Project Documentation

Rythamic Tunes

1. **Introduction**

A Rhythmic Tunes Project is a creative endeavor focused on exploring the intricate relationship between rhythm and melody in music. The core idea of such a project is to develop and showcase musical compositions that highlight rhythmic patterns as a central element, often with diverse genres, tempos, and time signatures. Whether it’s electronic, acoustic, or experimental, the project delves into how rhythm can drive the mood and flow of music, enhancing its emotional impact.

• **Project Title** : Rhythmic Tunes

• **Team ID**  : NM2025TMID39394

• **Team Leader**: ROSHAN R & [gibeyonroshan@gmail.com](mailto:gibeyonroshan@gmail.com)

• **Team Members**:

– GOKULA KANNAN I & [gokulkannan2105@gmail.com](mailto:gokulkannan2105@gmail.com)

– MURUGAN M & [murugan742006@gmail.com](mailto:murugan742006@gmail.com)

– NAVEEN K & [kalinaveen21@gmail.com](mailto:kalinaveen21@gmail.com)

2. **Project Overview**

• **Purpose**: SB Works connects clients and freelancers through project postings, bidding, and real-time communication.

• **Features**:

– Project posting and bidding

– Secure chat system

– Feedback and review system

– Admin control panel

3. **Architecture**

• **Frontend**: React.js with Bootstrap and Material UI

• **Backend**: Node.js and Express.js managing server logic and API endpoints

• **Database**: MongoDB stores user data, project information, applications, and chat messages

4. **Setup Instructions**

• **Prerequisites**:

– [Node.js](http://node.js)

– MongoDB

– Git

– React.js

– Express.js – Mongoose – 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**

SB-Works/

|-- client/ #React frontend

|\_\_components/

L\_\_ pages/

|\_\_ server/ #Node.js backend

|\_\_routes/

|\_\_ models/

|\_\_ controllers/

6. **Running the Application**

• **Frontend**:

cd client

npm start

• **Backend**:

cd server npm

Start

• **Access**:Visit <http://localhost:3000>

7. **API Documentation**

• **User**:

– /api/user/register

– /api/user/login

• **Projects**:

– /api/projects/create

– /api/projects/:id • Applications: /api/apply

• **Chats**:

– /api/chat/send

– /api/chat/:userId

8. **Authentication**

• JWT-based authentication for secure login

• Middleware protects private routes

9. **User Interface**

• Landing Page

• Freelancer Dashboard

• Admin Panel

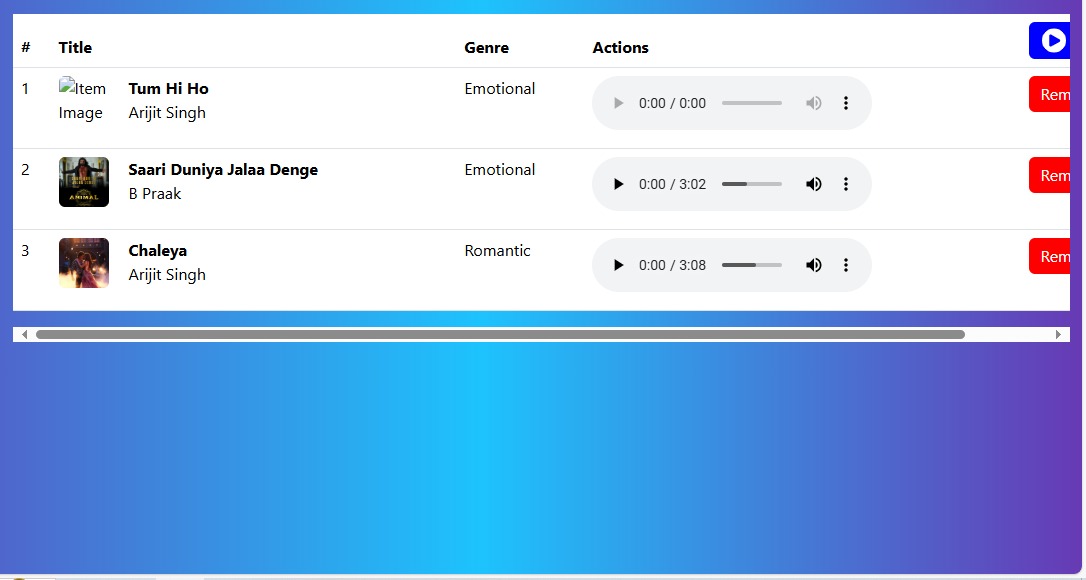
• Project Details Page

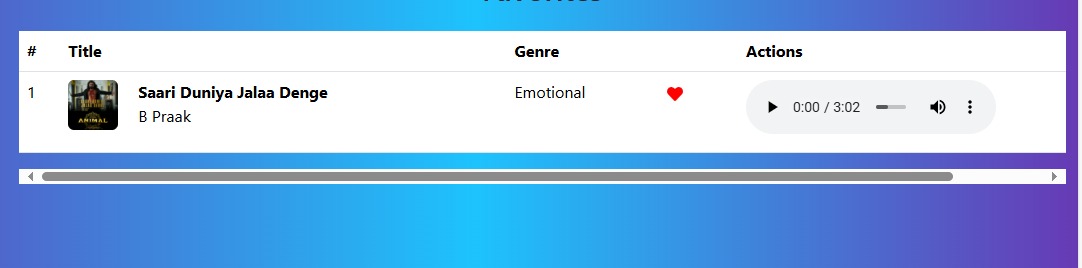
10. **Testing**

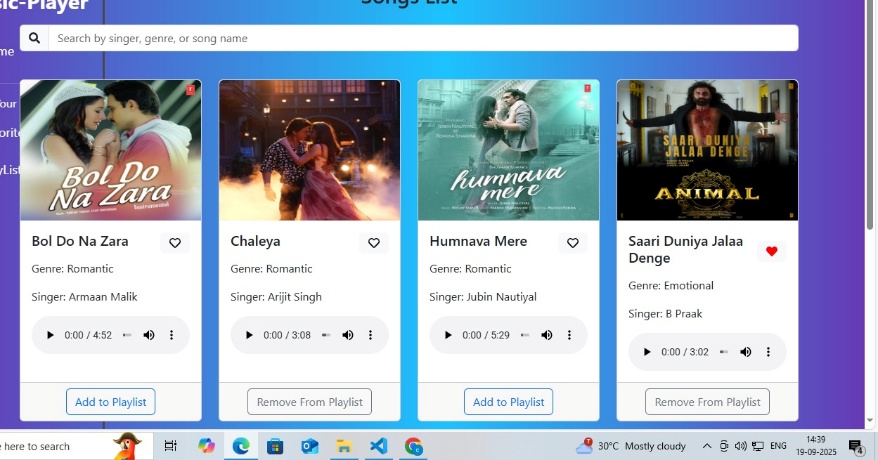
• Manual testing during milestones

• Tools: Postman, Chrome Dev Tools

11. **Screenshots**

****

****

****

**or Demo**

12. **Known Issues**

13. **Future En hancements**

1. AI-Powered Music Generation

Use machine learning to create adaptive background scores based on user mood, genre, or activity.

Auto-generate loops and remixes from existing tracks.

2. Personalized Playlists & Recommendations

Smart playlist creation based on listening history.

Mood-based music suggestions (happy, workout, focus, etc.).