Project Documentation

***Insight Stream:*** *Your Intelligent Data Companion*

***1. Introduction***

* ***Project Title:*** *Insight Stream: Your Intelligent Data Companion*
* ***Team ID:*** *NM2025TMID40002*
* ***Team Leader:*** *SWEATHA R [sweatha.r1505@gmail.com]*
* ***Team Members:***

*– VASANTHI M [sac2427csc5196@ssacollegechennai.com]*

*– TAMIL SELVI S [sac2427csc5485@ssacollegechennai.com]*

*– VIDHUNA N [sac2427csc5516@ssacollegechennai.com]*

***2. Project Overview***

***Purpose:*** *Insight Stream is a data analytics and visualization platform that allows users to gather, process, and interpret insights from structured and unstructured data. It provides real-time dashboards, predictive analysis, and simplified decision-making support.*

***Features:***

*– Upload datasets in multiple formats (CSV, JSON, Excel)*

*– Real-time data visualization with interactive charts*

*– Dashboard creation and management*

*– Predictive analytics using AI/ML models*

*– User accounts with saved dashboards and reports*

*– Admin panel for data governance and access control*

***3. Architecture***

* ***Frontend:*** *React.js with Material-UI for dynamic dashboards and visualizations*
* ***Backend:*** *Node.js + Express.js for APIs and server logic*
* ***Database:*** *MongoDB (stores users, datasets, dashboards, analytics reports)*
* ***Analytics Engine:*** *Python-based microservices for ML and data processing*

***4. Setup Instructions***

* ***Prerequisites:***

*– Node.js*

*– MongoDB*

*– Git*

*– Visual Studio Code*

*– Python 3.x with required libraries (pandas, scikit-learn, matplotlib)*

* ***Installation Steps:***

*# Clone the repository git clone*

*# Install client dependencies cd client npm install*

*# Install server dependencies cd ../server npm install*

*# Install analytics dependencies cd ../analytics pip install -r requirements.txt*

***5. Folder Structure***

*insight-stream/*

*├── client/ # React frontend*

*│ ├── components/*

*│ ├── pages/*

*│ └── assets/*

*├── server/ # Node.js backend*

*│ ├── routes/*

*│ ├── models/*

*│ ├── controllers/*

*│ └── middleware/*

*├── analytics/ # Python ML/AI services*

*│ ├── models/*

*│ ├── scripts/*

*│ └── notebooks/*

*└── README.md*

***6. Running the Application***

* ***Frontend:*** *cd client npm start*
* ***Backend:*** *cd server npm start*
* ***Analytics Engine:*** *cd analytics python run.py*
* ***Access:*** *Visit http://localhost:5173*

***7. API Documentation***

* ***User:***

*POST /api/user/register – Create account*

*POST /api/user/login – Log in*

* ***Datasets:***

*POST /api/datasets/upload – Upload dataset*

*GET /api/datasets/:id – Get dataset details*

* ***Dashboards:***

*POST /api/dashboards/create – Create dashboard*

*GET /api/dashboards/:userId – Retrieve dashboards*

***8. Authentication***

* *JSON Web Token (JWT) for login sessions*
* *Role-based access control for users and admins*

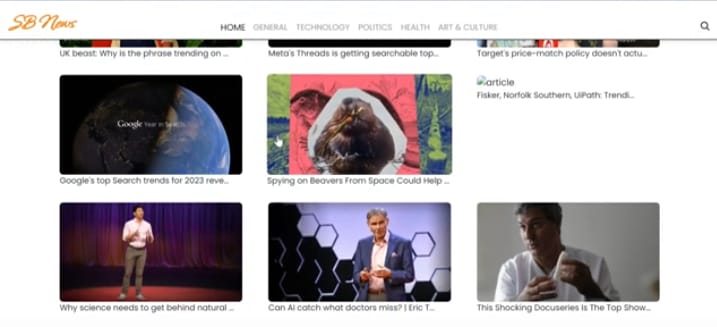
***9. User Interface***

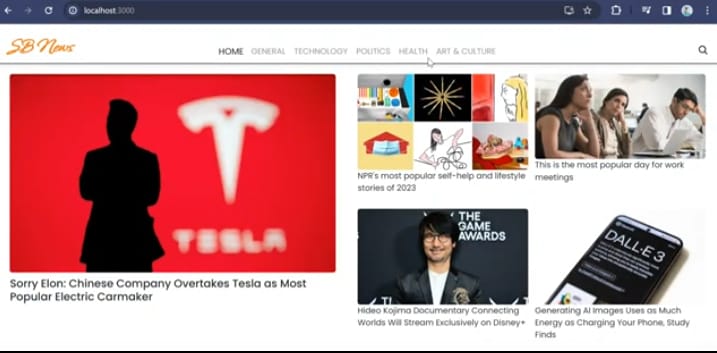
* *Landing Page with featured analytics use-cases*
* *Dataset Upload Page*
* *Interactive Dashboard with charts and graphs*
* *Admin Panel for user and dataset management*

***10. Testing***

* *Manual testing for dataset upload, chart rendering, and dashboard creation*
* *Tools: Postman, Chrome DevTools, Jest for frontend tests, PyTest for analytics engine*

***11.Screenshot or Demo***





***12. Known Issues***

* *Large dataset uploads may take extra time*
* *Real-time predictive analysis limited to small datasets*
* *Occasional rendering delay for complex charts*
* *Mobile UI optimization in progress*

***13. Future Enhancements***

* *Integration with cloud data sources (AWS S3, Google BigQuery)*
* *AI-driven automated insights*