilitates the association between users and their respective roles.
Song: Contains details of individual songs.
Album: Stores information about albums, establishing a many-to-one relationship with songs.
Playlist: Represents playlists created by users.
Playlist_Song_Table: Manages the many-to-many relationship between playlists and songs, storing the relevant data to link them.

System Design:

The web application adheres to the MVC (Model-View-Controller) architectural pattern:
Model (M): Managed by Flask, which interacts with the database and oversees the data model.
View (V): Handled by Vue.js, where Vue components govern the interactive user interface.
Controller (C): Managed by Flask routes, responsible for executing all business logic on the backend.
Folder Structure:

Instance Folder: Contains the database of the application.
Main.py: The core file housing the Flask app instance, Celery app instance, and the initialization of APIs and databases.
Models.py: Holds code responsible for creating database tables.
Resources.py: Manages API calls related to songs and albums.
Worker.py, Celeryconfig.py, Task.py: Handle Celery configuration, scheduled jobs, and daily reminders.
Views.py: Contains the code for all routes and endpoints within the application.

To run the app, follow these steps:
Unzip the Project Folder: Extract the contents of the project folder to your desired location.
Install Dependencies: and run main.py file.

Link of Presentation:-
https://drive.google.com/file/d/1qKcP2bm5fGdZ75P2s4vPEfpIV79MNJlp/view?usp=sharing