Project Documentation

Rhythmic Tunes: Your Melodic Companion

1.Introduction :

- **Project Title: Rhythmic Tunes: Your Melodic Companion**
- **Team ID**: NM2025TMID39978
- **Team Leader**: *Ishwarya P [sac2427csc5542@ssacollegechennai.com]*
- Team Member:

Jechintha joyce S [sac2427csc5492@ssacollegechennai.com]
Janusri D [sac2427csc5471@ssacollegechennai.com]
Jai sree R [sac2427csc5647@ssacollegechennai.com]

2. Project Overview :

- **Purpose :**
  - Rhythmic Tunes is a web app that lets users explore, play, and manage songs or melodies.
  - It provides personalized playlists, real-time playback, and a seamless music experience. It also aims to enhance user engagement by offering intuitive features and a user-friendly interface.
- **Features:**
  - Upload and stream audio tracks
  - Playlist creation and management
  - Real-time audio controls (play, pause, skip)
  - User accounts with favourites and history
  - Admin dashboard for content moderation
  - Personalized recommendations based on listening habits
  - Search and filter functionality for quick music discovery
  - Offline mode for playing downloaded songs
  - Social sharing of playlists and tracks

3.Architecture :

- **Frontend** : React.js with Material-UI for an elegant UI
- **Backend** : Node.js + Express.js for APIs and server logic
- **Database** : MongoDB (stores users, playlists, songs, playback data)
- **Authentication** : JSON Web Tokens (JWT) for secure login and session management
- **Cloud Storage** : AWS S3 / Google Cloud Storage for storing audio files
- **Streaming Service** : Integration of audio streaming protocols for smooth playback
- **State Management :** Redux (or Context API) for handling global state in frontend

- **API Security** : Middleware for input validation, rate limiting, and secure endpoints
- **Testing :** Test + Mocha/Chai for unit and integration testing
- **Deployment** : Docker + Kubernetes for containerization and scalable deployment

4.Setup Instructions :

• Prerequisites: – Node.js – MongoDB 1 – Git – Visual Studio Code

• Installation Steps:

- # Clone the repository git clone
- # Install client dependencies cd client npm install
- # Install server dependencies cd ../server npm install

5.Folder Structure :

```
rhythmic-tunes/
├── client/          # React frontend
│   ├── components/
│   ├── pages/
│   └── assets/
├── server/          # Node.js backend
│   ├── routes/
│   ├── models/
│   ├── controllers/
│   └── middleware/
└── README.md
```

6.Running the Application:

- **Frontend:** cd client npm start
- **Backend:** cd server npm start
- **Access:** Visit http://localhost:5173

7.API Documentation:

- **User:**
  - POST /api/user/register – Create account
  - POST /api/user/login – Log in
- **Tracks:**
  - POST /api/tracks/upload – Upload song
  - GET /api/tracks/:id – Get song details
- **Playlists:**
  - POST /api/playlists/create

- GET /api/playlists/:userId

8.Authentication :

- JSON Web Token (JWT) for loggin sessions
- Middleware to protect private routes
- Password hashing using bcrypt for secure storage.
- Token expiration and refresh mechanism for improved security.
- Role-based access control (Admin, User) .Secure password reset with email verification

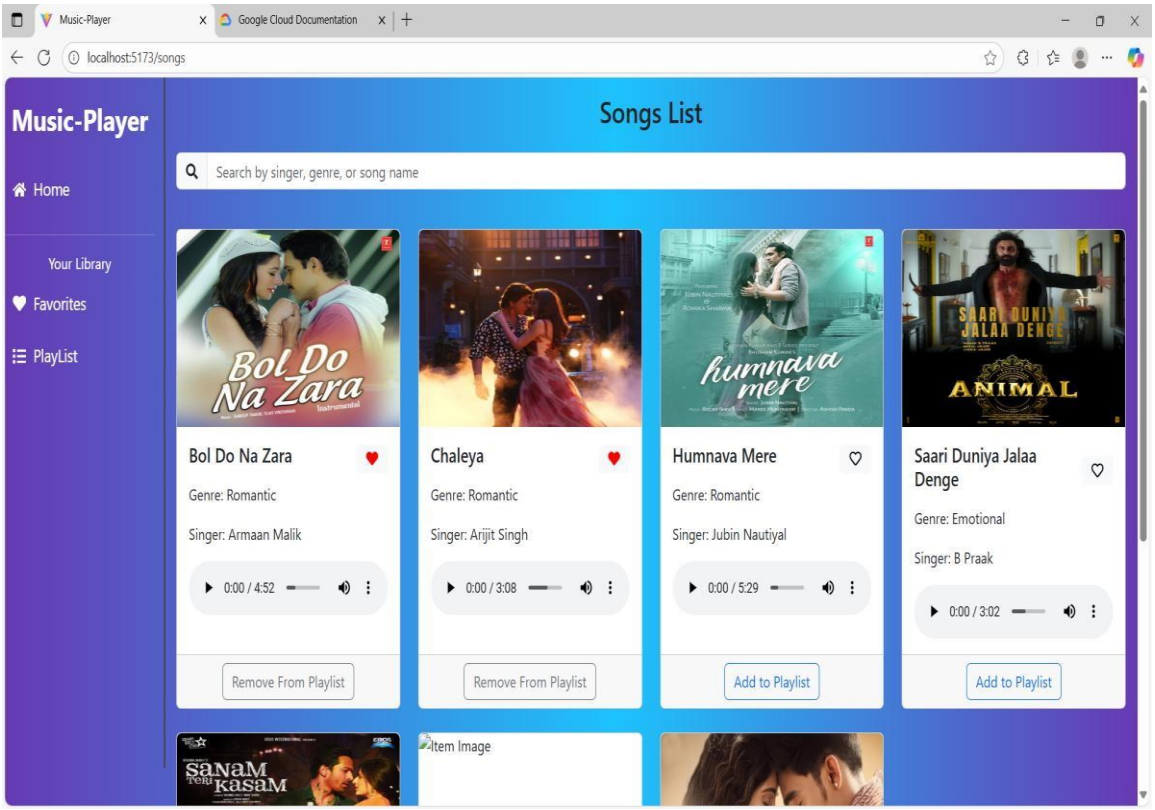
9.User Interface :

- Landing Page with featured songs .
- Music Player with controls .
- Playlist Dashboard .
- Admin Panel for uploads and moderation

10.Testing ;

- Manual testing for playback, upload, and playlist creation
- Tools: Postman, Chrome DevTools, Jest for unit tests
- Automated unit testing for APIs (login, upload, playlist creation).
- Integration testing for end-to-end workflows (e.g., user login→ upload song add to playlist).
- Cross-browser testing (Chrome, Firefox, Edge, Safari)

11.Screenshots or Demo:



12.Known Issues:

- Audio seeking not yet supported on some mobile browsers .
- Limited offline playback .
- Occasional delay in song loading on slow networks .
- Volume slider not persisting between sessions

13.Future Enhancements:

- Recommendation engine using AI .
- Collaborative playlists .
- Mobile app version .