

FRONTEND DEVELOPMENT WITH REACT.JS

PROJECT DOCUMENTATION

PROJECT TITLE: INSIGHTSTREAM NAVIGATE THE NEWS APP

TEAM ID: SWTID1741246731150404

TEAM MEMBERS:

TAMIZHSELVAN V	blackthamizhan7@gmail.com
SARVAN A	sarvansunil004@gmail.com
THAMIZHARASAN M	thamizharasan2605@gmail.com
THULASIRAMAN R	thulasiraman13a@gmail.com
VETRIVEL S	ksvetri637@gmail.com

INTRODUCTION:

A **News App** is a digital platform that provides users with real-time news, articles, and multimedia content on various topics such as politics, sports, business, technology, entertainment, and more. These apps aggregate content from multiple sources and present it in an organized, user-friendly format.

Key Features of a News App:

1. **Personalized News Feed** – Users can customize their feed based on interests and preferences.
2. **Breaking News Alerts** – Instant notifications for important and trending news.
3. **Multimedia Support** – Articles with images, videos, and podcasts for better engagement.
4. **Offline Reading** – Allows users to save articles for later reading without the internet.
5. **Dark Mode & UI Customization** – Provides a comfortable reading experience.
6. **Multi-Language Support** – News content available in different languages.
7. **Comment & Share Features** – Users can discuss and share news on social media.

Popular News Apps

- Google News
- Apple News
- BBC News
- CNN

- Flipboard

2. Project Overview

Purpose:

The NEWS APP is a frontend application built with React.js that allows users to browse and read the latest news articles from various categories. The goal is to provide a seamless and responsive user experience for accessing news content.

Features:

- **Category-based News Filtering:** Users can filter news by categories like Technology, Sports, Business, etc.
 - **Search Functionality:** Users can search for specific news articles.
 - **Responsive Design:** The app is optimized for desktop, tablet, and mobile devices.
 - **Dark/Light Mode:** Users can toggle between dark and light themes.
-

3. Architecture

Component Structure:

- **App Component:** The root component that manages routing and global state.
- **Header Component:** Displays the app title, search bar, and theme toggle.
- **NewsList Component:** Renders a list of news articles.
- **NewsItem Component:** Displays individual news articles with title, description, and image.
- **Footer Component:** Contains app-related links and information.

State Management:

- **Context API:** Used for managing global state like theme (dark/light mode) and user preferences.

- **Local State:** Used within components for managing UI-specific states (e.g., form inputs, toggles).

Routing:

- **React Router:** Used for navigation between pages (e.g., Home, Category, Article Details).
 - Routes:
 - `/`: Home page with all news articles.
 - `/category/:categoryName`: Displays news articles for a specific category.
 - `/article/:articleId`: Shows detailed information about a specific article.
-

4. Setup Instructions

Prerequisites:

- Node.js (v16 or higher)
- npm (v8 or higher)

Installation:

1. Clone the repository: `git clone https://github.com/asunm12911998/news-app.git`
 2. Navigate to the project directory: `cd news-app/client`
 3. Install dependencies: `npm install`
 4. Configure environment variables:
Create a `.env` file in the `client` directory and add the following:
`REACT_APP_API_KEY: react-scripts start`
 5. Start the development server:
-

5. Folder Structure

Client:

- **src/components:** Contains reusable components like Header, Footer, NewsItem, etc.
- **src/pages:** Contains page-level components like Home, Category, ArticleDetails.
- **src/assets:** Stores images, icons, and other static files.
- **src/context:** Contains Context API providers and hooks.
- **src/utils:** Includes helper functions and custom hooks.

Utilities:

- **useFetch:** Custom hook for fetching data from the API.
- **formatDate:** Utility function to format article dates.

6. Running the Application

Frontend:

Run the following command in the client directory: `npm start`

The app will be available at `http://localhost:3000`

7. Component Documentation

Key Components:

- **Header:** Displays the app title, search bar, and theme toggle.
 - Props: `on Search`, `on Toggle Theme`
- **News List:** Renders a list of news articles.
 - Props: `articles` (array of news articles)
- **News Item:** Displays individual news articles.
 - Props: `title`, `description`, `image Url`, `published At`

Reusable Components:

- **Button:** A customizable button component.
 - Props: `on Click`, `label`, `variant` (primary/secondary)

8. State Management

Global State:

- **Theme Context:** Manages the app's theme (dark/light mode).
- **News Context:** Manages the fetched news data and category filters.

Local State:

- **Search Component:** Manages the search input state.
 - **Toggle Component:** Manages the state of UI toggles (e.g., theme toggle).
-

8. User Interface

1. Home Screen (News Feed)

✦ Features:

- Clean, scrollable news feed with headlines, images, and summaries.
- Category tabs (e.g., Top Stories, Politics, Sports, Tech, etc.).
- Personalized news recommendations.
- “Trending” or “Breaking News” section highlighted at the top.

✦ UI Elements:

- ✓ Card-based layout with article images & short descriptions.
 - ✓ Pull-to-refresh for the latest updates.
 - ✓ Floating action button (FAB) for quick access to bookmarks or settings.
-

2. Article Page

✦ Features:

- Full article with images, videos, or infographics.
- Font size adjustment and dark mode toggle.
- Share, bookmark, and like/dislike buttons.

◆ UI Elements:

- ✓ Large, readable fonts with good line spacing.
 - ✓ Embedded media support (videos, podcasts, etc.).
 - ✓ Comments section for user engagement.
-

3. Search & Filter Screen

◆ Features:

- Keyword-based search for finding articles.
- Filters for categories, sources, date range, and popularity.

◆ UI Elements:

- ✓ Search bar with auto-suggestions.
 - ✓ Drop-down filters for narrowing results.
 - ✓ “Most Read” or “Editor’s Picks” section.
-

4. Categories & Sections

◆ Features:

- News categorized into sections like World, Business, Science, Health, etc.
- Quick-access tabs for switching between categories.

◆ UI Elements:

- ✓ Horizontal scrolling tabs or a side menu.
 - ✓ Icons for easy recognition of categories.
-

5. Notifications & Alerts

◆ Features:

- Breaking news push notifications.
- Personalized alerts for selected topics.
- Notification history for missed updates.

◆ UI Elements:

- ✓ Bell icon for notification settings.
 - ✓ Categorized notifications (breaking, personalized, recommended).
-

6. User Profile & Settings

◆ Features:

- User preferences for topics & sources.
- Saved articles/bookmarks.
- Dark mode, font size, and language selection.

◆ UI Elements:

- ✓ Clean and minimal profile page.
 - ✓ Toggle buttons for settings (dark mode, language, etc.).
-

7. Offline Mode UI

◆ Features:

- Saved articles available for offline reading.
- Progress bar for syncing offline content.

◆ UI Elements:

- ✓ “Offline” label for downloaded articles.
 - ✓ Auto-sync indicator when connected to Wi-Fi.
-

8. Live News & Video Section

◆ Features:

- Live streaming of news channels.
- Short video news clips (similar to reels/stories).

◆ UI Elements:

- ✓ Floating video player.
- ✓ Scrollable video feed for quick news updates.

9. Interactive UI Elements

◆ Features:

- Polls & surveys on trending topics.
- Gesture-based navigation (swipe for next article).

◆ UI Elements:

- ✓ Animated transitions for smooth navigation.
- ✓ Polls displayed within articles.

10. Dark Mode & Accessibility Enhancements

◆ Features:

- Dark mode toggle for better nighttime reading.
- Text-to-speech support.
- High-contrast mode for visually impaired users.

◆ UI Elements:

- ✓ Dark theme option in settings.
- ✓ Accessibility-friendly font sizes & color contrast.

10. Styling

CSS Frameworks/Libraries:

- **Styled-Components:** Used for component-level styling.
- **Sass:** Used for global styles and theming.

Theming:

- The app supports dark and light themes, implemented using CSS variables and Context API.

11. Testing

Testing Strategy:

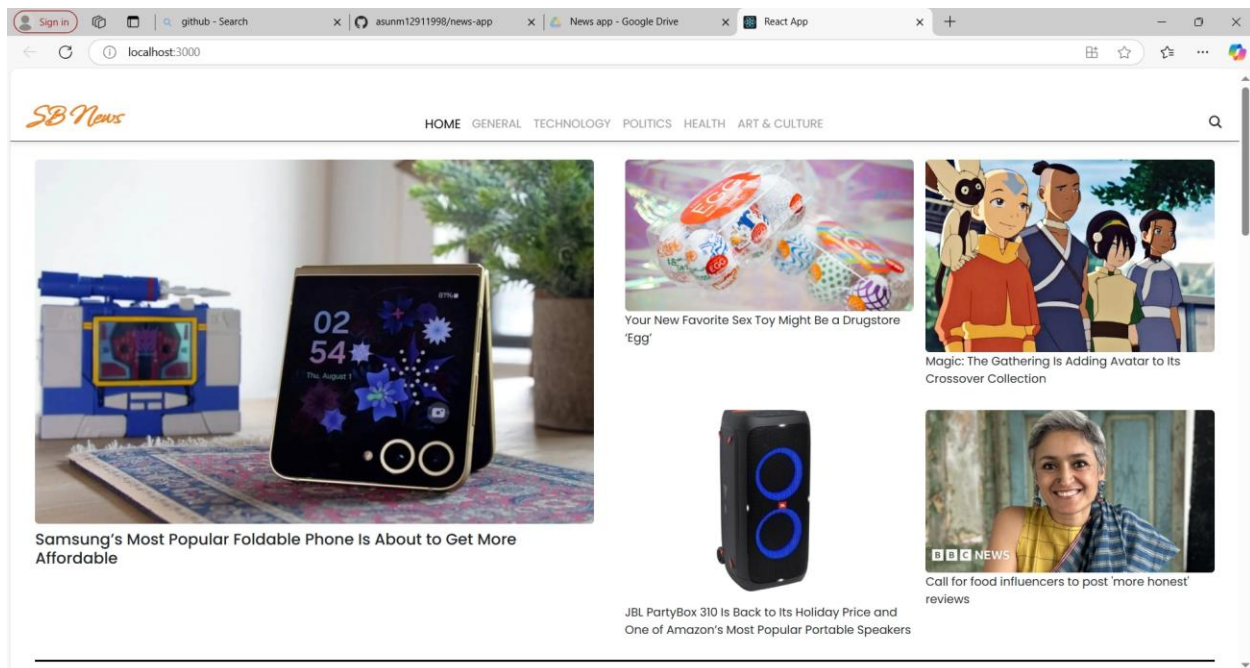
- **Unit Testing:** Jest and React Testing Library for testing individual components.
- **Integration Testing:** Ensures components work together as expected.
- **End-to-End Testing:** Cypress for testing user flows.

Code Coverage:

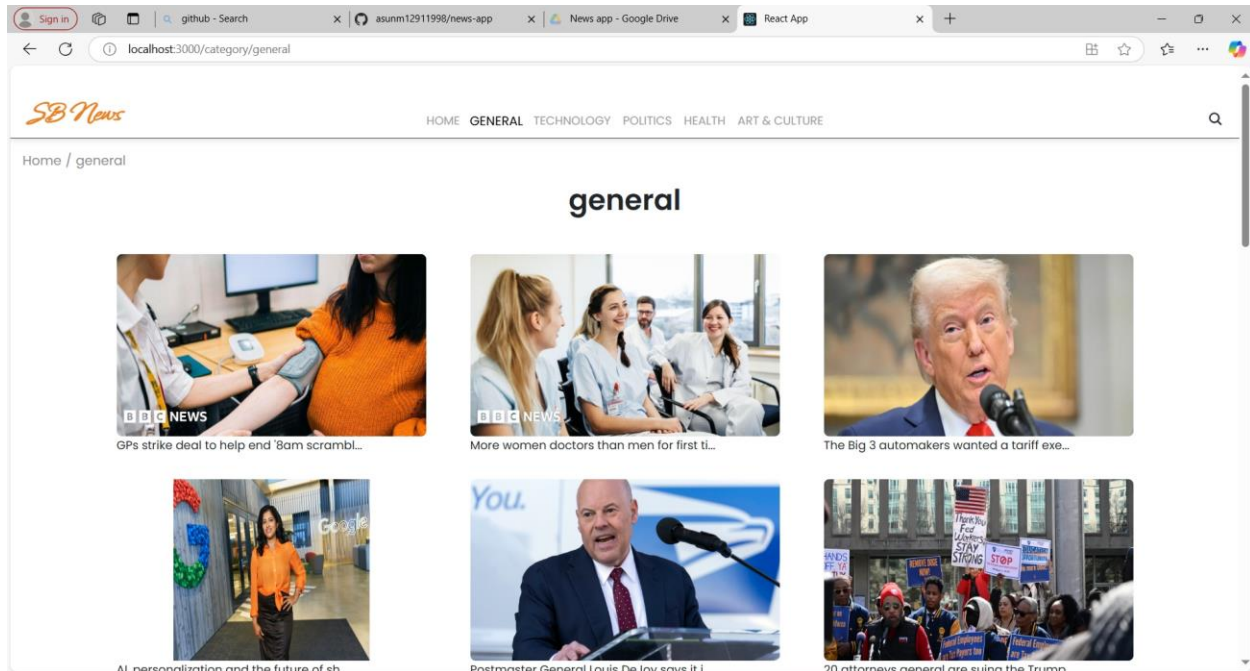
- Code coverage is tracked using Jest's coverage report.
-

12. Screenshots

DASHBOARD:



SEARCH PAGE:

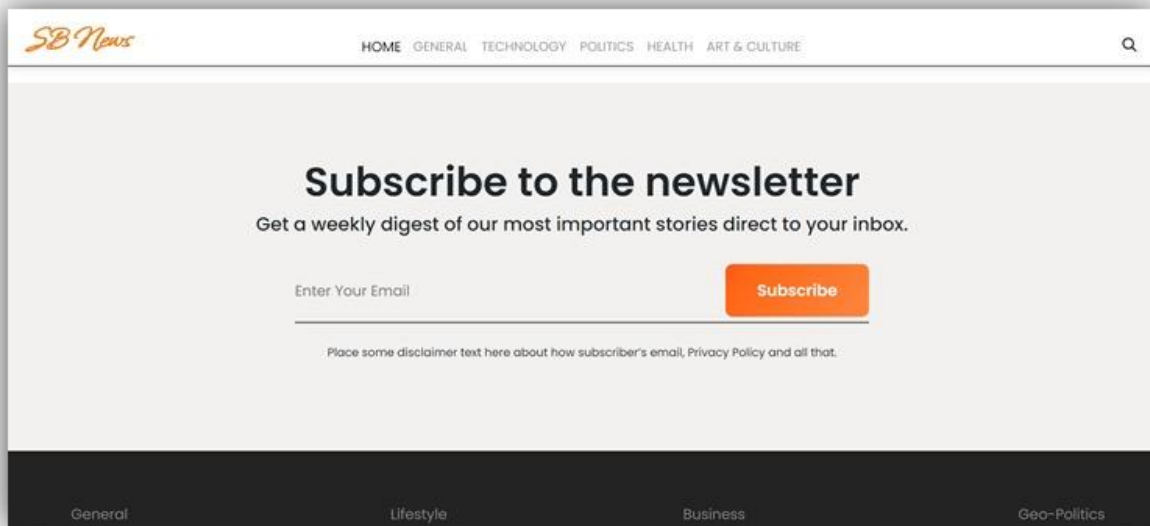


PROJECT EXECUTION:

Github link - <https://github.com/asunm12911998/news-app.git>

After completing the code react application by using the command "npm start" or "npm run dev"

NEWS LETTER:



13. Known Issues

1. Fake News & Misinformation

● **Issue:** Users may come across false or misleading news, reducing credibility.

✓ **Solution:**

- Implement AI-based fact-checking and credibility scoring.
- Partner with verified news agencies and sources.
- Allow users to report fake news for review.

2. Slow Loading & Performance Issues

● **Issue:** Heavy images, videos, and ads can slow down the app.

✓ **Solution:**

- Optimize content loading with lazy loading techniques.
- Implement caching and content delivery networks (CDNs).
- Reduce excessive background processes.

3. Poor Personalization & Recommendation Accuracy

● **Issue:** Users receive irrelevant or repetitive news articles.

✓ **Solution:**

- Enhance AI algorithms for better content curation.
- Allow users to manually adjust their preferences.
- Use behavior-based learning to refine recommendations.

4. Privacy & Data Security Concerns

● **Issue:** Unauthorized data tracking and privacy breaches.

✓ **Solution:**

- Ensure GDPR/CCPA compliance with transparent data policies.
- Implement end-to-end encryption for user data.
- Allow users to control data-sharing settings.

5. Excessive Ads & Paywalls

● **Issue:** Too many ads can frustrate users, and paywalls may limit access.

✓ **Solution:**

- Offer an ad-free subscription model.
- Balance free and premium content without forcing paywalls.
- Implement native ads that blend seamlessly with content.