Insight Stream – Frontend Development with React.js

1. Introduction

Project Title: Insight Stream

TEAM LEADER:

SAVITHA JENIFER A

TEAM MEMBER:

SOUNDARYA M

SATHYA K

RAMYA SRI R

SATHIYA N

2. Project Overview

Purpose:

Insight Stream is a platform visualization of data streams. The frontend aims to offer a clean, interactive, and responsive interface for monitoring live data.

* Features:
* Real-time data dashboard
* Interactive charts and graphs
* User authentication (login/signup)

* Data filtering and search
* Notifications/alerts for important events

* Responsive design (mobile & desktop)

3. Architecture

* Component Structure:

* App.js – Main entry point
* Navbar – Navigation menu
* ashboard – Displays live inDsights & widgets
* ChartComponent – Reusable chart component
* NotificationPanel – User alerts and messages
* Auth – Login/Register pages
* Profile – User profile and settings

State Management:

* Global State: Redux Toolkit (for user data, notifications, and live stream updates)

* Local State: React useState for component-level interactions

Routing:

* react-router-dom used
* / → Landing Page
* /dashboard → Main dashboard

* /login → Login page
* /register → Registration page

* /profile → User profile

4. Setup Instructions

Prerequisites:

* Node.js v18+
* npm or yarn

Installation:

* # Clone repository
* git clone https://github.com/your-repo/insight-stream.git
* # Navigate to project folder
* cd insight-stream/client
* # Install dependencies
* npm install
* # Run application
* npm start

5. Folder Structure

* client/
* │── public/ # Static files
* │── src/
* │ ├── assets/ # Images, icons, styles
* │ ├── components/ # Reusable components (Navbar, ChartComponent)
* │ ├── pages/ # Page-level components (Dashboard, Auth, Profile)
* │ ├── store/ # Redux store setup
* │ ├── hooks/ # Custom hooks
* │ ├── utils/ # Utility functions
* │ └── App.jS

6. Running the Application

* cd client
* npm start
* The app will run on http://localhost:3000.

7. Component Documentation

Key Components:

* Dashboard: Displays insights and charts
* ChartComponent: Accepts data, type, and title as props
* NotificationPanel: Shows real-time alerts

Reusable Components:

* Button
* Modal
* Card

8. State Management

Global State:

* Managed via Redux Toolkit (user, notifications, streamData).

Local State:

* Form inputs, modal visibility, toggle states managed with useState.

9. User Interface

* Dashboard with line, bar, and pie charts
* Real-time notification panel
* Mobile-friendly navigation drawer

10. Styling

* CSS Framework: Tailwind CSS

* Icons: Lucide React
* Theming: Dark/Light mode toggle

11. Testing

Strategy:

* Jest + React Testing Library → Unit & integration testing
* Cypress → End-to-end testing

Code Coverage:

* Configured via Jest coverage report

12. Screenshots / Demo

* Include UI screenshots (Dashboard, Login, Notifications)
* Provide link to live demo (if hosted on Vercel/Netlify)

13. Known Issues

Occasional delay in chart rendering with large datasets

* Dark mode styling inconsistencies on mobile

14. Future Enhancements

* AI-based predictive analytics
* Export dashboard reports as PDF/Excel
* Enhanced real-time collaboration (multiple users)
* Voice command integration