

1. Introduction

Project Title: Insight Stream – Navigate the News Landscape

Team ID: NM2025TMID28741

Team Leader: Srinithi. R (srinidhiravi20@gmail.com)

Team Members:

- Devika. R (devika24.rajendran@gmail.com)
- Keerthana. R (kr6712460@gmail.com)
- Shiny Hema. K (stellshiny.4088@gmail.com)

2. Project Overview

Purpose: Insight Stream is a platform designed to help users navigate the fast-changing news landscape. It curates, categorizes, and personalizes news content for better accessibility and engagement.

Features:

- Personalized news feed based on user preferences
- Real-time trending news updates
- Categorization by topics, region, and source
- Search and filter options for news articles
- Bookmarking and sharing features
- Admin control panel for content management

3. Architecture

Frontend: React.js with Tailwind CSS and Material UI for responsive design.

Backend: Node.js and Express.js for API handling and server-side logic.

Database: MongoDB for storing user profiles, preferences, news data, and bookmarks.

API Integration: News APIs (e.g., NewsAPI, Google News RSS) for fetching live updates.

4. Setup Instructions

Prerequisites: Node.js, MongoDB, Git, React.js, Express.js, Visual Studio Code

Installation Steps:

1. Clone the repository using: git clone
2. Install frontend dependencies: cd client @ npm install
3. Install backend dependencies: cd ../server @ npm install
4. Setup environment variables (API keys, DB connection string)

5. Run the frontend: npm start (inside client folder)
6. Run the backend: npm start (inside server folder)

5. Folder Structure

InsightStream/ |-- client/ # React frontend | |-- components/ | |-- 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> in browser

7. API Documentation

User: /api/user/register, /api/user/login

News: /api/news/fetch, /api/news/search

Bookmarks: /api/bookmarks/add, /api/bookmarks/remove

Admin: /api/admin/manage

8. Authentication

JWT-based authentication ensures secure login and protects private routes. Middleware is used for user role management (admin vs normal user).

9. User Interface

- Landing Page with top headlines and categories
- Personalized Dashboard for users
- Search and Filter Page
- Admin Panel for managing news sources and users

10. Testing

Manual and automated testing using Postman, Jest, and Chrome Dev Tools.

11. Screenshots or Demo

Screenshots of UI and demo video will be included in the final report.

12. Known Issues

- API rate limits may restrict real-time data access
- Mobile responsiveness requires further optimization

13. Future Enhancements

- AI-driven news summarization
- Voice-assisted news reading
- Multilingual news support
- Advanced analytics for trending patterns

