**PROJECT DOCUMENTATION**

**InsightStream: Navigate the News Landscape**

**.**

**1. Introduction**

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

**• Team ID:** NM20255TMID31185

**• Team Leader:VIJAYABHARATHI K – vijaybharathivijay38@gmail.com**

**• Team Members:**

**o VASIKARAN N–vjvasi1234@gmail.com**

**o VIGNESH D–vigneshvignesh03375@gmail.com**

**o VINNARASAN S–vinnarasan1420@gmail.com**

**o YOGESH M – yogrsh120296@gmail.com**

**2. Project Overview**

* **Purpose:**

**The purpose of this project is to provide users with a streamlined way to explore and navigate the vast news landscape. It brings together top stories and categorized news such as business, politics, technology, health, and culture. Users can easily access relevant articles through clear navigation and search options. The platform also encourages engagement with features like newsletters and trending highlights. Overall, it aims to deliver a modern, accessible, and insightful news-reading experience.**

* **Goals:** 
  + 1. **To provide users with the latest news articles from multiple sources.**
    2. **To enable filtering by category, source, and publication date.**
    3. **To implement search functionality for quick access to specific news.**
    4. **To deliver a clean, distraction-free reading experience.**
    5. **To allow users to explore trending topics and insights visually.**
* **Key Features:**

1. **Dynamic News Categories** – Browse news by sections like General, Technology, Politics, Health, and Art & Culture.
2. **Top Stories Highlight** – Displays the most trending and important articles in a featured section.
3. **Category Pages** – Each category has a dedicated page showing relevant articles.
4. **Full Article View** – Users can click on a headline to read the complete news story.
5. **Newsletter Subscription** – Visitors can subscribe with their email to receive regular updates.
6. **Responsive Design** – Layout adapts smoothly across desktop, tablet, and mobile devices.
7. **Reusable Components** – Modular components (Navbar, Footer, Hero, TopStories, etc.) for clean and maintainable code.
8. **State Management with Context** – Centralized handling of news data using GeneralContext.jsx.

**3. Architecture**

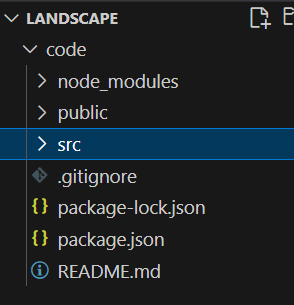
* **Component Structure** 
  + **index.js** → Entry point, renders <App /> into the DOM.
  + **App.js** → Root component, sets up routes and wraps Navbar + Footer.
  + **NavbarComponent.jsx** → Provides navigation (Home, General, Technology, Politics, Health, Art & Culture).
  + **Hero.jsx** → Displays featured or top news on the homepage.
  + **TopStories.jsx** → Highlights the trending or most important articles.
  + **HomeArticles.jsx** → Lists categorized news sections like Business, Technology, Politics.
  + **NewsLetter.jsx** → Section for users to subscribe with their email.
  + **Footer.jsx** → Branding, copyright, and important links.
  + Pages
    - **Home.jsx** → Homepage container, uses Hero, TopStories, HomeArticles, and NewsLetter.
    - **CategoryPage.jsx** → Displays news filtered by category (General, Technology, Politics, Health, Art & Culture).
    - **NewsPage.jsx** → Displays full article details when a user selects a news item.
  + Context
    - **GeneralContext.jsx** → Provides shared state management for categories, articles, and API data.
  + Styles
    - Each component imports its own dedicated CSS module (e.g., Hero.css, TopStories.css, NewsPage.css) for styling consistency.
* **State Management** 
  + **Local State: Managed using React useState and useEffect.**
  + **API Integration: Axios is used to fetch real-time news data from external news APIs..**
* **Routing** 
  + **Library: react-router-dom**
  + **Routes:** 
    - **/ → Home.jsx – Displays Top Stories, Business, Technology, Politics, and newsletter section.**
    - **/general → CategoryPage.jsx – Shows articles under the General category.**
    - **/technology → CategoryPage.jsx – Shows articles under Technology.**
    - **/politics → CategoryPage.jsx – Shows articles under Politics.**
    - **/health → CategoryPage.jsx – Shows articles under Health.**
    - **/art-culture → CategoryPage.jsx – Shows articles under Art & Culture.**
    - **/news/:id → NewsPage.jsx – Displays full details of a single news article**

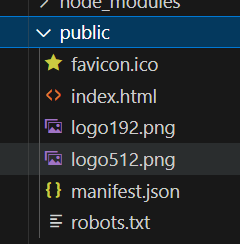
**4. Setup Instructions**

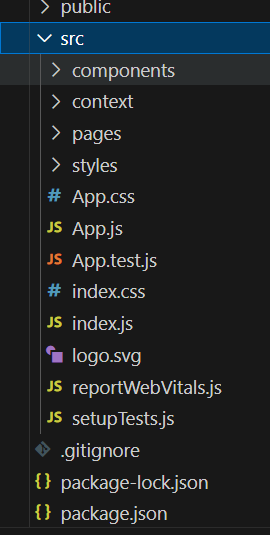
**Prerequisites**

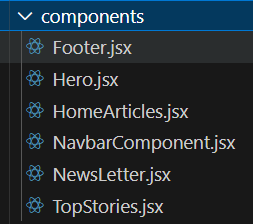
* **Node.js & npm** 
  + **Node.js is required to run React applications.**
  + **npm (Node Package Manager) is used to install dependencies.**
  + **Download Node.js**
* **React.js** 
  + **React is the main JavaScript library used to build this project.**
  + **If you don’t have an existing React app, create one using:**
  + **npx create-react-app my-app**
  + **cd my-app**
  + **npm start**
  + **In SB Fitzz, the React app is already created, so you just need to install dependencies (npm install).**
* **Git** 
  + **Used for cloning and version control.**
  + **Download Git**
* **Code Editor** 
  + **Recommended: Visual Studio Code (VS Code)**
  + **Download VS Code**
* **Basic Knowledge** 
  + **HTML, CSS, JavaScript**
  + **React concepts (components, props, hooks, state, routing)**
* **Installation** 
  + **Get the code:** 
    - **Download the code from the drive link given below:** 
      * + <https://drive.google.com/drive/folders/1E9QxUsJfs_W1pTK3qt8lB_yWaYAecCWd?usp=drive_link>
* **Install Dependencies:** 
  + **Navigate into the cloned repository directory and install libraries:** 
    - **Navigate into the cloned repository directory and install libraries:** 
      * **cd CODE**
      * **npm install**
  + **Start the Development Server:** 
    - **To start the development server, execute the following command:** 
      * **npm start**
* **Access the App:** 
  + **Open your web browser and navigate to http://localhost:3000.**
  + **You should see the application's homepage, indicating that the installation and setup were successful.**
* **Environment Variables** 
  + **Create a .env file with:** 
    - **REACT\_APP\_API\_URL=<https://exercisedb.p.rapidapi.com/exercises/equipmentList>**
    - **REACT\_APP\_YOUTUBE\_API\_KEY=<33cf3a7616msh4c3b1e3204f24e2p1294b3jsne16a7323d732>**

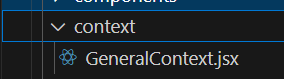
**5. Folder Structure**

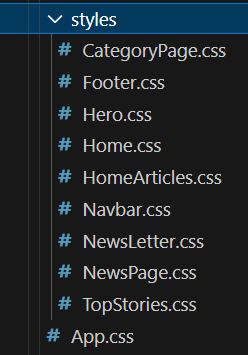












**6.Running the Application**

* **Start development server:** 
  + **npm start**
* **Build for production:** 
  + **npm run build**
* **Run tests:** 
  + **npm test**

**7. Component Documentation**

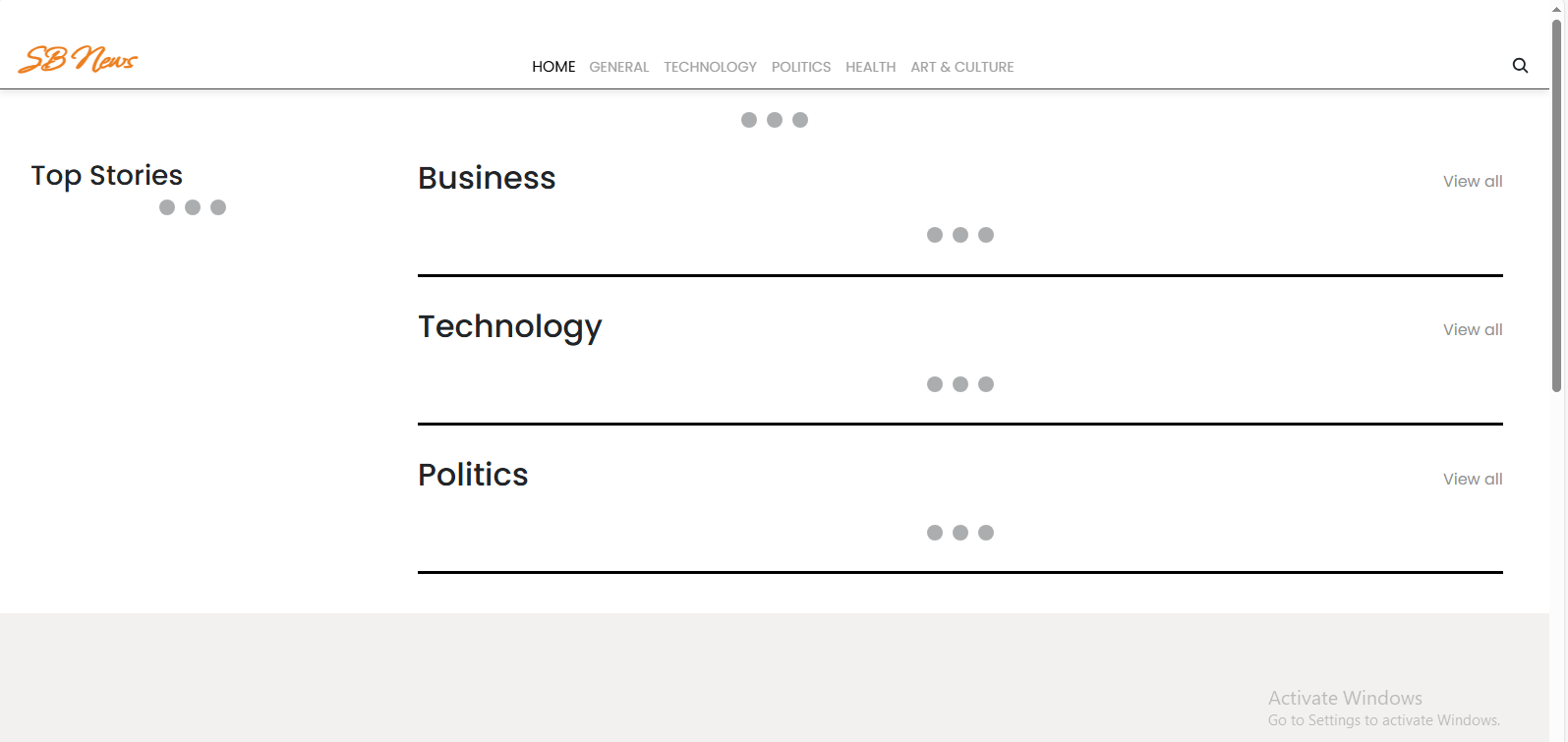
* **index.js** → Entry point that renders <App /> into the DOM.
* **App.js** → Wraps NavbarComponent, defines all routes, and includes Footer.
* **NavbarComponent.jsx** → Provides navigation links (Home, General, Technology, Politics, Health, Art & Culture).
* **Home.jsx** → Homepage that uses Hero, TopStories, HomeArticles, and NewsLetter.
* **CategoryPage.jsx** → Displays news articles filtered by category (General, Technology, Politics, Health, Art & Culture).
* **NewsPage.jsx** → Shows the detailed view of a single article when selected.
* **Footer.jsx** → Displays branding, copyright, and important links.

**8. State Management**

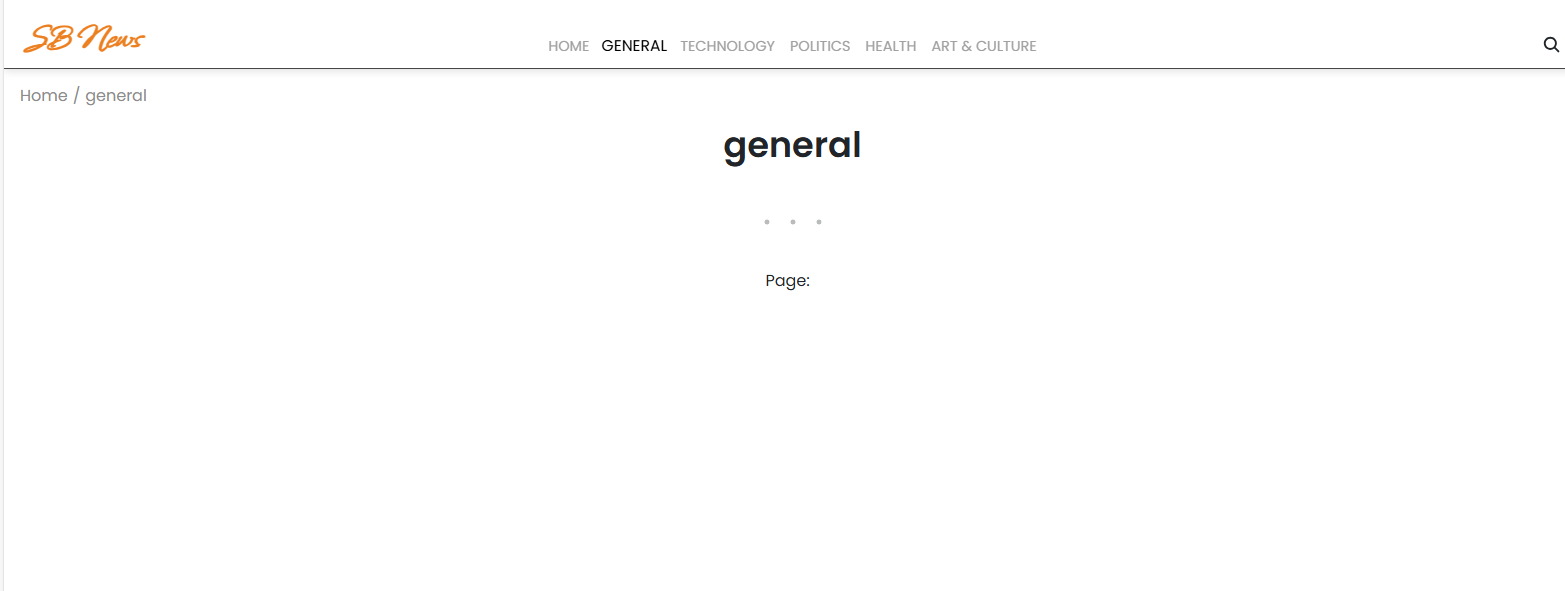
* **Local State:** 
  + **Search queries stored in HomeSearch.**
  + **API data fetched and stored per-page.**
* **Global State:** 
  + **Not implemented — app uses component-level state.**

**9. User Interface**

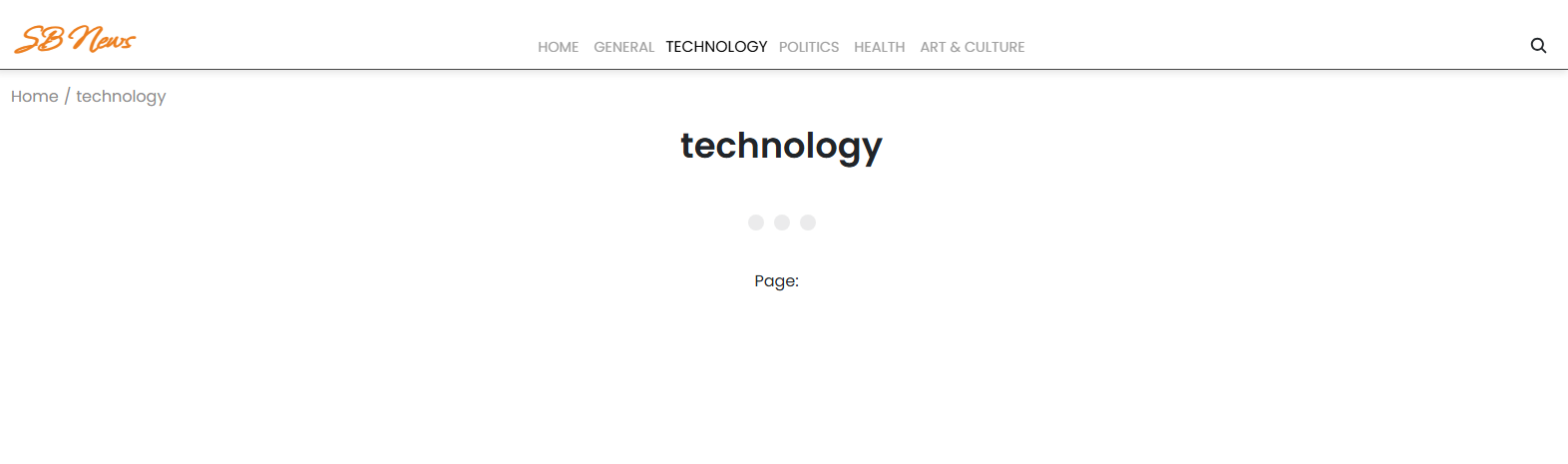
* **Pages include:**



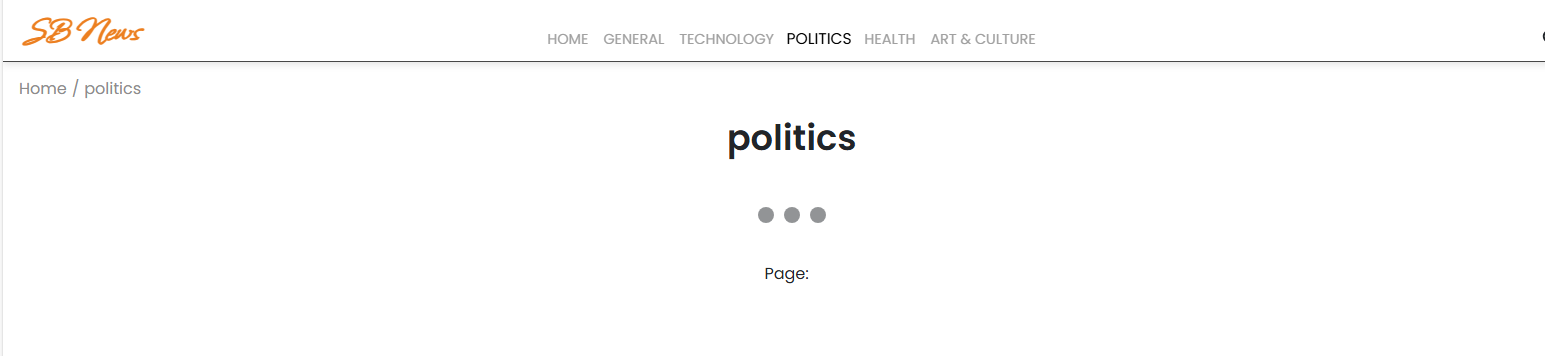
* General



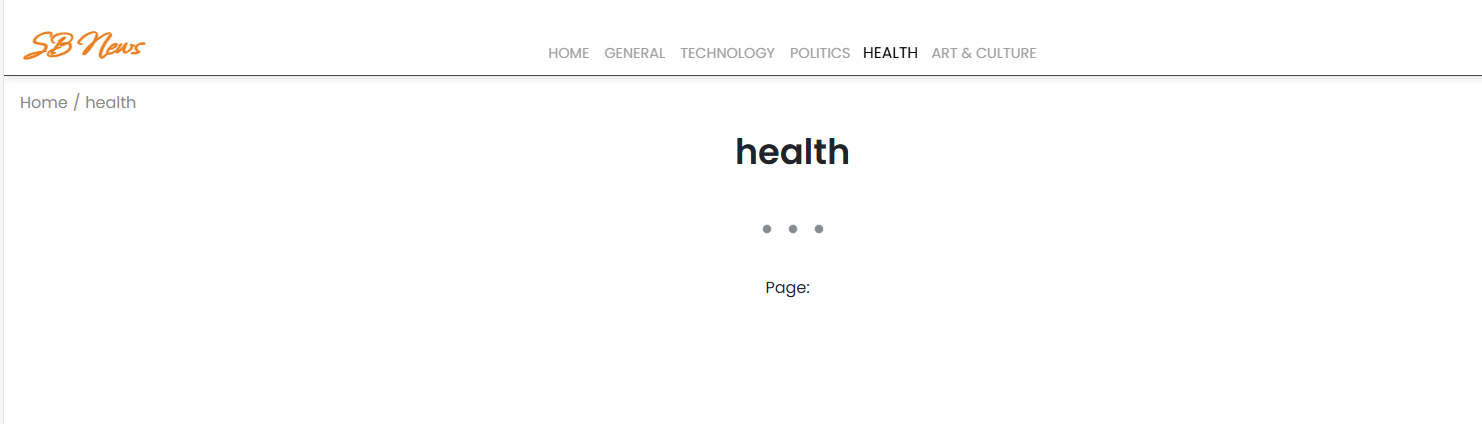
* Technology



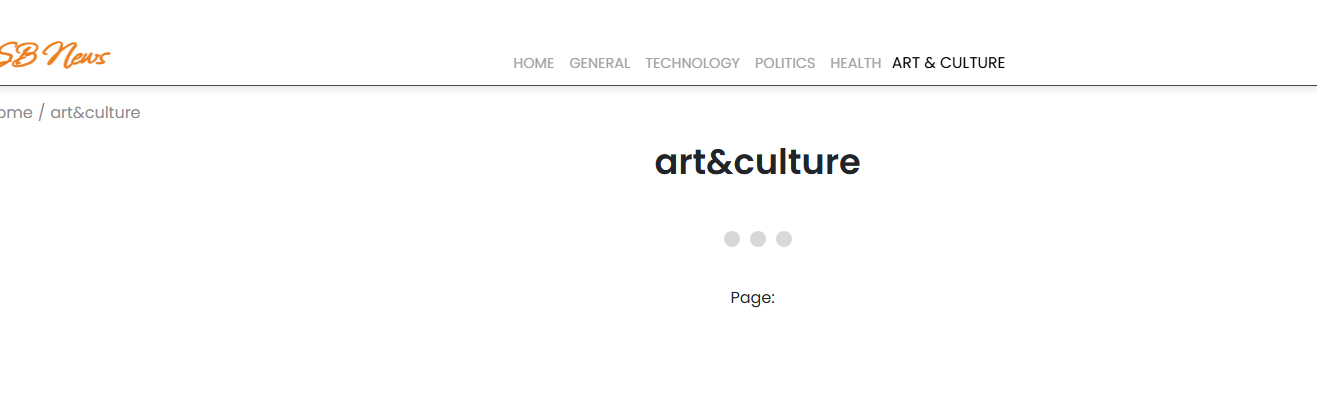
* Politics



* Health



* Art & Culture



**10. Styling**

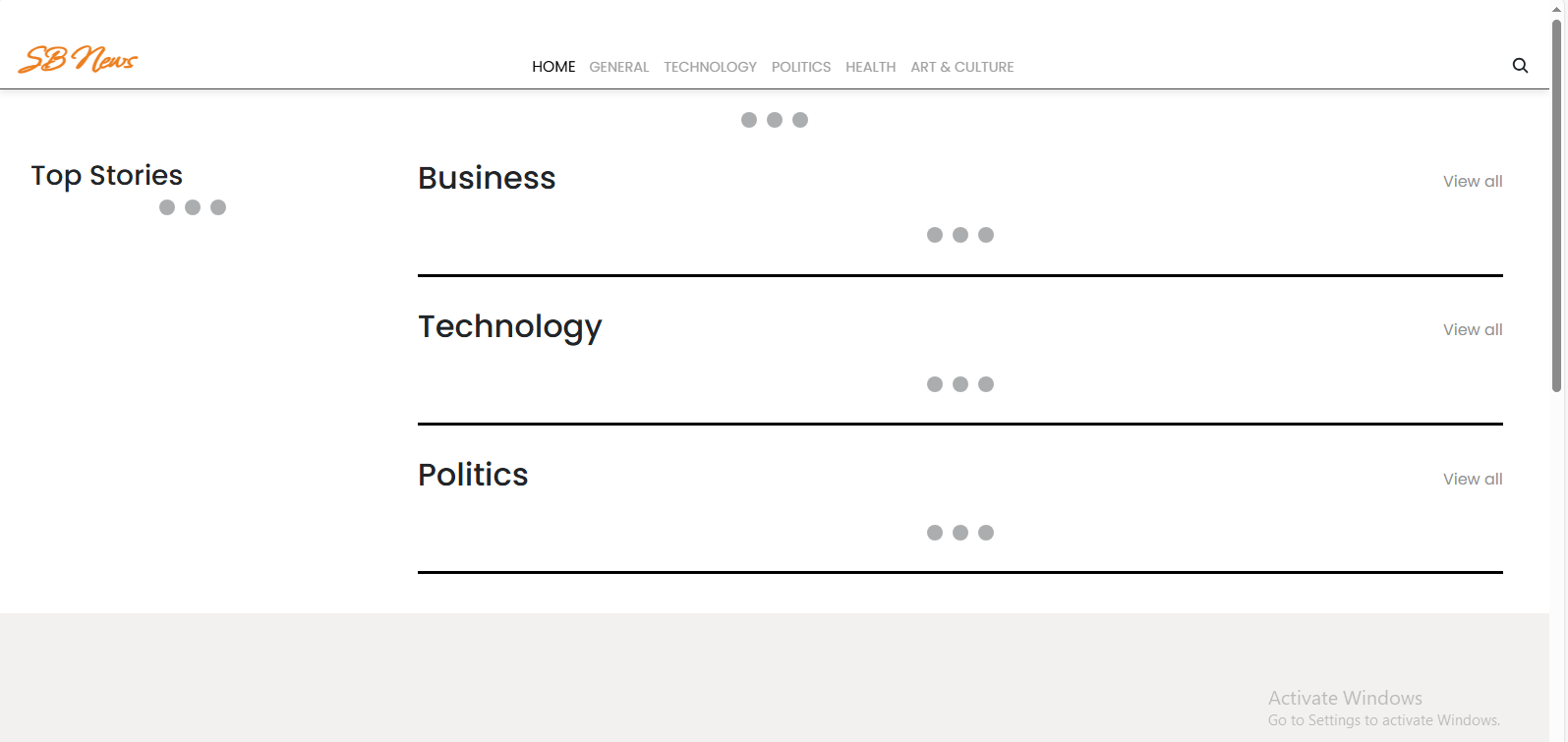
* **Frameworks Used: Tailwind CSS / Bootstrap.**
* **Custom CSS: Stored in src/styles/.**
* **Each page/component has a dedicated CSS file for modularity.**

**11. Testing**

* **Libraries Used: Jest, React Testing Library.**
* **Unit Tests: Written in App.test.js.**
* **Setup: Configured with setupTests.js.**

**12. Screenshots / Demo:**

* **Screenshot:**



**13. Known Issues**

* + **API rate-limit may cause some exercises not to load.**
  + **YouTube API sometimes fails to fetch related videos.**

**14. Future Enhancements**

* **User Authentication** – Allow users to create accounts, log in, and personalize their news feed.
* **Advanced Search & Filters** – Search articles by keywords, authors, publication date, or trending topics.
* **Bookmarks & Reading List** – Users can save articles for later reading and organize them into collections.
* **Dark Mode Support** – Provide a toggle for light and dark themes to improve readability.
* **Multi-Language Support** – Offer news content in multiple languages for wider accessibility.
* **Push Notifications** – Notify users about breaking news and trending stories in real-time.
* **Analytics Dashboard** – Visualize news trends with graphs, word clouds, and engagement insights.