**RhythmicTunes: Your Melodic Companion**

1. **Introduction**

**Project Title**: RhythmicTunes: Your Melodic Companion

**Team ID :** NM2025TMID32982

**Team Size :** 5

**Team Leader :** SETHUMAHARUBINI G

**Team member :** NITHYA SRI S

**Team member :** DIVYA R

**Team member :** JAYALAKSHMI R

**Team member :** DHEESHIKA R

## 2. Project Overview

Project Description:

Welcome to the future of musical indulgence – an unparalleled audio experience awaits you with our cutting-edge Music Streaming Application, meticulously crafted using the power of React.js. Seamlessly blending innovation with user-centric design, our application is set to redefine how you interact with and immerse yourself in the world of music.

Designed for the modern music enthusiast, our React-based Music Streaming Application offers a harmonious fusion of robust functionality and an intuitive user interface. From discovering the latest chart-toppers to rediscovering timeless classics, our platform ensures an all-encompassing musical journey tailored to your unique taste.

The heart of our Music Streaming Application lies in React, a dynamic and feature-rich JavaScript library. Immerse yourself in a visually stunning and interactive interface, where every click, scroll, and playlist creation feels like a musical revelation. Whether you're on a desktop, tablet, or smartphone, our responsive design ensures a consistent and enjoyable experience across all devices.

Say goodbye to the limitations of traditional music listening and welcome a world of possibilities with our React-based Music Streaming Application. Join us on this journey as we transform the way you connect with and savor the universal language of music. Get ready to elevate your auditory experience – it's time to press play on a new era of music streaming.

**Scenario-Based Intro:-**

Imagine stepping onto a bustling city street, the sounds of cars honking, people chatting, and street performers playing in the background. You're on your way to work, and you need a little something to elevate your mood. You pull out your phone and open your favorite music streaming app, "RythimicTunes."

With just a few taps, you're transported to a world of music tailored to your tastes. As you walk, the app’s smart playlist kicks in, starting with an upbeat pop song that gets your feet tapping. As you board the train, the music shifts to a relaxing indie track, perfectly matching your need to unwind during the commute.

**Target Audience:-**

Music Streaming is designed for a diverse audience, including:

**Music Enthusiasts:** People passionate about enjoying and listening Music Through out there free time to relax themselves.

## 3. Architecture

### Component Structure

* **App.jsx** - Root component managing routes and layout
* **Header.jsx** - Contains navigation and branding
* **Footer.jsx** - Displays copyright and links
* **HomePage.jsx** - Displays top and trending news
* **CategoryPage.jsx** - Shows news based on user-selected category
* **ArticleCard.jsx** - Reusable component to display individual news articles
* **AboutPage.jsx** - Information about InsightStream
* **SubscribePage.jsx** - Allows users to subscribe for updates

### State Management

* **Context API** is used for global state management, storing news data, user preferences, and subscriptions.
* Local state management within individual components for UI interactions.

### Routing

* **React Router** is used to handle navigation between pages:
  + / - Home Page
  + /about - About Page
  + /subscribe - Subscription Page
  + /category/:categoryName - Dynamic route for news categories

## 4. Setup Instructions

### Prerequisites

Ensure you have the following installed:

* Node.js (latest LTS version recommended)
* Git download
* npm or yarn

### Installation

* Clone the repository:
* Git clone https://github.com/Ashu1090/insightstream.git
* Navigate into the project directory:
* Cdinsightstream
* Install dependencies:
* npm install
* Start the development server:
* npm run dev

## 5. Folder Structure

insightstream/

│── src/

│ │── components/ # Reusable components (Header.jsx, Footer.jsx, ArticleCard.jsx)

│ │── pages/ # Page components (HomePage.jsx, AboutPage.jsx, SubscribePage.jsx)

│ │── assets/ # Images, icons, and styles

│ │── utils/ # Utility functions and custom hooks

│ │── context/ # Context API files for global state

│── public/ # Static files

│── package.json # Project dependencies and scripts

│── README.md # Project documentation

## 6. Running the Application

To start the frontend server locally, run:

npm run dev

This will launch the application at http://localhost:5173/.

## 7. Component Documentation

### Key Components

* **ArticleCard.jsx**: Displays a single news article with title, image, and description.
* **CategoryList.jsx**: Renders a list of available news categories.
* **SubscribeForm.jsx**: Handles user subscription input and submission.

### Reusable Components

* **Button.jsx**: Styled button used across the app.
* **Loader.jsx**: Displays a loading spinner while fetching data.

## 8. State Management

### Global State

* Managed via **Context API** to store and share news data and user preferences.

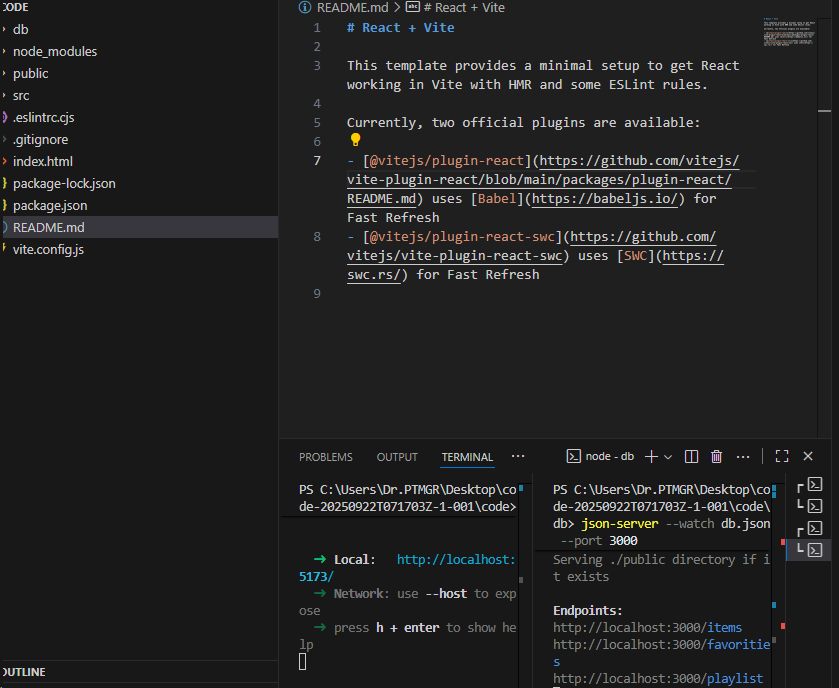
### Local State

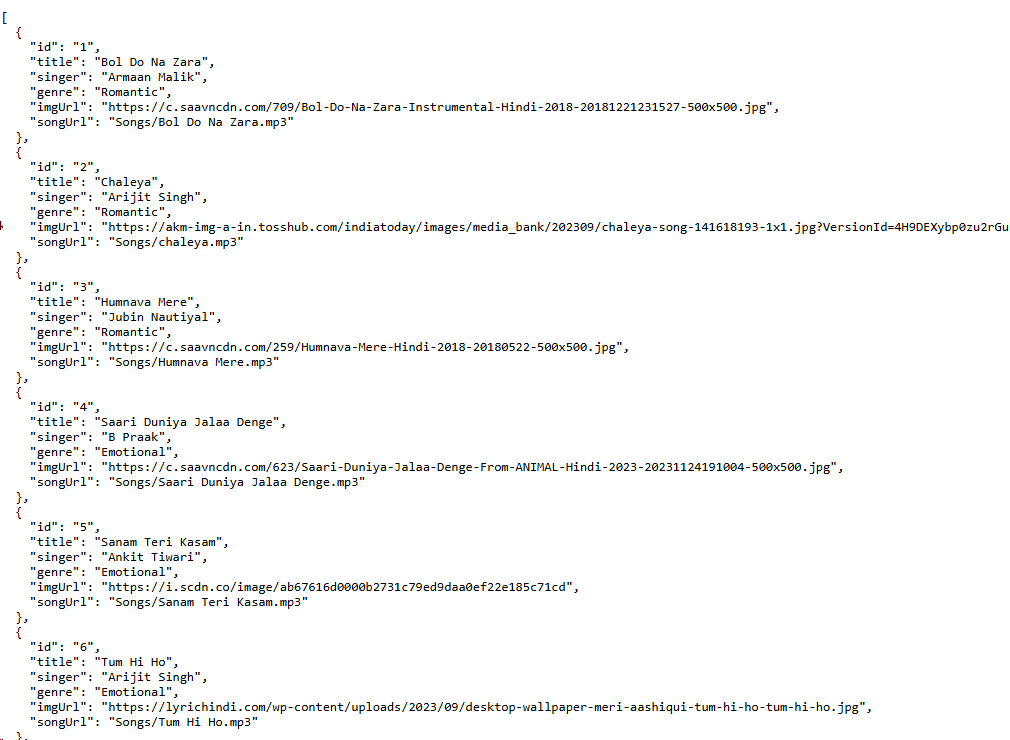
* Used within components to handle UI interactions such as toggles and input fields.

## 9. User Interface

Screenshots or GIFs showcasing:

**Homepage with trending news**



****

## 10. Styling

### CSS Frameworks/Libraries

* **Styled Components** for modular and dynamic styling.
* **CSS Modules** for scoped styles in individual components.

### Theming

* Dark mode support (if implemented in future versions)

## 11. Testing

### Testing Strategy

* **Jest & React Testing Library** for unit and integration tests.
* **Cypress** (future enhancement) for end-to-end testing.

### Code Coverage

* Code coverage reports generated using Jest.

## Demo

https://drive.google.com/file/d/1jCAaZsSEHGekJodRV74is3VkvhEqvDgV/view?usp=drive\_link

## 13. Known Issues

* Performance optimizations needed for large news datasets.
* Improve mobile responsiveness for small-screen devices.

## 14. Future Enhancements

* Implement dark mode for better accessibility.
* Add a search functionality for articles.
* Improve animation and transitions for a smoother user experience.