**Project Documentation**

**Insightstream:Navigate the News Landscape**

1.Introduction:

* Project Title: Insightsream:Navigate the newslandscape.
* Team ID:
* Team Leader: Dakshata.S [dakshata10197@gmail.com]
* Team Members:

Deepika.M [deepikaam744@gmail.com]

Deepika.K [deepikakuppusamy263@gmail.com]

Dharshini.S [dharshinisurendhiran7@gmail.com]

2. Project Overview:

* Purpose: InsightStream’s main goal is to create an easy-to-use platform for people who love staying updated, exploring different news topics, and getting the latest information.
* Features:

-**News from API Sources.**

**-Visual News Exploration.**

**- Intuitive Design.**

**- Advanced Search Feature.**

3. Architecture:

* Component Structure: Reusable UI parts (Navbar, Footer, NewsCard, NewsList, SearchBar, CategoryFilter, Loader).

**context/** – Global state (NewsContext).

**pages/** – Full pages for routing (HomePage, ArticlePage, CategoriesPage, AboutPage)

**styles/** – CSS files (variables.css, components.css)

**App.js** – Main layout and routes.**ndex.js** – Entry point for rendering the app.

* State management: We use React’s useState for local state and useContext for shared state, so we don’t need extra tools like Redux.
* Routing : React Router handles page navigation smoothly, letting users switch pages without reloading.

4. Setup Instructions:

Prerequisites:

* Install Node.js (version 14 or newer)
* Use npm (included with Node.js) to manage packages.

Installation Steps:

1. Clone the repository:

https://github.com/dharshinisuren/Insightstream-Navigate-the-newslandscape

1. Open the project folder:

cd InsightStream

1. Install dependencies:

npm install

1. (Optional) Add environment variables in a .env file.
2. **Folder Structure**

Client (React App):

* src/components/ — Reusable UI components
* src/pages/ — Main page components connected to routes
* src/assets/ — Images, icons, and other static files
* src/utils/ — Utility functions and custom hooks for tasks like data formatting and API calls

**6. Running the Application**

* Start the app locally:

npm start

* Opens in your browser at [http://localhost:3000](http://localhost:3000/) and reloads automatically when you make changes.

**7. Component Documentation**  
Key Components:

* Dashboard — Displays live data insights and visualizations
* ReportList — Shows a list of reports with filtering options
* NavigationBar — Provides app-wide navigation

Reusable Components:

* Chart — Versatile chart for multiple data types
* Button — Customizable styled button
* Modal — Popup dialog for user actions

**8. State Management**

* **Global State:** Managed with React Context API for shared data like authentication and theme.
* **Local State:** Handled within components using useState for things like forms and toggles.

**9. User Interface**

* Clean, modern layout for easy navigation.
* Interactive charts and tables for exploring data.

**10. Styling**

* Uses CSS modules for scoped styles.
* May include Styled-Components for maintainable, component-level styling.

**Theming**

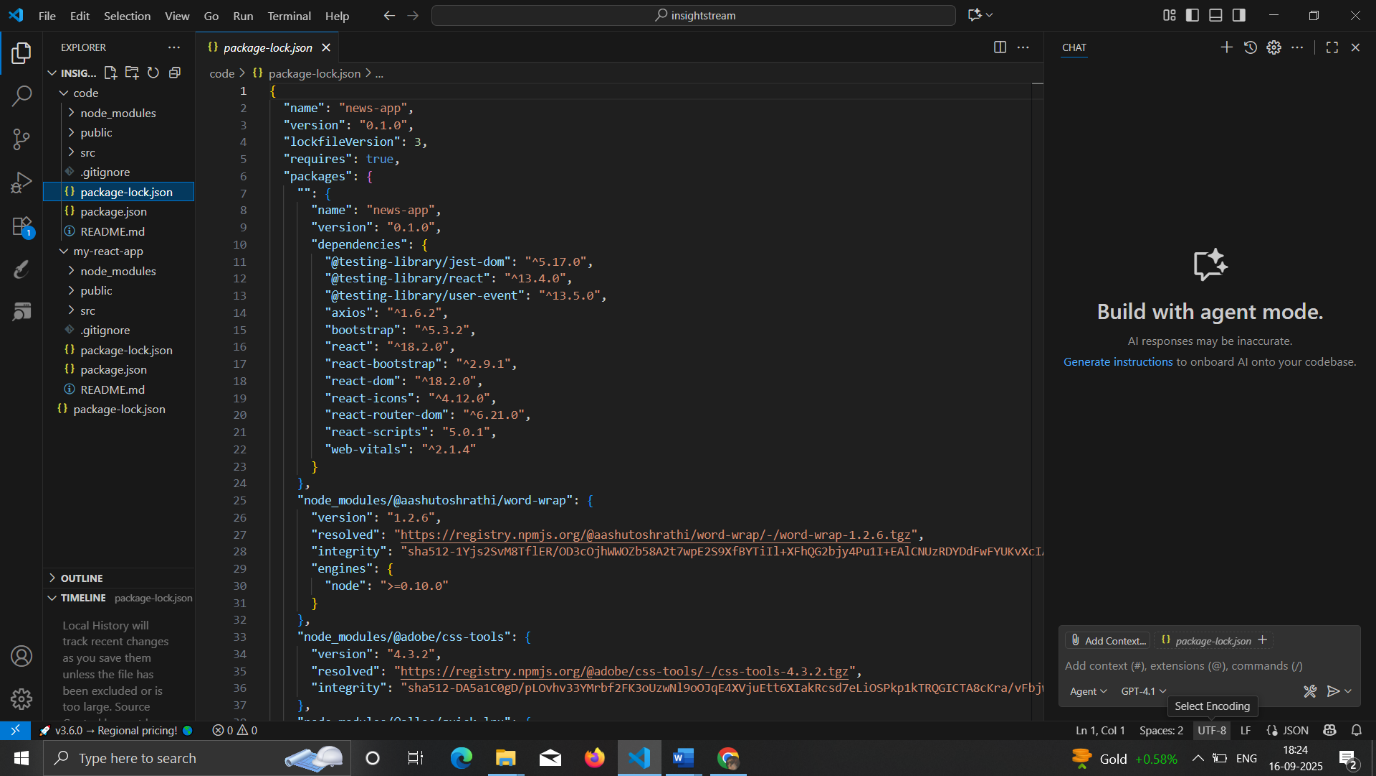
* Supports light and dark modes, letting users switch themes easily.

**11. Testing**

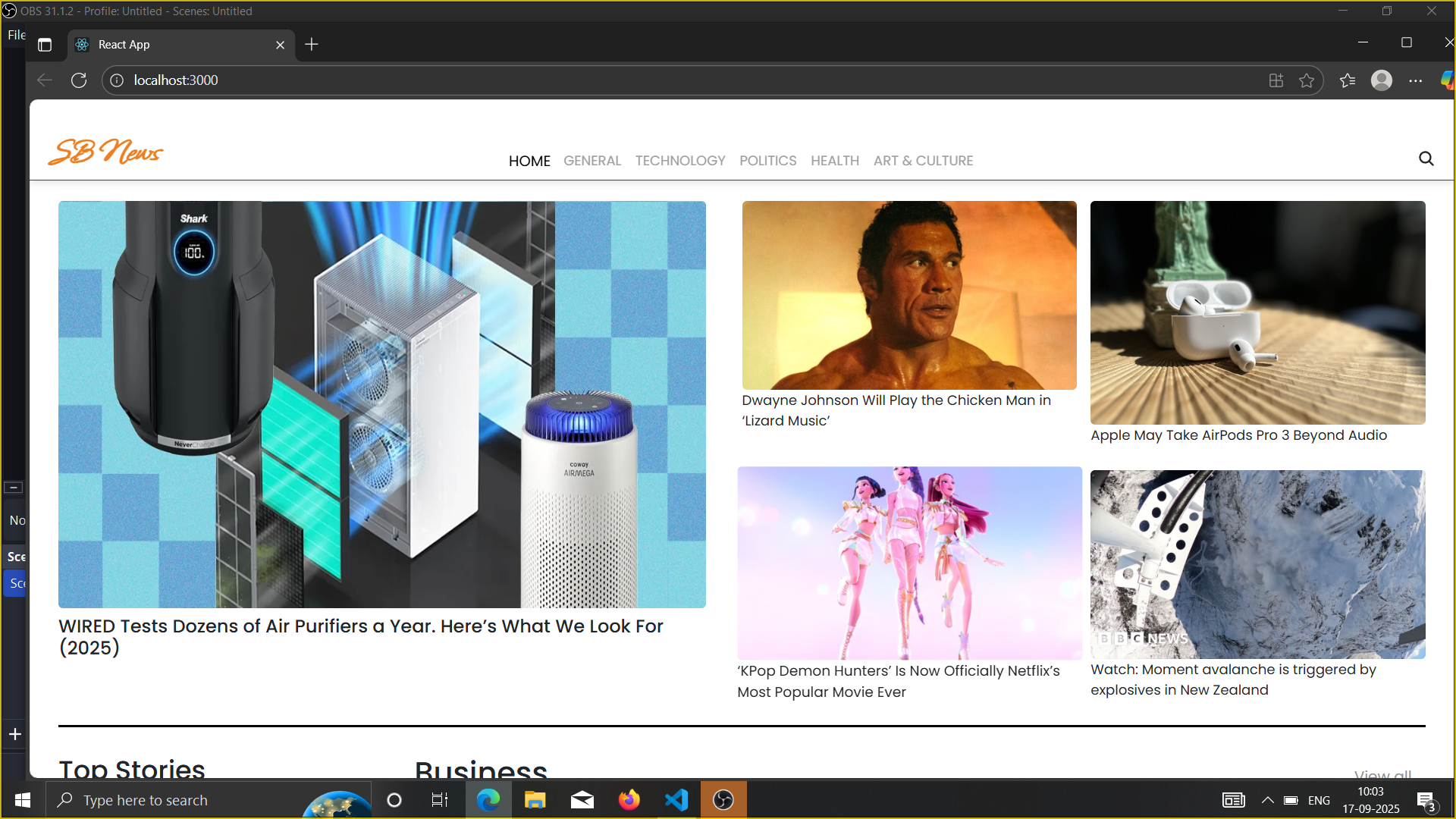
* **Testing Strategy:** Uses Jest and React Testing Library for unit and integration tests to ensure components work correctly.
* **Code Coverage:** Coverage reports show tested areas and highlight gaps needing more tests.

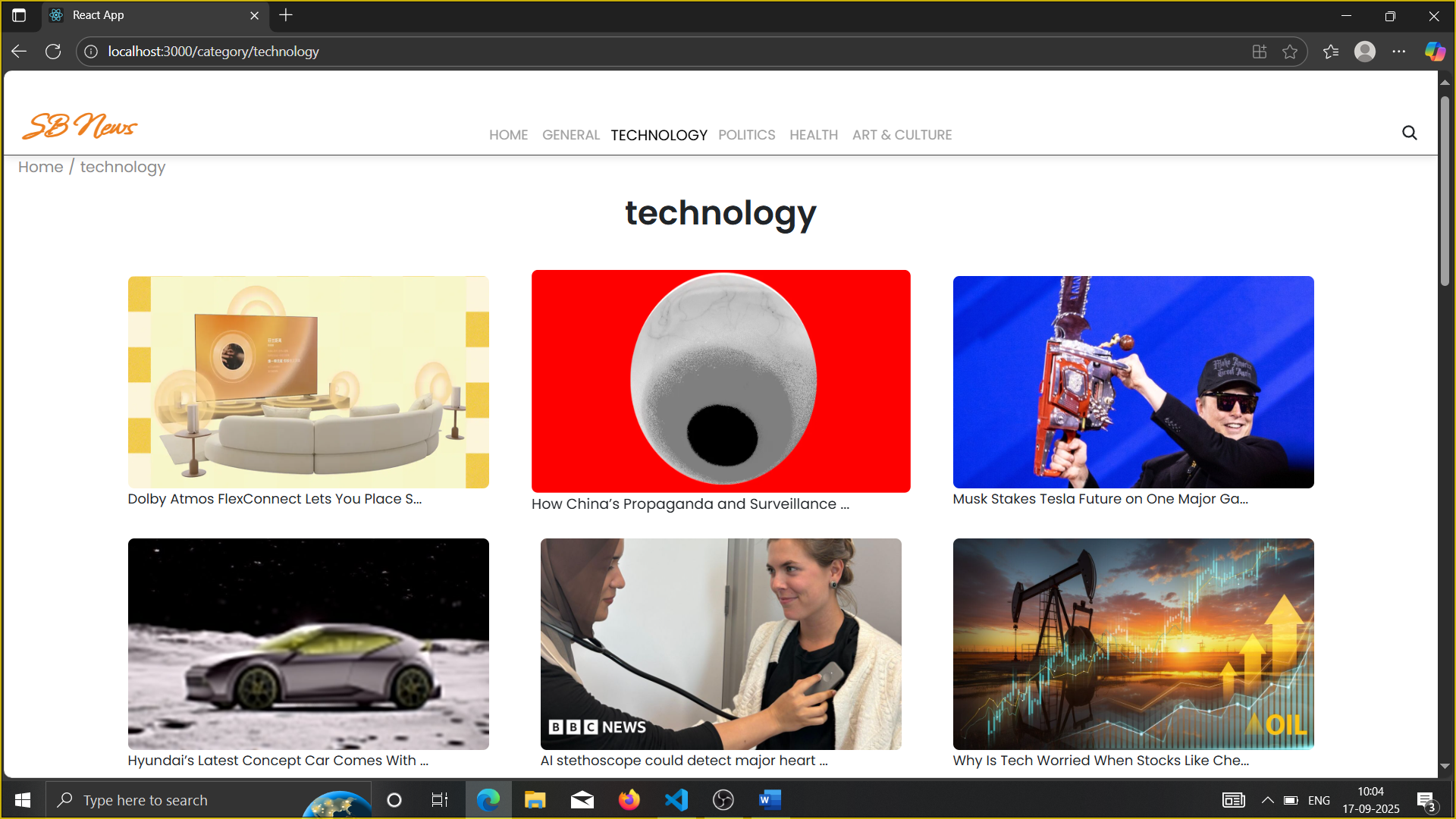
12. SCREENSHOTS OR DEMO

Coding:



Output:





**13. Known Issues**

* Some lint warnings may appear during development but don’t impact functionality.
* The build process might need adjustments if new dependencies are added.

**14. Future Enhancements**

* Add Redux for more scalable state management.
* Introduce animations and smooth transitions for a better user experience.
* Expand testing to include end-to-end scenarios.
* Add more detailed analytics and reporting features.
* Improve accessibility and mobile responsiveness.