**Project Documentations**

**Rhythmic Tunes:Your Melodic Companion**

**1. Introduction**

**• Rhythmic Tunes:**Your Melodic companion

**• Team ID :** NM2025TMID46907

**• Team Leader**:Dhiyasri A & a.dhiyasri9655022892003@gmail.com

**• Team Members:** 5

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Email ID** |
| **1.** | NITHYA M | nithyammurugesan775@gamil. com |
| **2.** | MONISHA N | Sathiyamonisha57@gmail.com |
| **3.** | NAVITHA S | Navithanavitha1827@gmail.com |
| **4.** | NEELA N | Neela.n11032007@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

– 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

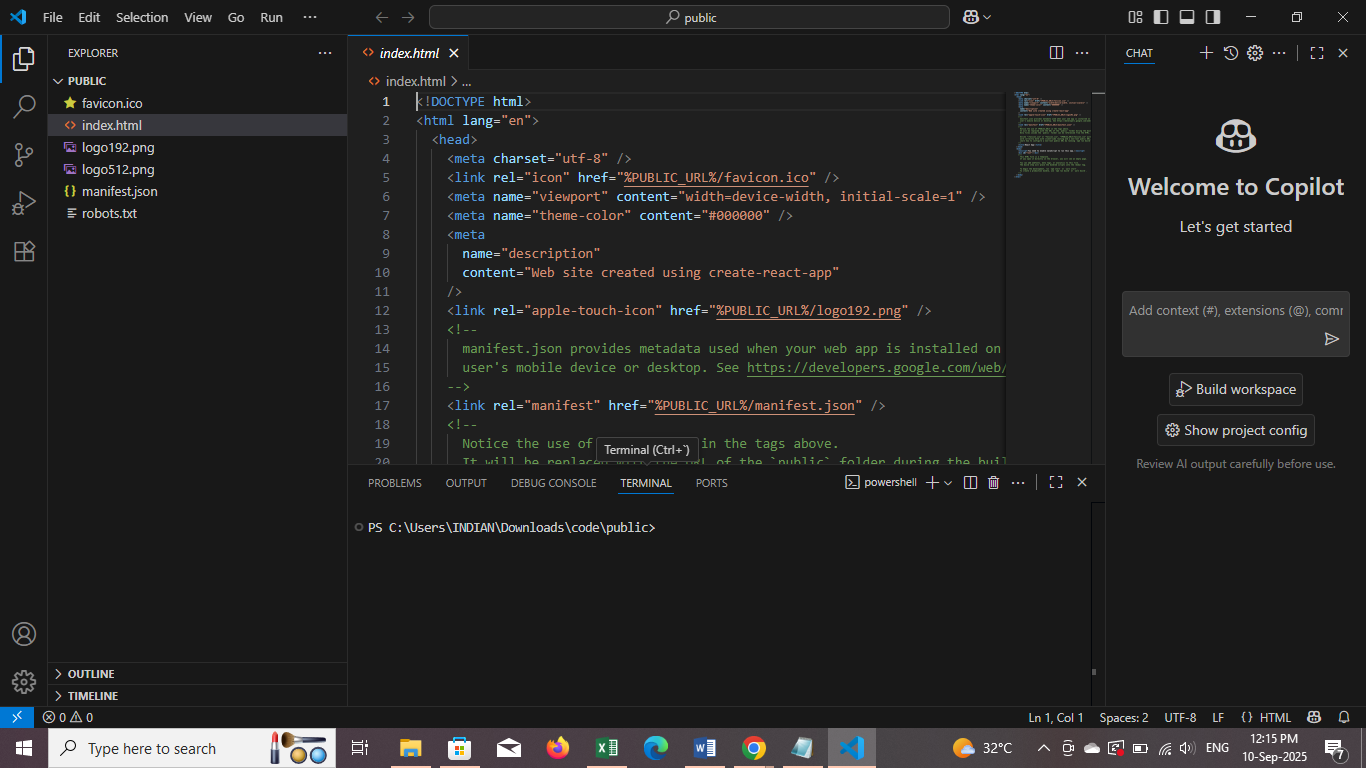
• Project Details Page

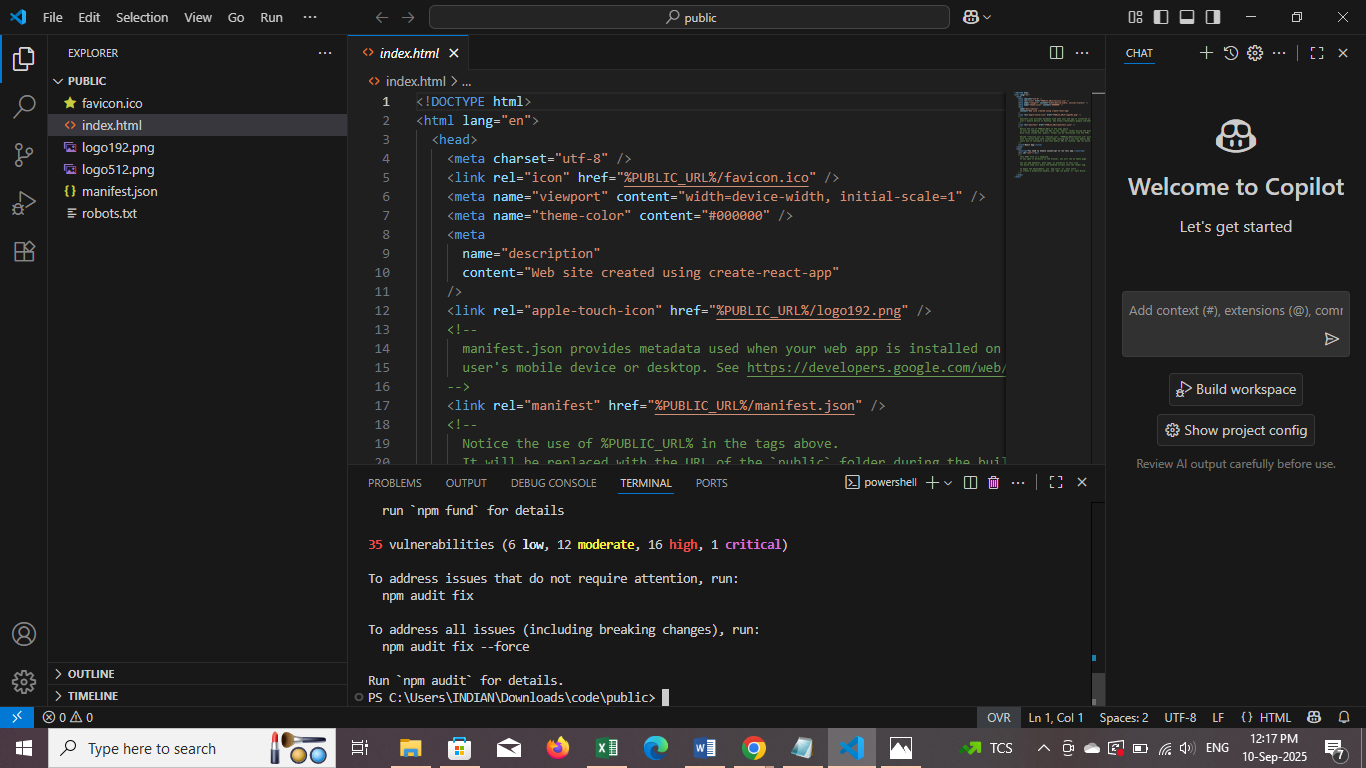
**10. Testing**

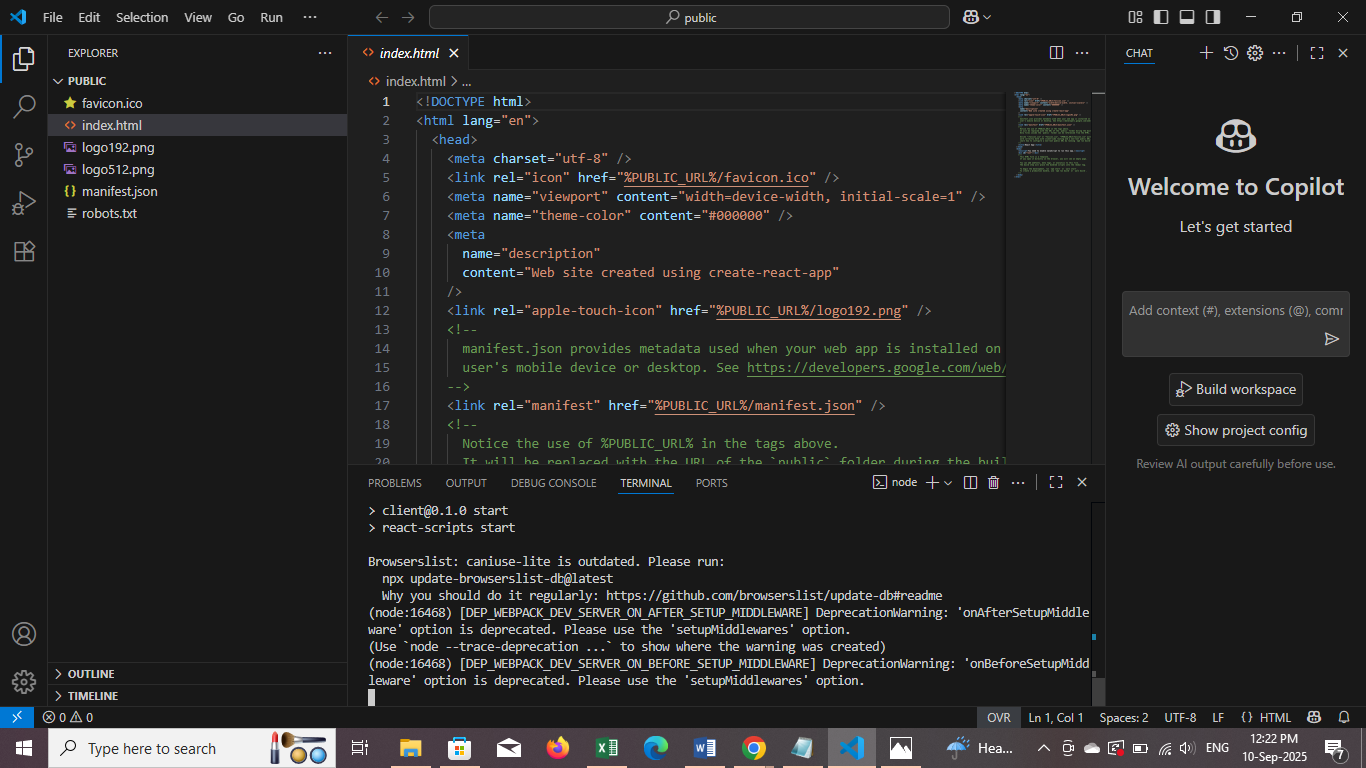
• Manual testing during milestones

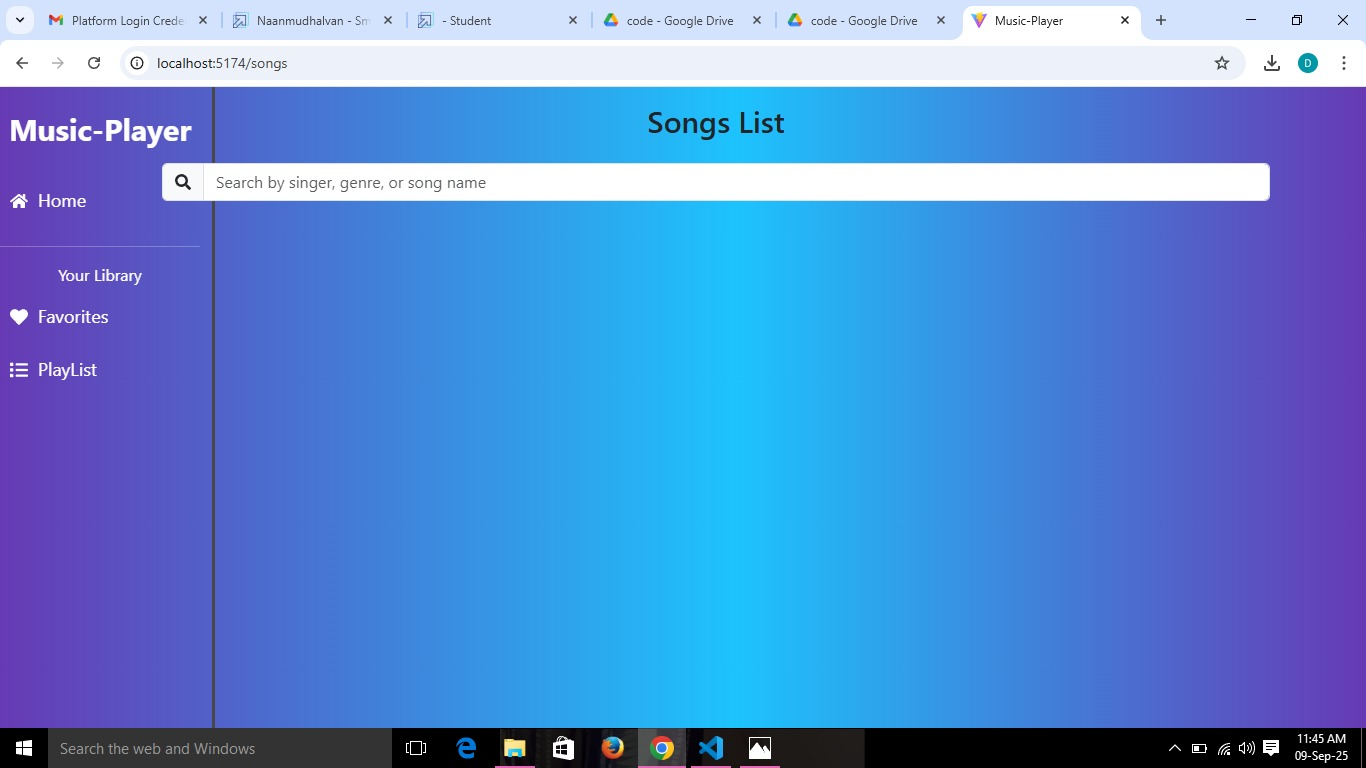
• **Tools**: Postman, Chrome Dev

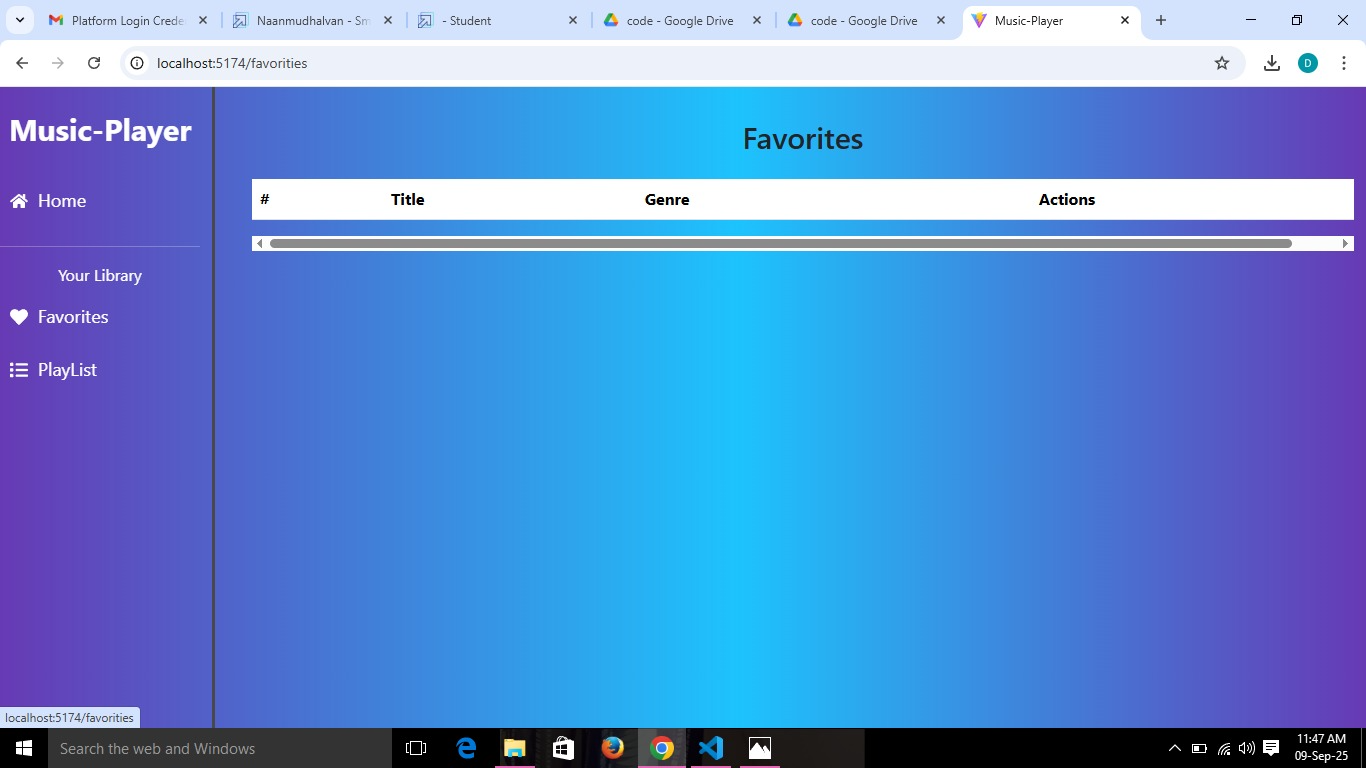
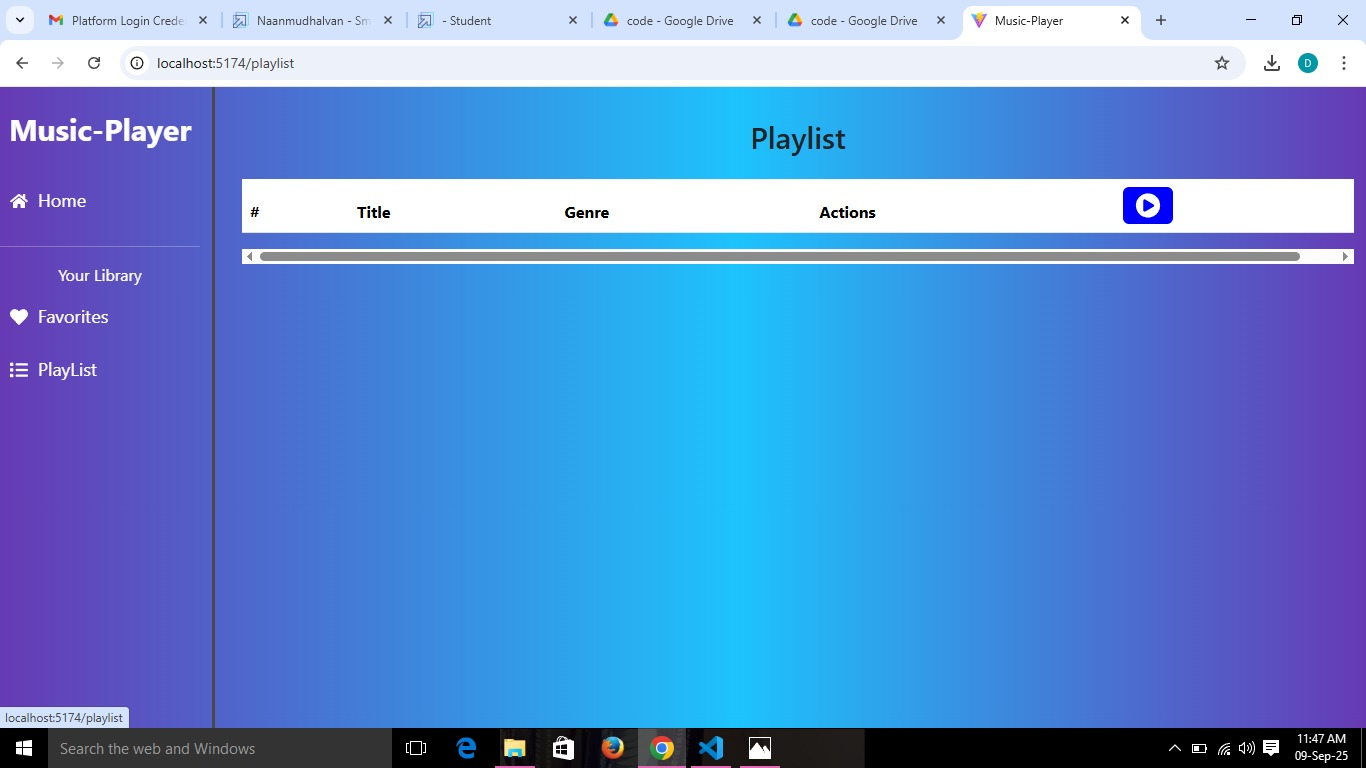
**11. Screenshots or Demo**

****





ss



**12. Known Issues**

1. Transition Challenges

2. Soloing and Syncopation

3. Rhythmic Notation Errors

4. Maintaining a Steady Beat

5. Cognitive and Learning Challenges

**13. Future Enhancements**

**Features**

1. Personalized Playlists: Customized playlists tailored to individual preferences and moods.

2. Mood-Based Music: Music selection based on emotional states, such as relaxation, energy, or focus.

3. Rhythmic Variety: A diverse range of rhythms and genres to cater to different tastes.

4. Dynamic Soundscapes: Immersive soundscapes that adapt to the user's environment or activity.

5. Interactive Music: Music that responds to user input, such as tempo or genre changes.

**Technical Enhancements**

1. AI-Powered Rhythm Generation: Tools like Rhythm X use AI to create, visualize, and edit rhythmic sheet music for percussion, offering personalized advice and suggestions.

2.Dynamic Instrumentation: This technique involves varying instrumental arrangements and textures to enhance emotional impact, narrative flow, or listener engagement.

3. Rhythmic Beat Generation: AI-powered tools can generate high-quality rhythmic beat music, allowing users to create tracks effortlessly.

4. Loop-Based Rhythmic Beats: This technique uses repeated loops to form the core rhythmic structure, common in electronic music, lo-fi, and hip-hop.

5. Polyrhythmic Beats: Involves two or more conflicting rhythmic patterns played simultaneously, popular in African traditional music, progressive rock, and experimental jazz.