

PROJECT DOCUMENTATION

Rhythmic Tunes :Your Melodic Companion

- **INTRODUCTION:**
- **Project title:**
- **Rhythmic Tunes: Your Melodic Companion**
Team ID: NM2025TMID4228
- **Team Leader :**
HEMALATHA.V &
hemalathaviayagam2006@gmail.com
- **Team members :** JANANI.R& rjananiravi143@gmail.com
KAVIYA.P& Kaviyapalani719@gmail.com
JANANI.S& K55182981@gmail.com

1.PROJECT OVERVIEW :

SB Works is a platform designed to connect clients and freelancers, allowing for seamless project management from posting to completion. The core functionality is built around:

- **Project posting and bidding:** Clients can create projects and freelancers can submit bids.
- **Secure chat system:** A real-time communication channel for clients and freelancers.
- **Feedback and review system:** A mechanism for users to rate and review each other, building trust within the community.
- **Admin control panel:** An interface for administrators to manage users and content

2.Architecture

The application is built using a modern full-stack approach the following technologies:

- **Frontend:** The user-facing part is built with **React.js**, styled using **Bootstrap** and **Material UI** for a responsive and clean design.
- **Backend:** The server logic and API endpoints are managed by **Node.js** and **Express.js**.
- **Database:** **MongoDB** is used to store all the application data, including user information, project details, applications, and chat messages.

Setup Instructions

To get the project running, you will need to have the following prerequisites installed:

- Node.js

- MongoDB
- Git
- React.js
- Express.js - Mongoose
- Visual Studio Code

The installation process involves cloning the repository, navigating into the `client` and `server` directories, and running `npm install` in each to install the dependencies.

Running the Application

Once all dependencies are installed, you can start the application by running the following commands in their respective directories:

- For the frontend: `cd client` and then `npm start`.
- For the backend: `cd server` and then `npm start`.

You can then access the application in your web browser by visiting <http://localhost:3000>.

6. Folder structure:

A well-thought-out folder structure is essential for a large-scale application like "Insight Stream." It helps organize the codebase, makes it easier for developers to find and manage files, and is crucial for the long-term maintainability and scalability of the project.

Root Directory

This is the top level of your project. It contains configuration files and the main folders for the front-end and back-end.

- `/client`: All front-end code for the web or mobile application (e.g., built with React, Vue, or a native framework).
- `/server`: All back-end code for the API, database models, and business logic.
- `/shared`: A folder for code that is used by both the client and server, such as data models, utility functions, or API schemas.
- `package.json`: Manages project dependencies and scripts.
- `.gitignore`: Specifies files and folders to be ignored by version control.
- `README.md`: Provides documentation for the entire project.

7. API Documentation

A well-structured authentication API documentation is vital for a platform like "Rhythmic Tunes" with user chat features. It serves as the single source of truth for developers, ensuring they can

securely integrate user login, registration, and session management, which are all prerequisites for enabling chat.

For a chat-enabled news platform, the authentication documentation would not only cover standard user authentication but also the specific requirements for real-time communication. This would include:

- **User Identification for Chat:** The core purpose is to obtain a unique, verified user ID or token that the chat service can use to identify who is sending and receiving messages.
- **Permissions and Roles:** The documentation would explain how to retrieve a user's role (e.g., "subscriber," "moderator," "admin") from the authentication response, which is crucial for managing chat permissions. For example, only moderators might be able to delete messages or ban users in a chat room.
- **Session Management for Real-time:** Real-time chat requires persistent connections, so the documentation would detail how to handle long-lived sessions and how to use refresh tokens to keep a user authenticated without requiring them to log in again.
- **Security Best Practices:** The documentation would include specific security protocols for chat, such as how to prevent injection attacks in chat messages and how to handle data encryption for messages at rest and in transit.

8. Authentication

The platform uses **JWT-based authentication** for secure user logins, with middleware in place to protect private routes.

- **Provide a Personalized Experience:** By knowing who the user is, the platform can access their preferences, saved articles, custom topic streams, and past reading history. This allows the system to deliver a truly personalized "Rhythmic Tunes" rather than a generic news feed.
- **Enable Premium Access:** For subscription-based news models, authentication gates access to premium content, exclusive analyses, and advanced features like data visualizations or in-depth reports.
- **Ensure Data Security:** Authentication protects the user's personal data, such as their reading habits, demographic information, and any payment details, from unauthorized access.
- **Enhance Community Features:** If the platform includes features for commenting, sharing, or curating content, authentication ensures that contributions are tied to a verified user, which helps maintain a respectful and trustworthy community.

9. User interface:

An "Rhythmic Tunes" user interface for navigating the news landscape would likely be designed to help users move beyond a simple, chronological feed of headlines. Instead, it would focus on

providing a more structured and interactive experience. The core elements of such a UI would typically include:

- **A Dashboard or Central Hub:** This would serve as the user's home base, offering a personalized overview of the news. It might feature a curated selection of top stories, a summary of news from their preferred categories, and a quick-view of trending topics.
- **Dynamic Search and Filtering:** Unlike a basic search bar, this would allow for advanced queries and real-time filtering. Users could search by keyword, source, topic, or even sentiment, and the results would update instantly.
- **Interactive Visualizations:** Instead of just text, the interface could use charts, graphs, and network maps to show relationships between stories, track the evolution of a topic over time, or highlight the most influential sources. This would help users spot connections and patterns they might otherwise miss.
- **Personalization and Customization:** The UI would learn from user behaviour to suggest relevant articles and categories. It would also give users the ability to manually customize their feed, save articles for later, and create their own "streams" based on specific interests.

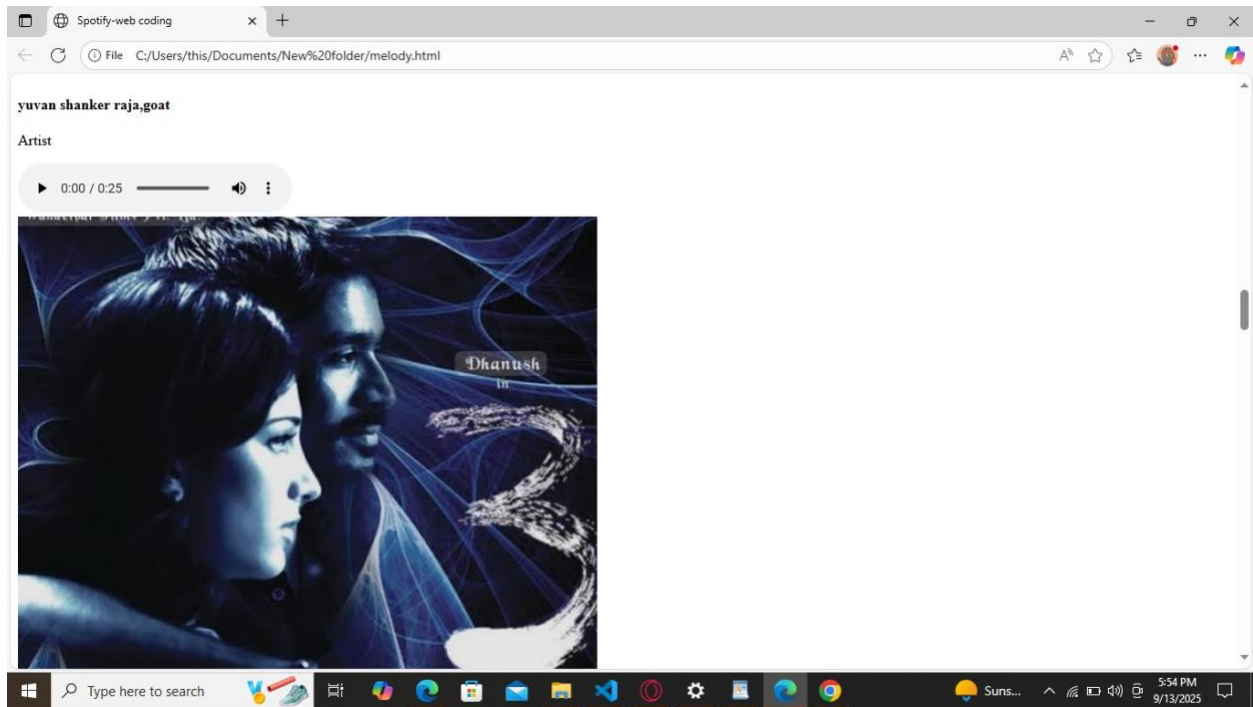
10. Testing

Testing for a chat-enabled news platform like "Rhythmic Tunes" requires a robust strategy that goes beyond simple functionality checks. It's about ensuring the security, reliability, and performance of every user interaction, particularly those involving real-time communication.

Here's a breakdown of the key testing areas for such a platform:

- **Authentication and Authorization Testing:** This is the foundation. You must verify that the authentication API works as documented. This includes testing for:
 - **Successful Login:** Both with valid and invalid credentials.
 - **MFA Flows:** Ensuring multi-factor authentication works correctly. ○
 - **Validation:** Confirming that the access tokens and refresh tokens are correct generated, have the right permissions, and expire as they should.
 - **Role-Based Access:** Testing that users with different roles (e.g., a standard user versus a moderator) can only access the features and chat rooms they are authorized to.

11. Screenshot;



Demo link

<https://drive.google.com/file/d/1ecp7vzfMiy4tUHJ4993KQChEhbcziopI/view?usp=drivesdk>

12. Known issue:

In an “Rhythmic Tunes” news and chat platform, a “known issue” is a problem or bug that has been identified, documented, and is currently being worked on by the development team. These issues can range from minor UI glitches to significant performance and security flaws.

Here are some common known issues you might encounter in a complex, integrated system like this:

Data and Synchronization Issues

This is a critical area for a platform that combines real-time news with chat.

- News Feed Latency: A known issue could be a delay in news articles appearing in the feed. For instance, a major news event breaks, but the platform's API takes

13. Future enhancement:

Future enhancements for a news platform like "Rhythmic Tunes" would likely focus on leveraging

AI to create a more personalized and intelligent user experience.

Here are some key areas for future development:

- Generative AI for Content Summarization: Instead of just showing the full article, the platform could use AI to generate concise, abstractive summaries. This would allow users to quickly grasp the core information of multiple articles without having to read them in full.

- Enhanced Personalization: Beyond simple topic selection, the system could learn from a user's reading behavior (e.g., time spent on a page, scrolling speed, and even sentiment expressed in chats) to create a much more nuanced user profile. This could lead to hyper-personalized feeds that anticipate interests the user hasn't explicitly stated.

- Real-Time Data Visualization: Integrating live data streams and creating dynamic charts or graphs to visualize the evolution of a breaking news story. This would help users see the "big picture" of an event as it unfolds, such as tracking public sentiment or the geographic spread of a story.

- Interactive Chat with AI: The chat function could be enhanced with an AI assistant that can answer questions about articles, find related stories, or even fact-check

Claims made within the chat conversation in real-time.

Several minutes to pull the story, leading to a frustrating user experience where the

News is not “live.”

- Chat Message Sync Errors: Messages in a chat channel don't appear in the correct

Order for all users, or a message sent from a mobile app doesn't show up on the web

Version right away. This is often caused by race conditions or poor handling of realtime data from web sockets.

- Duplicate Content: The news aggregation system pulls the same article from

Multiple sources, leading to a cluttered and redundant news feed. This is a common

Problem with web scraping