# MUSIC PLAYER: YOUR MELODIC COMPANION

## Introduction

- **Project Title : MUSIC PLAYER**
- **Team ID : NM2025TMID38524**
- **Team Leader : S. Srinidhi & srinithi22nd@gmail.com**

  **(CODER and VIDEO RECORDER)**
- **Team Members :**

  1. **R. Supriya & supriyajagathi2006@gmail.com**

     **(VIDEO RECORDER)**
  2. **R. Vasuki & vasukiii1128@gmail.com**

     **(DOCUMENTOR)**
  3. **D. Yogeshwari & priyadevaraj1307@gmail.com**

     **(DOCUMENTOR)**

# Project Overview

- Purpose: The primary goal of Music Streaming is to provide a seamless platform for music enthusiasts, enjoying, and sharing diverse musical experiences. Our objectives include.

- Features :

  - Create playlist- compile and organize a list of songs, videos, or other media into a custom collection for easy playback and access.
  - **Add favourite songs-** saving specific songs to a personalized list or collection, allowing easy access and quick playback of preferred tracks.
  - **Download songs-** to save music files from the internet or an app onto a device for offline listening.
  - **Playback speed-** allowing users to listen or watch faster or slower than the original speed.
  - **Add songs**- include or insert music tracks into a playlist, library, or collection for easier access and organization.

# Architecture

**Frontend:** The **frontend** is the part of a website or application that users interact with directly. It includes everything visible in the browser—like layout, design, and user interface—typically built with HTML, CSS, and Java script, react js,node.js and boots trap.

**Backend:** The **backend** is the server-side part of a website or application that handles business logic, database interactions, and user authentication. It processes requests from the frontend and sends back the appropriate data, react, node.js.

**Database:** A **database** is an organized collection of data stored electronically for easy access, management, and updating. It enables applications to efficiently retrieve, insert, and manipulate information, Mongo DB stores user data, project information, applications, and chat messages.

# Setup Instructions

- Prerequisites :
  - Html
  - CSS
  - Java script
  - Mongo DB
  - Node.js
  - Git
  - React.js
  - Vite
  - Visual Studio Code

- Installation Steps :

```
# Clone the repository git clone

# Install client dependencies cd client npm install

# Install server dependencies cd/server npm install
```

## Folder Structure

```
|--assets/
   |__components/
    L__ app.css/
|__ app.jsx/                    # React frontend
   |__#index.css/
    |__ main.jsx/
      |__ .eslintrc.cjs/
        |__.gitignore/
|__index.html/
  |__index1.html/
    |__index2.html/             # Node.js backend
      |__index3.html/
      |__package-lock.json/
   |__package.json/
       |__readme.md/
          |__vite.config.js/
```

## Running the Application

To run the code on Terminal:

- Step 1: npm install
- Step 2: npm run dev

Access: http://localhost:5173/

# API Documentation

## Sign up page:

```
<body>
<html>
   <form action="index2.html" method="POST">
     <input type="text" placeholder="Full Name" required />
     <input type="email" placeholder="Email" required />
     <input type="password" placeholder="Password" required />
     <input type="password" placeholder="Confirm Password" required />
     <button type="submit">Sign Up</button>
   </form>
   <p>Already have an account? <a href="index2.html">Sign In</a></p>
  </div>
</body>

</html>
```

## Sign in page:

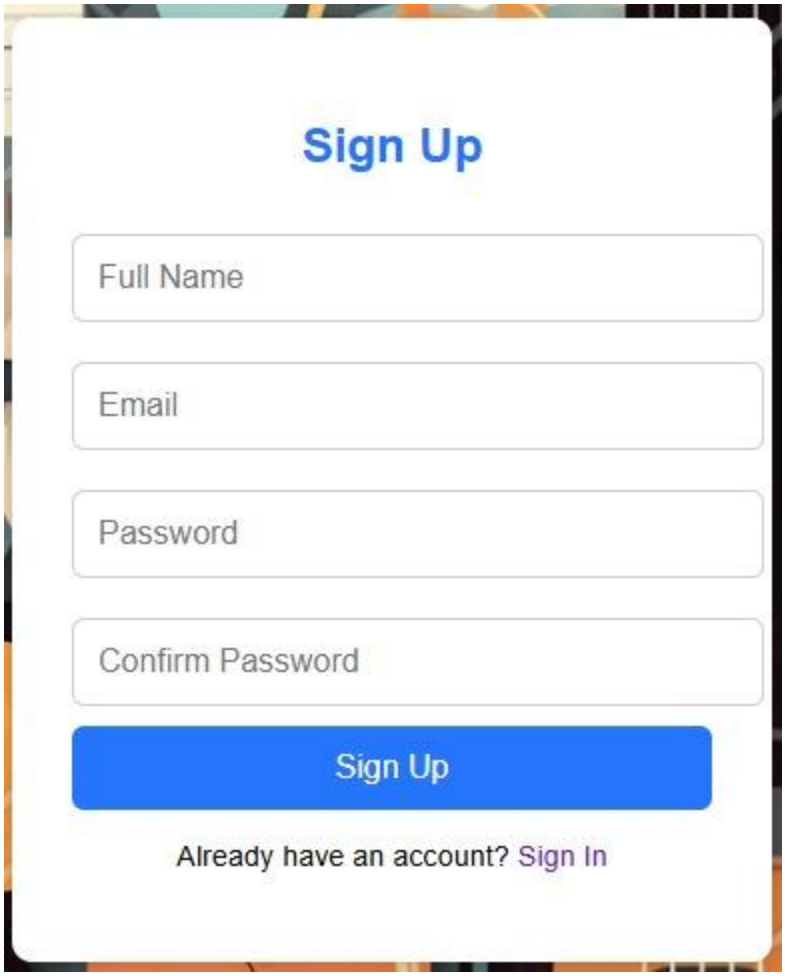
```
<body>
<html>
   <form action="index3.html" method="POST">
     <input type="email" placeholder="Email" required />
     <input type="password" placeholder="Password" required />
     <button type="submit">Sign In</button>
   </form>
   <p>Don't have an account? <a href="index3.html">Sign Up</a></p>
  </div>
</body>
</html>
```

# Home page:

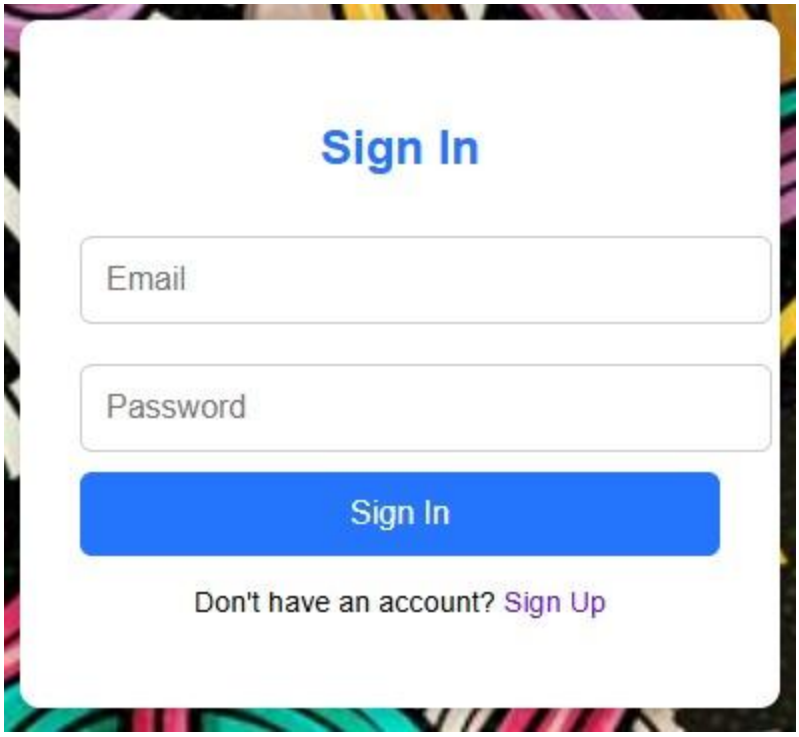
```
home page <!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css"
rel="stylesheet">
    <title>Music-Player</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

## User Interface

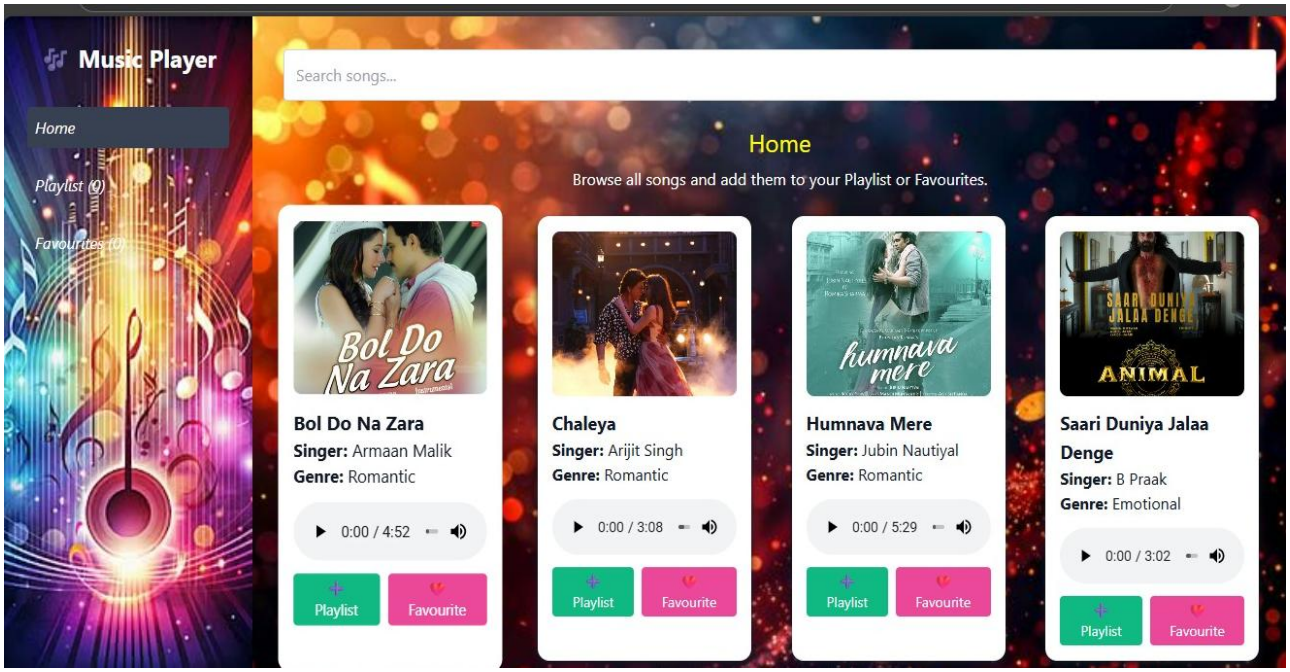
Sign up page:



# Sign in page:




# Home page:



# Demo link

Music Player Demo Video.mp4

# Future enhancements

- **Mood-Based Music Playback**

  Automatically adjusts music based on mood, using AI and biometric data.

- **Brain-Controlled Music**

  Use brain-computer interfaces (BCIs) to play, pause, or change music with thoughts.

- **Health & Therapy Integration**

  Music for mental health, sleep aid, stress relief, or physical rehabilitation.

- **Location-Aware Music**

  Music changes based on GPS location or surroundings (e.g., parks, cities, stores).

- **Virtual Concert & Social Listening**

  Join virtual reality concerts or listen to music in sync with friends online.