SEC 160: FULL STACK DEVELOPMENT

INSIGHT STREM

Submitted by

**KALYAN [AP23110010268]**

**VIVEK [AP23110010277]**

**JATHIN [AP23110011215]**

**HARSHA [AP23110011366]**

**SEM: V**

**SECTION: N**

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

SCHOOL OF ENGINEERING AND SCIENCES



**SRM University-AP, Neerukonda, Andhra Pradesh 522240**

**December 2025**

Introduction:-

**InsightStream** is a modern news web application built using React. Its main purpose is to bring news from different categories—like Technology, Sports, Business, Entertainment, and General—into one easy-to-use platform.

The app gives users a smooth and clean reading experience with a dark theme and modern design. Users can browse news in different layouts, view full articles, and explore updates from multiple sources.

InsightStream also includes **login and signup features**, so each user gets a personalized experience. Once logged in, users can save articles for later, check what they have read before, and manage their profile.

Overall, InsightStream makes news reading simple, fast, and personalized by combining multiple news sources and giving users helpful features to stay updated.

Scenario-Based Intro:-

Imagine you want to read the latest news, but you have to open many websites—one for technology, another for sports, another for business, and so on. It becomes confusing, time-consuming, and hard to keep track of what you have already read.

To solve this problem, **InsightStream** was created.

InsightStream brings all important news into one place. Instead of switching between many news sites, users can simply log in and see all categories—Technology, Sports, Entertainment, Business, and General—in a clean and easy layout. Users can save articles for later, check their reading history, and enjoy a smooth, modern interface.

In this way, InsightStream makes news reading simple, fast, and personalized by giving users everything they need on one platform.

Target Audience:-

**Who is it for:**  
InsightStream is made for people who like to read news regularly and want it all in one place. This includes students, professionals, developers, and tech enthusiasts aged 18-45. These users want a platform where they **can see news from different categories, save articles, track what they have read**, and enjoy a modern, clean design.

**Categories Covered:**

Technology, Sports, Entertainment, Business, and General news.

**Project Goals and Objectives:**

**Main Goal:**  
To create an easy-to-use news app that shows real-time news and lets users save, track, and personalize their reading experience.

**Key Objectives:**

* Show news from NewsData.io in real-time across multiple categories
* Let users create accounts and access personal dashboards
* Allow users to save/bookmark articles for later
* Keep track of user reading history
* Build a responsive, modern interface with glassmorphism/neomorphism
* Ensure fast performance using React 19 and Vite
* Keep the code clean and organized

**User-Friendly Interface:**

## Design Idea:

The app has a modern look with soft shadows and glass-like effects (glassmorphism/neomorphism). The dark theme makes reading easy on the eyes.

**Key Features:**

* Switch between **grid and list views**
* **Save articles** and view reading history
* Filter news by **category**
* Smooth animations and simple navigation

**Accessibility:**

* Works on **all devices** (mobile, tablet, desktop)
* Easy to use with **clear visuals**

Modern Tech Stack:

InsightStream is built using **React 19** and **Vite**, which makes the app fast and easy to develop. **React Router DOM v7** is used to move between pages and protect user-only routes. The design uses **Tailwind CSS** with glass-like and soft shadow effects for a modern look. **React Context API** and **localStorage** keep track of user login and app state. **Axios** fetches news from the NewsData.io API, and **JSON Server** stores local user data. The app also uses **custom components**, **Lucide React icons**, and **responsive cards** to make the interface clean and easy to use.

Key Features:

* **User Authentication System:** Secure login, signup, and session management with protected pages.
* **Real-Time News Aggregation:** Fetches live news from multiple categories with filters and error handling.
* **Article Management:** View articles in grid/list format with automatic reading history tracking.
* **Bookmark System:** Save and manage articles with one-click bookmarks across sessions.
* **Reading History:** Automatically tracks and shows recently read articles.
* **Personalized Dashboard:** Shows welcome message, bookmarks, history, and category filters.
* **User Profile Management:** Displays user stats and allows profile updates and password changes.
* **Navigation & Layout:** Fixed, responsive navbar with active highlights and easy logout.
* **Modern UI/UX Design:** Dark theme with glassmorphism/neomorphism, smooth animations, and responsive design.
* **News Page Features:** Latest headlines with pagination, infinite scroll, and bookmark support.

Prerequisites & Setup:

**System Requirements:**

* **Node.js:** Version 16 or higher (recommended 18 or 20) to run the app.
* **npm:** Comes with Node.js for managing packages.
* **Web Browser:** Latest Chrome, Firefox, Edge, or Safari.
* **Internet:** Needed to fetch news from NewsData.io and install packages.

**External Services & API Keys:**

* **NewsData.io API Key:** Required to get news articles. Sign up at [NewsData.io](https://newsdata.io/) and add the key in Home.jsx, News.jsx, or a .env file (REACT\_APP\_NEWSDATA\_KEY=your\_api\_key).

**Backend Services:**

* **JSON Server:** Needed for user authentication and storing user data.
  + Install globally: npm install -g json-server
  + Run server: npx json-server --watch data/db.json --port 3000

**Development Tools (Optional):**

* Code Editor like **VS Code** or WebStorm
* **Git** for version control
* Browser **DevTools** for debugging

**Installation Steps:**

1. Clone or download the project repository.
2. Go to project folder: cd InsightStream-main
3. Install dependencies: npm install
4. Start JSON Server (in a separate terminal): npx json-server --watch data/db.json --port 3000
5. Add your NewsData.io API key in the code or .env file.
6. Start development server: npm run dev
7. Open in browser at http://localhost:5173

**Important Notes:**

* JSON Server must run on port 3000 for login and user data.
* NewsData.io API key is required for fetching news.
* Make sure ports 3000 and 5173 are free.
* All dependencies are listed in package.json and installed with npm install.

Project structure:



Project Flow:

Project demo:

Demo link: <https://drive.google.com/drive/folders/1hfIVg8s2OxhViSHutpX8Ob-Oc99Q8zk8?usp=sharing>

Use this code: <https://github.com/harshavardhan3108/InsightStream---ReactProject/tree/main/InsightStream-main>

Milestone 1: Project Setup and Configuration:

****Focus:**** Setting up the project structure, environment, and initial configuration.

* **Application Initialization:** src/main.jsx renders the app, src/App.jsx handles routing, and LoginToggle.jsx manages login/register toggle.
* **Authentication Flow:** Login.jsx and Signin.jsx handle user login and registration using **JSON Server** (data/db.json) and **localStorage**.
* **Protected Route Setup:** ProtectedRoute in App.jsx secures pages, and DashboardLayout.jsx wraps protected pages with the Navbar.
* **Data Configuration:** Connection setup for **localStorage** (sessions), **JSON Server** (user data), and **NewsData.io API** (news articles).

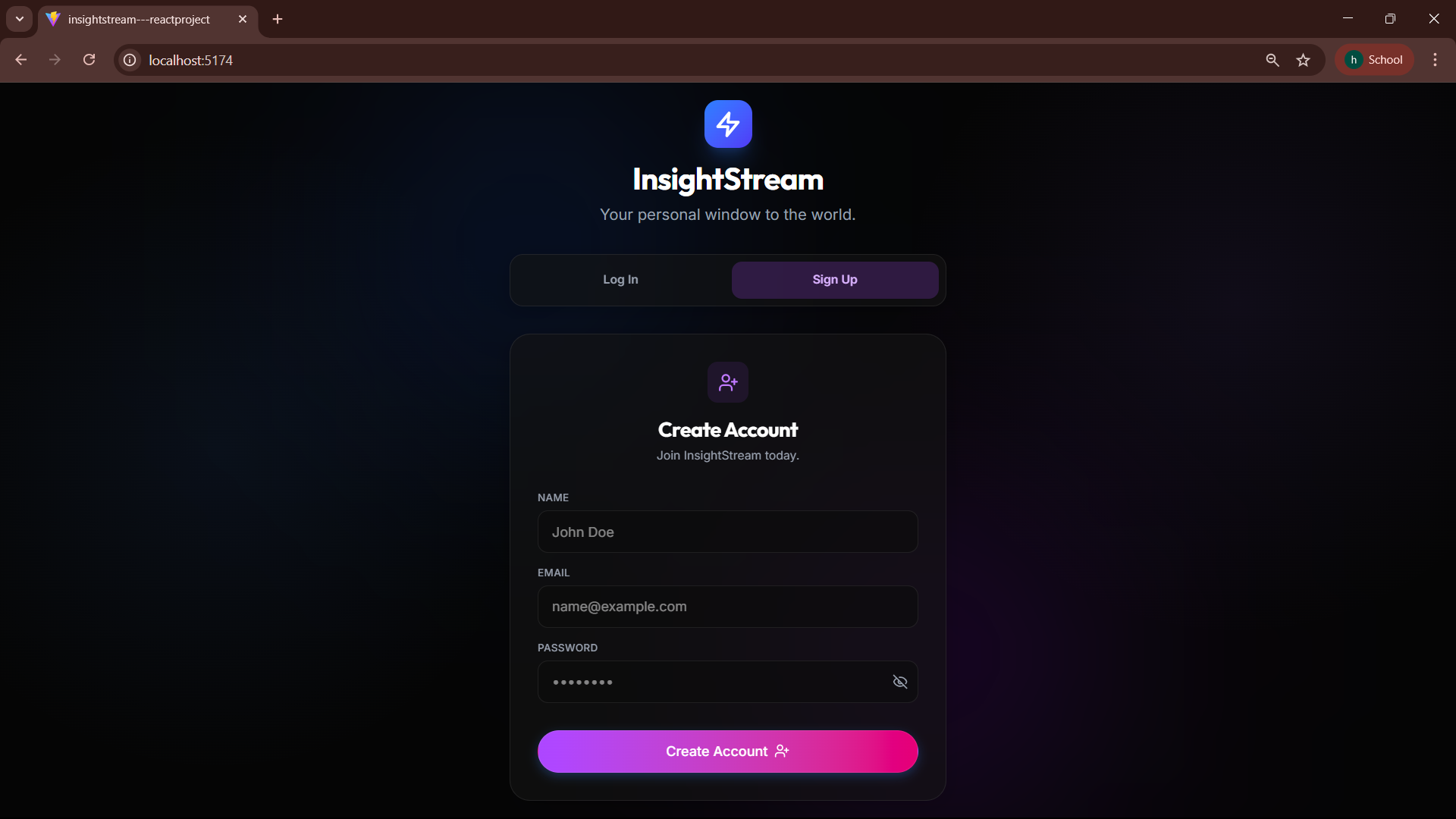
Milestone 2: Web Development:

****Focus:**** Building the core web interface, user interactions, and data flow.

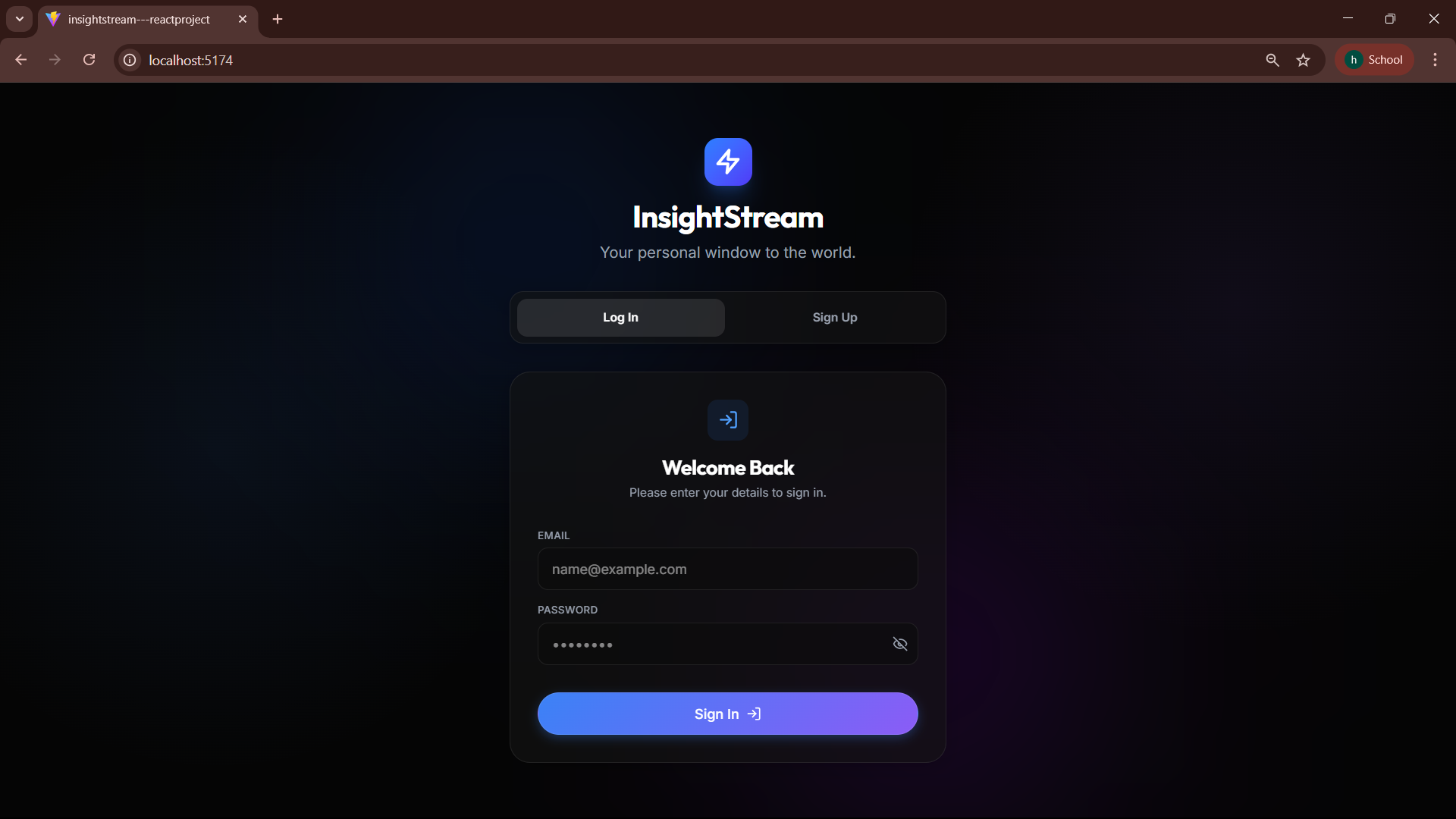
* **Dashboard & Navigation:** Home.jsx, News.jsx, Bookmarks.jsx, History.jsx, and Profile.jsx handle main pages, news display, bookmarks, history, and profile statistics.
* **User Interactions & State Management:** Functions in the pages handle user actions; state managed with useState; API calls via **Axios**.
* **Navigation Components:** Navbar.jsx provides navigation and logout functionality.
* **Error Handling & Lifecycle Management:** All pages include error handling, lifecycle management with useEffect, and logout handling in the Navbar.

Project Output:

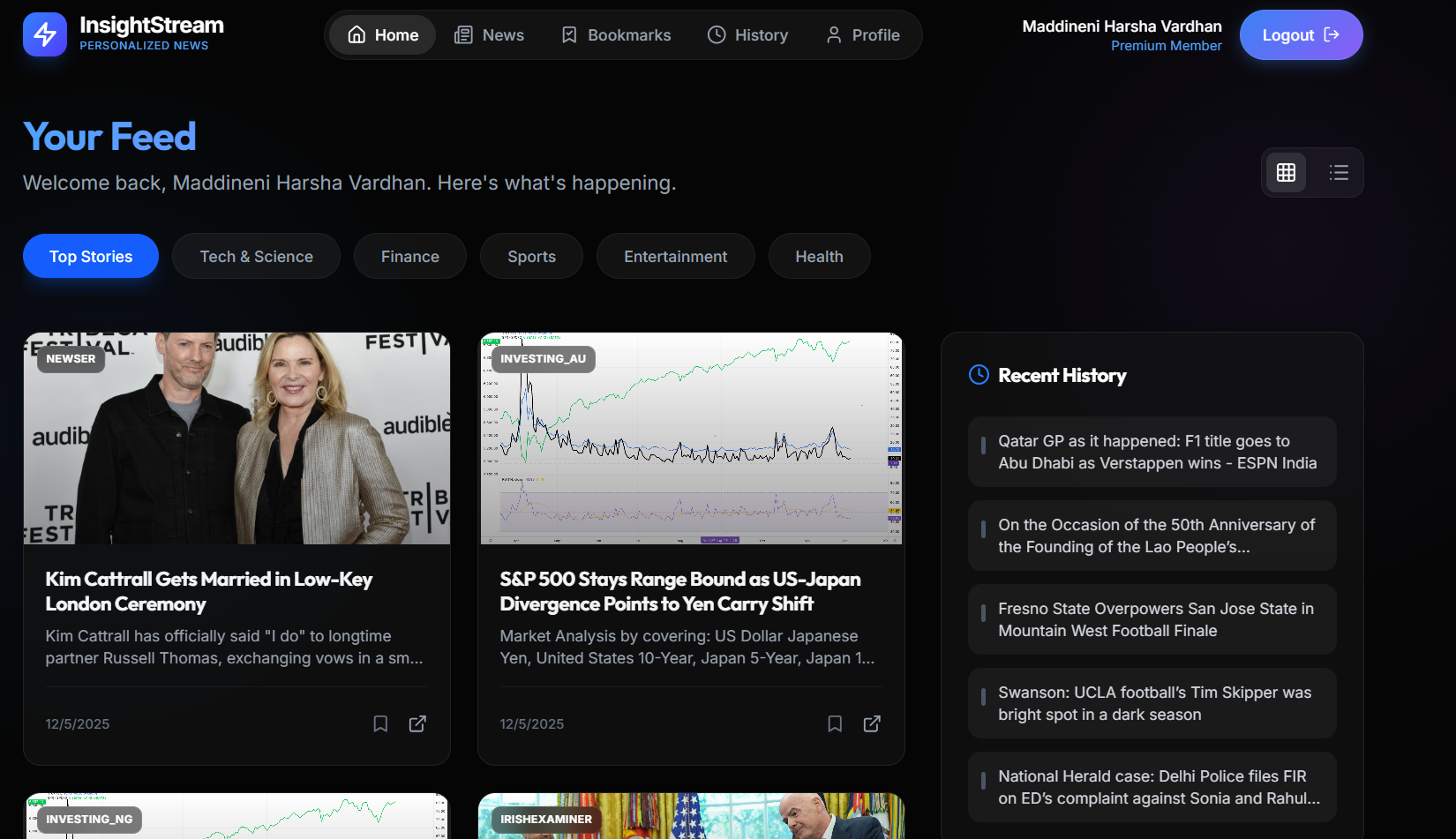
Register page:



Login page:

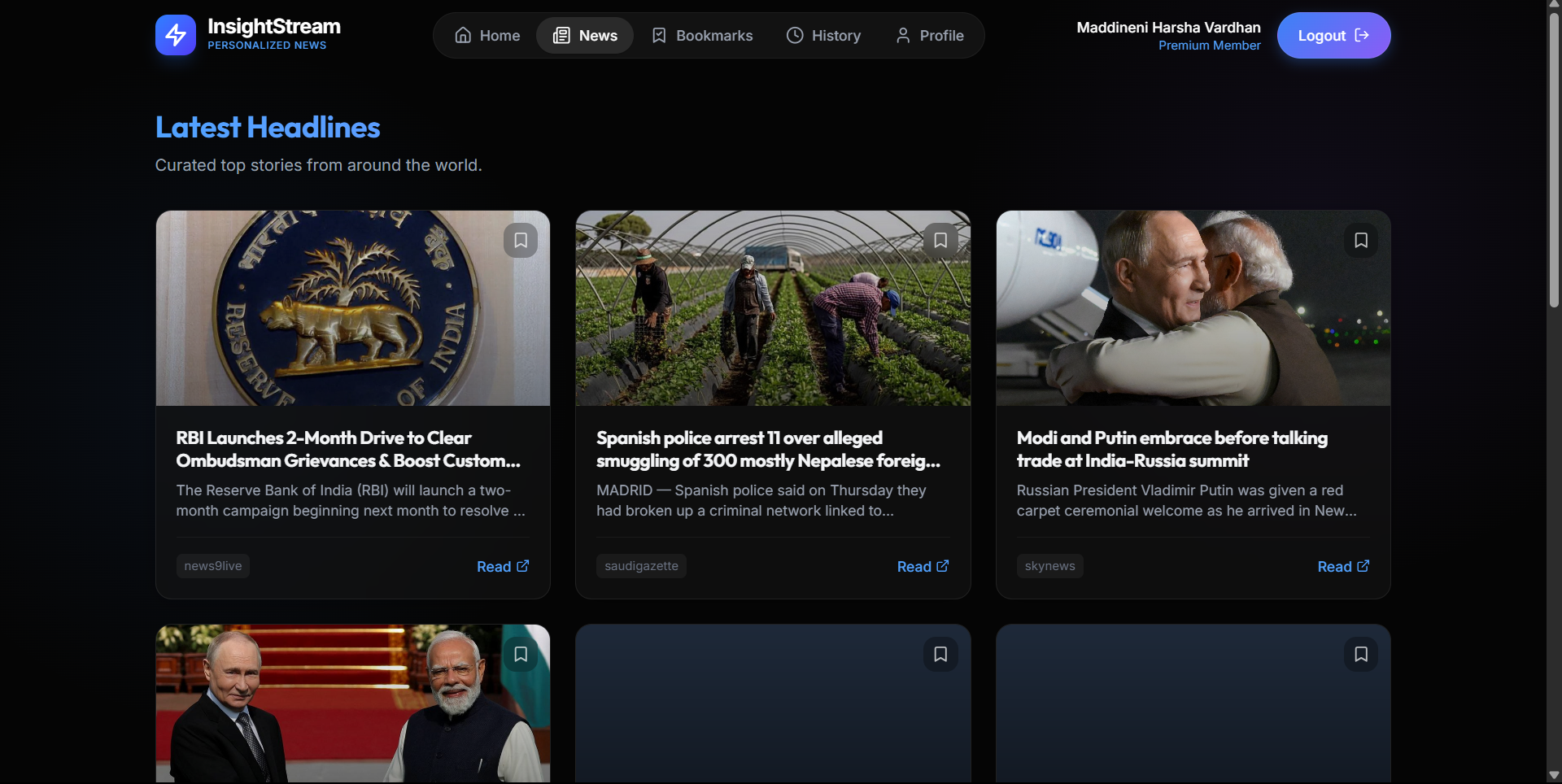


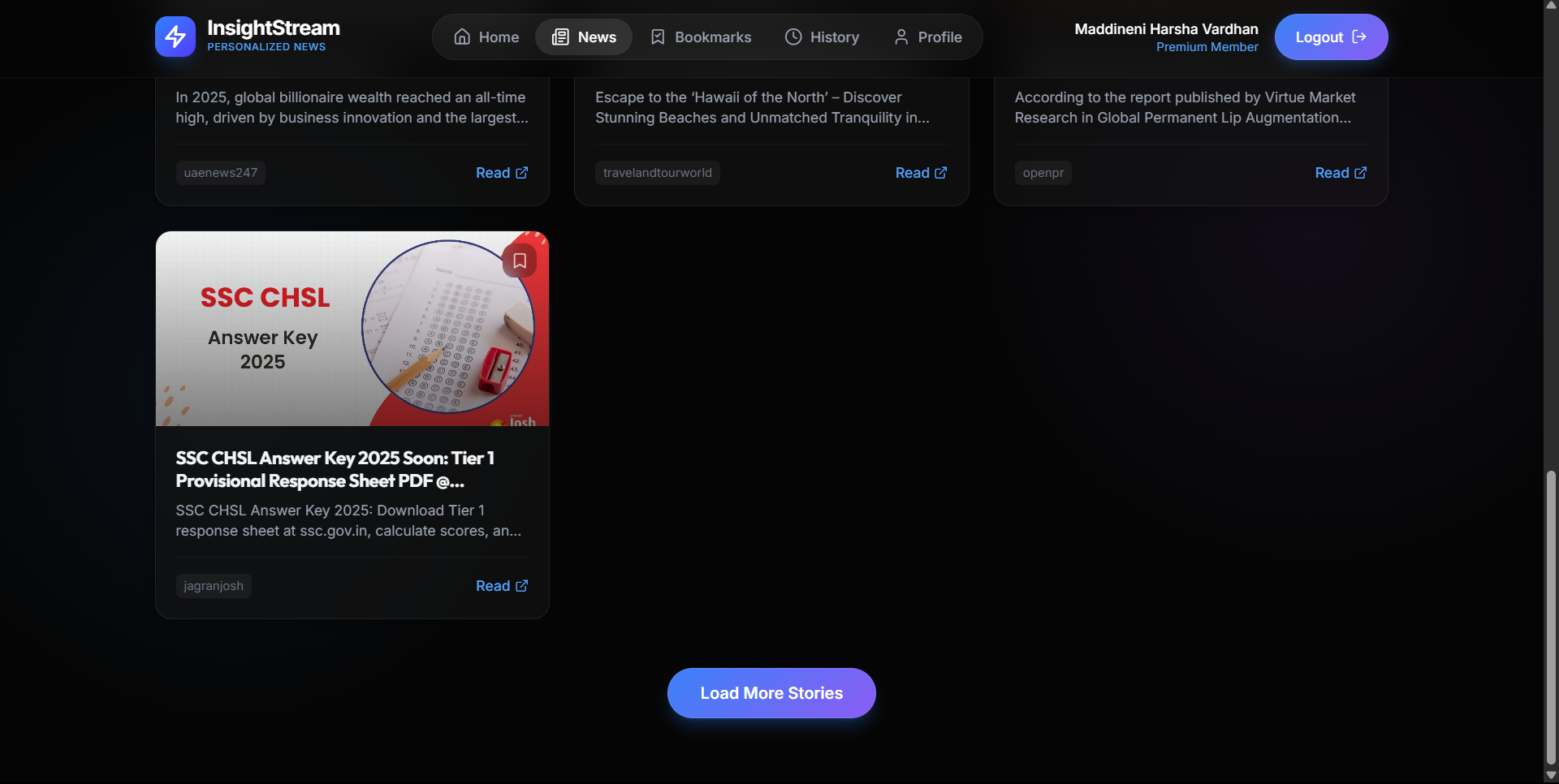
Home Page:



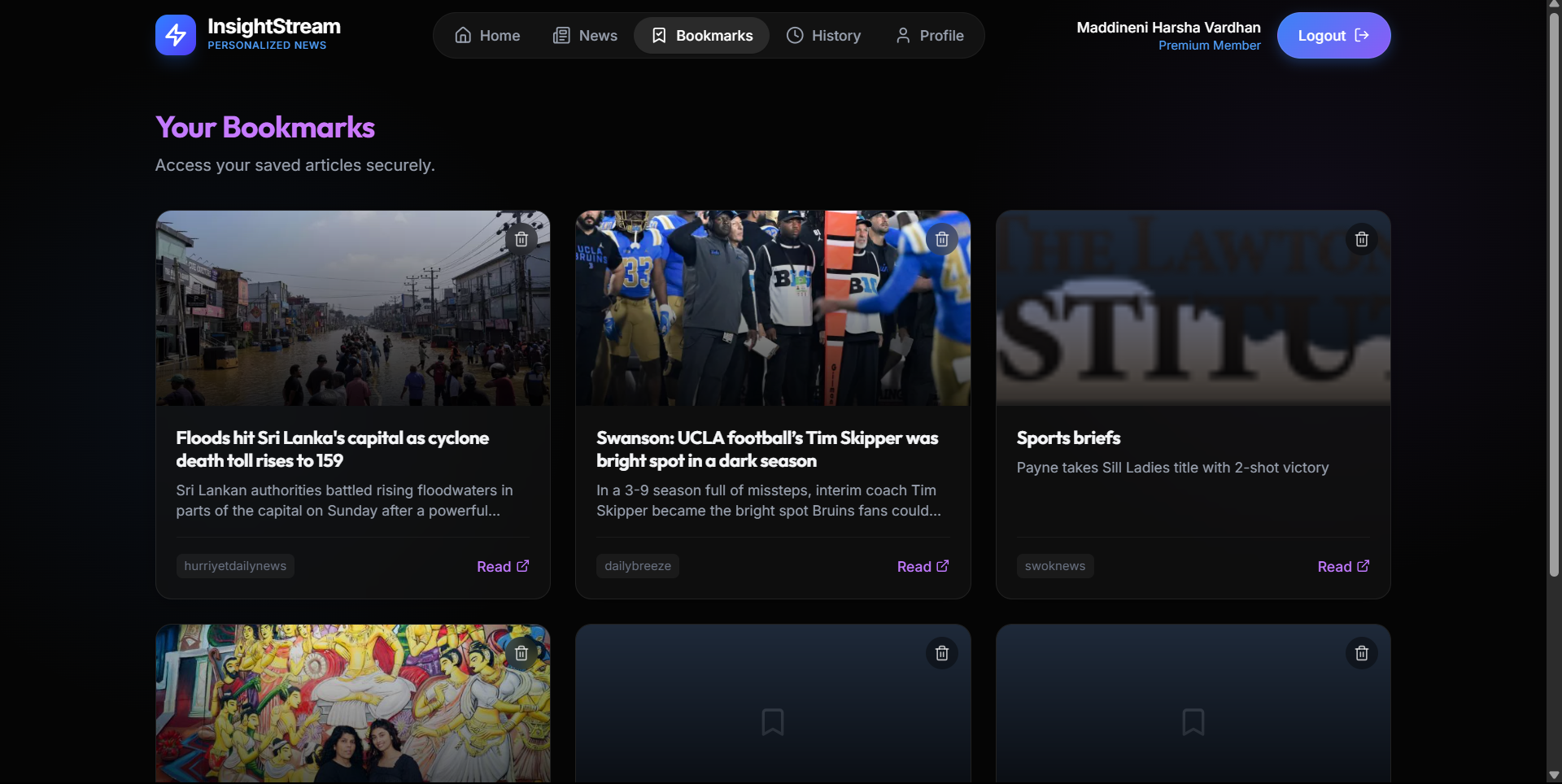


News Page:

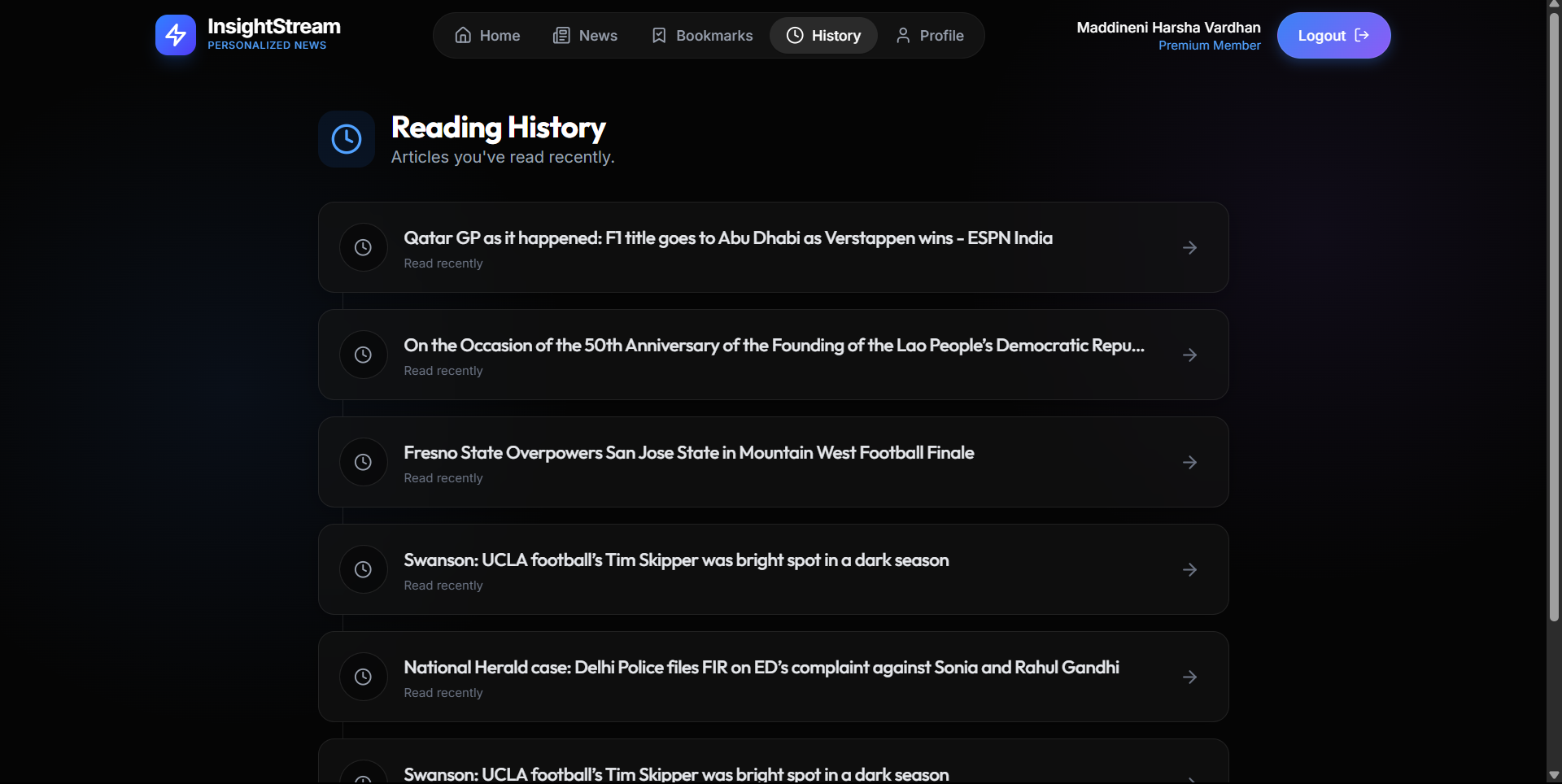




Bookmarks Page:



History Page:



Profile Page:



PROJECT DEMO LINK:

<https://drive.google.com/drive/folders/101QbJBy2StyoT5P1WCLCZWIsahqyGaOR?usp=sharing>

THANK YOU!