PROJECT DOCUMENTATION:

Project Title: InsightStream: Navigate the News Land Scape

1.INTRODUCTION PROJECT TITLE: InsightStream: Navigate the News Land Scape

*** Team Id:** NM2025TMID35076

*** Team Leader:** Afreen Benasir.M

(afreenbenasir@gmail.com)

*** Team Members:**

1)Ajitha.A

(ajithaarumugam2006@gmail.com)

2)Swetha.R

(swetharamesh729@gmail.com)

3)Rihana.M

(rihana292929@gmail.com)

2. Project Overview

Purpose:

InsightStream is a web-based news aggregation and recommendation platform. It allows users to explore trending news, filter by categories, save articles, and personalize their news feed for a streamlined reading experience.

Features:

- Browse latest news by category (Politics, Tech, Sports, Entertainment, etc.)

- Search news by keywords

- Save favorite articles for later reading - Personalized recommendations based on reading history

- Responsive, modern UI

3. Architecture

- Frontend: React.js with Tailwind CSS / Bootstrap

- Backend: Node.js / Express (REST API)

- Database: MongoDB (for users, saved articles, preferences)

- External API: News API (for real-time news articles)

4. Technologies Used

- React.js - Node.js / Express

- MongoDB - Tailwind CSS / Bootstrap

- News API

- Git & VS Code

5. Installation Steps

```
# Clone the project
git clone https://github.com/your-repo/InsightStream.git
```

```
cd InsightStream
# Install dependencies
npm install

# Start the React frontend
npm run dev
```

In a separate terminal (backend):

```
cd server
npm install
npm start
```

Access at: http://localhost:5173

6. Folder Structure

```
InsightStream/
■-- client/ # Frontend (React)
■ ■-- src/
■ ■-- assets/ # Images, icons
■ ■-- components/
■ ■-- pages/
■ ■-- App.jsx

■ ■-- main.jsx
■-- server/ # Backend (Node.js + Express)
■ ■-- routes/
■ ■-- models/
■ ■-- controllers/
■ ■-- server.js
■-- README.md
```

7. Running the Application

Frontend:

```
npm run dev
```

Backend:

```
npm start
```

Access: http://localhost:5173

8. API Documentation

Articles:

- GET /api/articles → Get all articles
- GET /api/articles/:id → Get article by ID
- POST /api/articles/save → Save article
- DELETE /api/articles/:id → Delete saved article

Users:

- POST /api/users/register → Register new user
- POST /api/users/login → Login
- GET /api/users/:id → Get user details

Recommendations:

- GET /api/recommendations/:userId → Get personalized news feed

9. Authentication

- Implemented using JWT (JSON Web Token)
- Passwords stored securely with hashing

10. User Interface

- Landing Page (Trending News)
- Sidebar Navigation (Categories: Tech, Sports, etc.)
- Article Cards (title, image, description, source, save button)
- Personalized Feed (AI-driven recommendations)

- Saved Articles Page

11. Testing

- Manual Testing: Verified news fetching, search, and article saving

- Tools: Postman, Chrome Dev Tools, MongoDB Compass

12. Demo Video

https://drive.google.com/file/d/1y29XxIhutgXVxoQmnxtPPvdD5rK1N7Zg/view?usp=drive_link

13. Known Issues

– API request limits with free News API key

- Basic personalization model (not fully AI-driven yet)

- Saved articles only accessible online

14. Future Enhancements

- Advanced AI-based recommendations (NLP sentiment analysis)

- Mobile App (React Native)

- Offline reading mode

- Multi-language news support

- Push notifications for