Project Documentation

Insight stream: Navigate the News Landscape

1. Introduction

- **Project Title:**   Insight stream : Navigate the news landscape

- **Team ID:**   NM2025TMID36607

- **Team Leader:**   RAJ KUMAR M   &   manivannan30091980@gmail.com

- **Team Members:**

  - DHANANCHEZIAN M & dhana2006ms@gmail.com
  - DINESH KUMAR S   &   sdineshk064@gmail.com
  - DRISHA K & kdrisha93@gmail.com

2. Project Overview

- **Purpose:** Insight Stream provides a streamlined platform for users to navigate the news landscape efficiently. It enables readers to explore, filter, and interact with news articles in real time, connecting them with the most relevant and credible sources**.**

- **Features:**

  ✓ News aggregation and categorization

  ✓ Real-time updates

- ✓ Personalized recommendations
- ✓ Search and filtering options
- ✓ Secure user authentication
- ✓ Feedback and bookmarking system
- ✓ Admin control panel

3. Architecture

- **Frontend (React.js + Bootstrap/Material UI)**

Provides an interactive user interface for browsing, searching, and bookmarking news.

- **Backend (Node.js + Express.js)**

Handles APIs, authentication, and business logic, connecting the frontend with the database.

- **Database (MongoDB)**

Stores user accounts, news articles, bookmarks, and feedback in a scalable NoSQL structure.

4. Setup Instruction

- **Prerequisites**:

  - ➢ **Node.js** – Runtime environment for executing JavaScript on the   server side.
  - ➢ **MongoDB** – NoSQL database for storing users, news articles, bookmarks, and feedback.

- **GitHub** – Version control system to clone and manage the project repository.
- **React.js** – Frontend framework for building an interactive user interface.
- **Express.js** – Backend framework for handling APIs and server logic.
- **Mongoose** – ODM library to interact with MongoDB using schemas.
- **Visual Studio Code** – Recommended IDE for writing and managing project code.

Installation Steps:

1. Clone repository

2. Install dependencies using npm install

3. Start the development server using npm start

5. Folder Structure

```
Insight stream/
|-- 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

7. API Documentation

•**User APIs:**

POST /api/user/register → Register a new user

POST /api/user/login → User login and JWT token generation

GET /api/user/:id → Fetch user profile details

•**News APIs:**

GET /api/news → Get all news articles (with filters like category, date, keyword)

GET /api/news/:id → Get details of a specific news article

POST /api/news/create → Add a news article (Admin only)

•**Bookmark APIs:**

POST /api/bookmark → Add a bookmark for a logged-in user

GET /api/bookmark/:user ID → Retrieve all bookmarks of a user

DELETE /api/bookmark/:id → Remove a specific bookmark

8. Authentication

•**JWT Authentication**: Users log in with email & password, and receive a token.

•**Middleware Protection**: Token is verified before accessing protected routes.

•**Role-Based Access:** Users can read/bookmark news, Admins can manage articles.

9. User Interface

•**Landing Page** – Homepage displaying top and trending news categories.

•**User Dashboard** – Personalized news feed, search, filters, and bookmarks.

•**News Details Page** – Full article view with feedback and sharing options.

•**Bookmarks Page** – Saved articles for quick access.

•**Admin Panel** – Manage news sources, categories, and user activity.

10. Testing

•**Manual Testing** – Core features (login, news, bookmarks).

•**API Testing** – Verified with Postman.

•**UI Testing** – Checked navigation & responsiveness via Chrome DevTools

11. Demo Video Link

https://drive.google.com/file/d/14h613gYLwmCHjHzjG-6Guk8el16offUy/view?usp=drivesdk

12. Known Issues

- Performance slowdown when handling large volumes of news data.
- No offline mode or caching support.

- Limited error handling for invalid requests.

13. Future Enhancements

•AI-powered recommendations for personalized news feeds.

•Sentiment analysis to classify articles as positive, negative, or neutral.

•Mobile app version for Android and iOS

•Multi-language support for global users.

•Push notifications for breaking news alerts.