*Melody Music Streaming Website*

**Introduction :**

**Project title:**

“rhythmic tunes”

**Team Members:**

* + - * + **Poovarasu.S**
        + **Prakash.M**
        + **Mohan.V**
        + **Sivanesh**

Purpose:

Melody Music is a responsive web application designed to provide users with a seamless music streaming experience focused on melodic songs. The platform offers an intuitive interface for discovering, playing, and managing music with a design inspired by popular music services like Wynk Music.

Features:

* Interactive Music Player: Play/pause, next/previous, volume control, and progress seeking
* Song Library: Curated collection of melody songs with metadata
* Recently Played Tracking: Automatically tracks and displays recently played songs
* Responsive Design: Optimized for both desktop and mobile devices
* Visual Feedback: Interactive UI elements with hover effects and playing state indicators

Architecture

*Component Structure:*

The application is built with a single HTML file containing:

* Header Component: Navigation and search functionality
* Hero Component: Welcome section with call-to-action
* Player Component: Music playback controls and progress display
* Playlist Components: Song grid layouts with interactive cards
* Footer Component: Basic site information

*State Management:*

* + Audio Element: Controls playback functionality
  + JavaScript Variables: Manage current song index, playback state, and recent songs
  + DOM Manipulation: Updates UI elements based on state changes

***Routing:***

Single-page application with all functionality contained within one HTML file.

Setup Instructions

*Prerequisites:*

* + Modern web browser with JavaScript support
  + Internet connection (for loading external resources and music files)

*Installation:*

* Save the HTML code to a file with .html extension
* Open the file in any modern web browser
* No additional installation or configuration required

Folder Structure:

melody-music/

├── index.html # Main application file

├── (no additional files needed)

Running the Application

* Provide commands to start the frontend server locally
* **Frontend:** npm start in the client directory

*Key Components:*

* + *Audio Player*: Handles music playback using HTML5 audio element
  + *Progress Bar*: Custom implementation for seeking and displaying playback progress
  + *Song Cards*: Interactive elements that display song information and respond to clicks

*Reusable Components:*

* All UI components are designed with consistent styling and interaction patterns.

State Management:

Global State:

* + currentSongIndex: Tracks currently playing song
  + isPlaying: Play/pause state
  + recentSongs: Array of recently played song indices

Local State:

* UI states (hover effects, active states) are handled through CSS classes.

User Interface:

* + The interface features a dark theme with gradient accents, responsive card-based layout, and intuitive music controls. Key UI elements include:
  + Sticky music player that remains accessible while browsing
  + Animated transitions and hover effects
  + Visual feedback for currently playing song

Styling:

* + Custom CSS: No external frameworks used
  + CSS Grid & Flexbox: For responsive layouts
  + CSS Variables: For consistent color scheme
  + Gradient Effects: For visual appeal
  + Responsive Design: Adapts to different screen sizes testing
    - * Manual testing across different browsers
      * Responsive design testing using browser developer tools
      * Music playback functionality testing

Screenshots:

The application features a modern, dark-themed interface with:

* + Prominent music player controls
  + Grid layout of song cards with cover art
  + Clean typography and visual hierarchy

Known Issues:

* + Limited to six sample songs in the current implementation
  + Music files are loaded from external sources (Pixabay)
  + No persistent storage for user preferences or play history

Future Enhancements:

* + User authentication and personalized playlists
  + Expanded music library with categories/genres
  + Lyrics display feature
  + Social features (sharing, following)
  + Download options for premium users
  + Podcasts and audio content expansion
  + Dark/light theme toggle
  + Offline playback capability