Project Documentation

Insightstream: Navigate the News Landscape

Introduction

• **Project Title:** Insightstream: Navigate the News Landscape

• **Team ID:** NM2025TMID39963

• **Team Leader:** DURGA DEVI.M [sac2427csc5522@ssacollegechennai.com]

• **Team Members:**

_ILAKKIYA P [sac2427csc5465@ssacollegechennai.com]

_HARINI S [sac2227csc5368@gssacollegechennai.com]

_DILLI RANI E [sac2427csc5448@ssacollegechennai.com]

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

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

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
```

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
```

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

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

Authentication

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

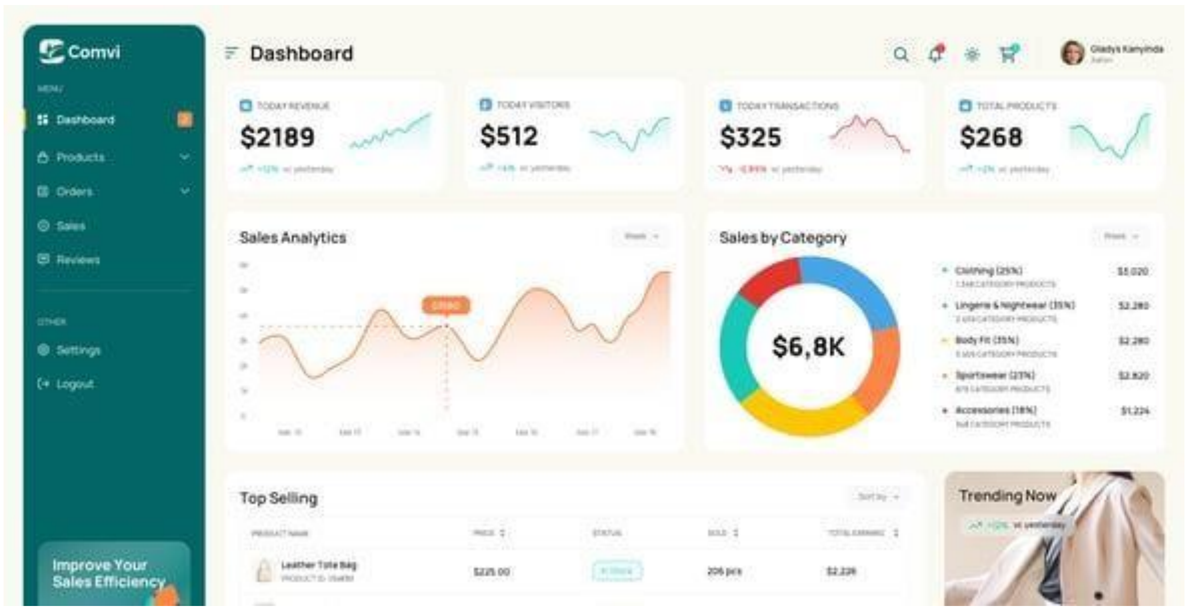
User Interface

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

Testing

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

Screenshots or Demo



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

Future Enhancements

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