INSIGHTSTREAM: Navigate the News Landscape (React Application)

TEAM ID: SWTID1741252677158511

TEAM SIZE: 4

TEAM LEADER:	EMAIL ID:
Srikanth R	kanthsri200428@gmail.com
TEAM MEMBERS:	EMAIL ID:
Udhayakumar S	uudhaya422@gmail.com
Vijay S	ssssvijay338@gmail.com
Dhanusri K	bagiyalakshmilakshmi@gmail.com

INTRODUCTION:

PROJECT TITLE:

INSIGHTSTREAM:Navigate the newslandscape(reactapplication)

In a world of constant change, staying informed is more than a choice—it’s a necessity. InsightSteam is your guide to navigating the evolving landscape of news, trends, and insights. We bridge the gap between information and action, delivering data-driven analysis, expert perspectives, and real-time updates to help you make smarter decisions.

With a commitment to clarity, accuracy, and innovation, InsightSteam transforms complexity into actionable intelligence. Whether you’re tracking industry shifts, market dynamics, or global developments, we empower you to Navigate the Newscape with confidence.

- **Project Overview:**

PURPOSE:

Insight Stream News Navigate is a dynamic news aggregation and navigation system designed to help users access categorized, real-time news updates efficiently. It provides a smooth and interactive experience, allowing users to browse, filter, and personalize their news feeds.

The News App is a frontend web application designed to deliver the latest news articles from different categories like technology, business, sports, entertainment, and more. The app fetches real-time data from a news API and presents it in a clean, user-friendly interface.

FEATURES:

✓ News from API Sources: Access a vast library of global news spanning various categories and interests, ensuring a well-rounded coverage of current affairs.

✓ Visual News Exploration: Discover breaking stories and explore different news categories through curated image galleries, enhancing the visual appeal of news discovery.

✓ Intuitive Design: Navigate the application seamlessly with a clean, modern interface designed for

optimal user experience and clarity in information presentation.

✓ Advanced Search Feature: Easily access news articles on specific topics through a powerful search feature, providing users with tailored news content based on their interests.

- **Architecture:**

1. Component Structure:

The News App is built with a modular approach using React components:

App.js: The root component, responsible for routing and managing high-level state.

Navbar.js: A navigation bar with category links like Technology, Business, Sports, etc.

NewsList.js: Fetches and displays a list of news articles. It maps through the data and renders individual news items.

NewsItem.js: A reusable component that shows article details — image, title, description, and a "Read More" link.

SearchBar.js: Provides search functionality, filtering articles based on user input.

. State Management:

We’re using React’s built-in useState and useEffect for state handling:

Global state:

✓ Selected news category.

✓Search query.

✓Fetched news articles.

Local State:

✓Component-level states for input values and UI controls.

Routing:

Routing is managed with react-router-dom, enabling category-based navigation:

/ → Home (top headlines)

/category/:name → Category-specific news

/search → Search results page.

4.Setup Instructions:

PRE-REQUISITES:

Here are the key prerequisites for developing a frontend application using React.js:

✓ Node.js and npm:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications. Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

● Download: https://nodejs.org/en/download/

● **Installation instructions:**
https://nodejs.org/en/download/package-manager/

✓ **React.js: React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.**

Install React.js, a JavaScript library for building user interfaces.

● **Create a new React app:**

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

● **Navigate to the project directory: cd my-react-app**

● **Running the React App: With the React app created, you can now start the development server and see your React application in action.**

● **Start the development server:**

npm start

This command launches the development server, and you can access your React app at http://localhost:3000 in your web browser.

✓ **HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.**

✓ **Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.**

• Git: Download and installation instructions can be found at: https://git-scm.com/downloads.

✓ Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from https://code.visualstudio.com/download

• Sublime Text: Download from https://www.sublimetext.com/download

• WebStorm: Download from https://www.jetbrains.com/webstorm/download

Installation:

To install and run the Application project from google drive:

Follow below steps:

✓ Get the code: • Download the code from the drive link given below:

• Navigate into the cloned repository directory and install libraries:

```
cd news-app-react
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 applications homepage, indicating that the installation and setup were successful.

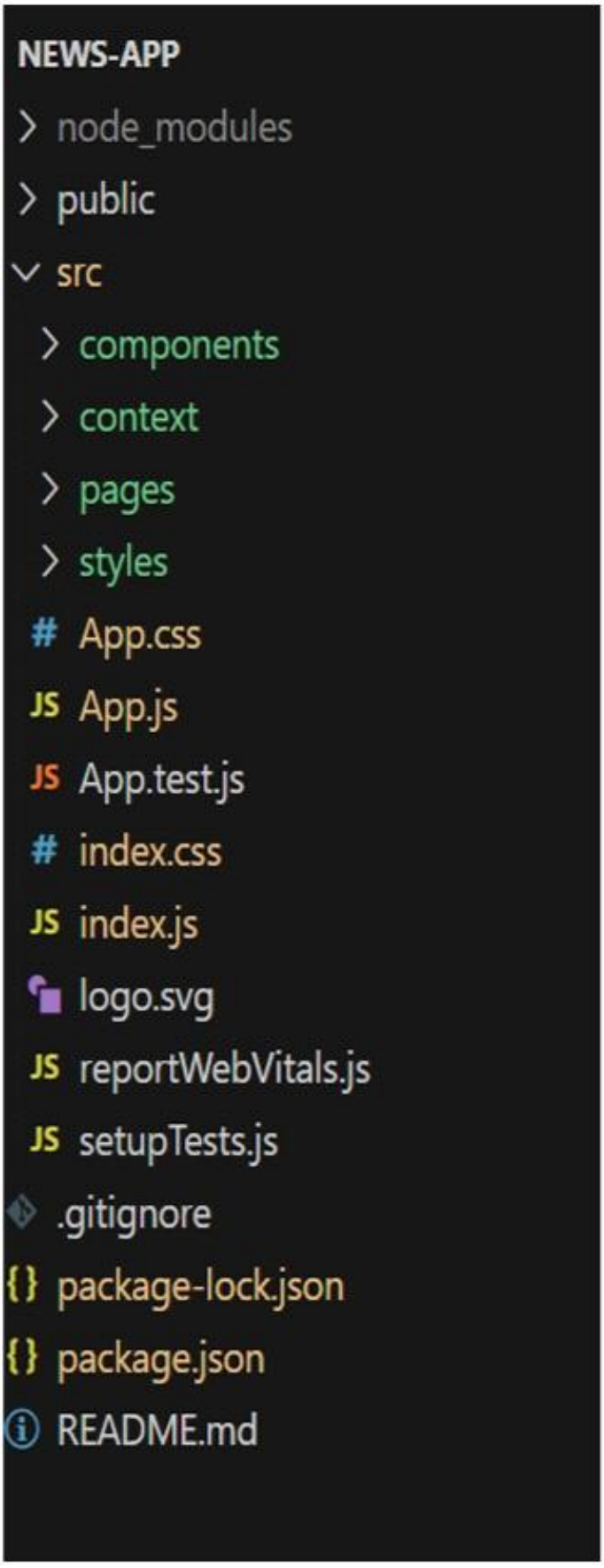
You have successfully installed and set up the application on your local machine.

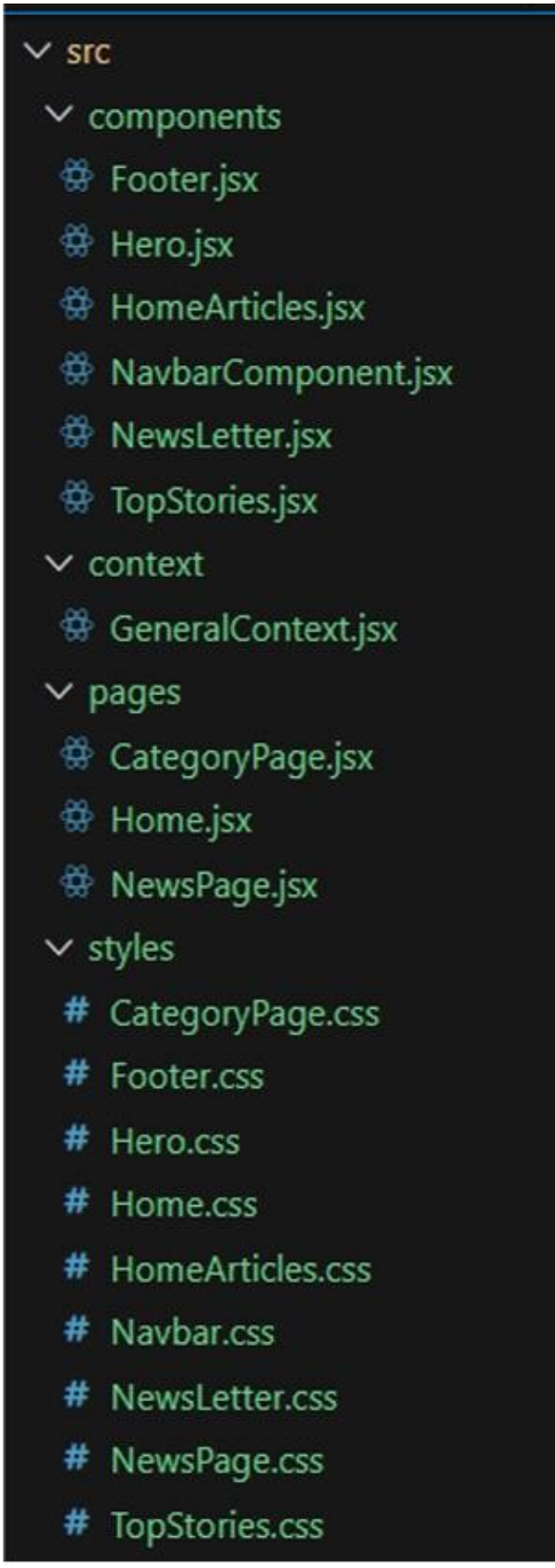
You can now proceed with further customization, development, and testing as needed.

5.Folder Structure:

Client:

In this project, we've split the files into 4 major folders, Components, Context, Pages and Styles. In the pages folder, we store the files that acts as pages at different URLs in the application. The components folder stores all the files, that returns the small components in the application. The context Api will be coded in the context folder. All the styling css files will be stored in the styles folder.


NEWS-APP
node_modules
public
src
components
context
pages
styles
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
README.md


src
components
Footer.jsx
Hero.jsx
HomeArticles.jsx
NavbarComponent.jsx
NewsLetter.jsx
TopStories.jsx
context
GeneralContext.jsx
pages
CategoryPage.jsx
Home.jsx
NewsPage.jsx
styles
CategoryPage.css
Footer.css
Hero.css
Home.css
HomeArticles.css
Navbar.css
NewsLetter.css
NewsPage.css
TopStories.css

6.Running the Application:

Installation of required tools:

To build InsightStream, we'll need a developer's toolkit. We'll use React.js for the interactive interface, React Router Dom for seamless navigation, and Axios to fetch news data.

For visual design, we'll choose either Bootstrap or Tailwind CSS for pre-built styles and icons.

Open the project folder to install necessary tools. In this project, we use:

- **React Js**
- **React Router Dom**
- **React Icons**
- **Bootstrap/tailwind css**
- **Axios**

7.Component documentation

1. Key Components:

App.js

Purpose: Root component that manages routing and global state.

Props: None.

State:

category → Stores the selected news category.

searchQuery → Stores the user's search input.

Navbar.js

Purpose: Navigation bar for selecting news categories and accessing search.

Props:

onCategoryChange(category) → Function to set the current category.

State: None.

NewsList.js

Purpose: Fetches and displays a list of news articles.

Props:

articles → Array of news articles to display.

State:

loading → Boolean to show a loading indicator while fetching data.

NewsItem.js

Purpose: Reusable component that displays individual news articles.

Props:

title → Article headline.

description → Short summary of the article.

imageUrl → URL of the article's thumbnail image.

newsUrl → Link to the full article.

State: None.

SearchBar.js

Purpose: Allows users to search for news articles by keyword.

Props:

onSearch(query) → Function to handle search input.

State:

searchInput → Stores the current value of the search field.

2. Reusable Components:

Button.js (optional)

Purpose: A customizable button used across the app.

Props:

label → Button text.

onClick() → Click event handler.

style → Custom inline styles.

Loader.js (optional)

Purpose: Displays a spinner while content is loading.

Props: None.

State: None.

8.**State management**

We're using React's built-in useState and useEffect for state handling:

Global state:

✓ Selected news category.

✓Search query.

✓Fetched news articles.

Local State:

✓Component-level states for input values and UI controls.

Routing:

Routing is managed with react-router-dom, enabling category-based navigation:

/ → Home (top headlines)

/category/:name → Category-specific news

/search → Search results page

9.User Interface

Hero components

In the hero component, the trending news articles are displayed. It is to highlight

them. Apart from that, the search bar is also available to search for various

articles and categories.

Popular categories

In the hero component, the trending news articles are displayed. It is to

highlight them. Apart from that, the search bar is also available to search for
various articles and categories.

Newsletter

Staying informed is key! This section would act as a magnet for users who want
to stay up-to-date on the latest news. A brief signup form with an email field
would be presented, along with a clear call to action button like "Subscribe Now"
or "Get Daily News Updates." With a simple click, users can join the
InsightStream community and receive curated news delivered straight to their
inbox.

Category/Search result page

Finding the news you crave is effortless with InsightStream. This page displays a
neatly organized list of articles matching your chosen category or specific search
query. Each entry would provide a clear headline, a concise summary, and if
available, an image to give you a quick glimpse into the story. To further refine

your exploration, filters or sorting options might be available.

Redirected Article page

This is where you dive deep! The article page proudly displays the complete

news story, retrieved directly from the original source. To keep you engaged and

exploring related topics, the page might also suggest additional articles based on

the current story. These suggestions can open doors to a world of interconnected

information, allowing you to become a well-rounded news connoisseur.

10.styling

1. CSS Frameworks/Libraries

Your current project uses CSS for styling without a framework. However, if you plan to enhance it, you can use CSS frameworks like Tailwind CSS or Bootstrap for better styling and responsiveness.

✅ Current CSS Approach

Uses plain CSS (style.css) for layout, colors, fonts, and responsiveness.

Includes custom styles for homepage, category pages, and recipe details pages.

✅ Future Enhancements (Optional)

Tailwind CSS – A utility-first CSS framework for faster styling.

npm install -D tailwindcss

Bootstrap – Provides pre-designed UI components for buttons, cards, and grids.

npm install bootstrap

2. Theming

Currently, the project does not have a dark/light theme switch or a custom design system. However, you can implement:

✅ Custom Theming with CSS Variables (Example)

```
:root {
  --primary-color: #30ac72;
  --background-color: #f5f5f5;
  --text-color: #333;
}
body {
  background: var(--background-color);
  color: var(--text-color);
```

}

This approach allows easy theme customization without modifying individual styles.

11.Testing

1. Testing Strategy

For a React-based version of My Recipe Book, testing can be divided into three levels:

✅ Unit Testing (Jest + React Testing Library)

Focuses on individual components to ensure they render correctly.

Example: Testing if RecipeCard.js correctly displays a recipe title and image.

Example Test for RecipeCard.js

```
import { render, screen } from "@testing-library/react";
import RecipeCard from "../components/RecipeCard";
test("renders recipe title and image", () => {
 render(<RecipeCard title="Grilled Cheese" image="/test-img.png" />);
   expect(screen.getByText("Grilled Cheese")).toBeInTheDocument();
```

```
expect(screen.getByRole("img")).toHaveAttribute("
src", "/test-img.png");
});
```

✅ Integration Testing (React Testing Library + Mock Data)

Ensures multiple components work together correctly.

Example: Testing if clicking a recipe card navigates to the RecipeDetails page.

Example Test for CategoryPage.js

```
import { render, screen } from "@testing-
library/react";
import { MemoryRouter } from "react-router-dom";
import CategoryPage from "../pages/CategoryPage";
test("renders recipes from a category", () => {
 render(
  <MemoryRouter>
   <CategoryPage categoryType="Meals" />
  </MemoryRouter>
 );

expect(screen.getByText("Meals")).toBeInTheDocu
ment();
```

});

✅ End-to-End (E2E) Testing (Cypress or Playwright)

Tests user interactions like clicking buttons, navigating pages, and checking ingredients.

Example: Simulating a user searching for a recipe and marking ingredients as used.

Example Cypress Test

```
describe("Recipe App User Flow", () => {
  it("should navigate to a recipe and check ingredients", () => {
    cy.visit("/");
    cy.get(".recipe-card").first().click();
    cy.get("input[type='checkbox']").first().check();
  });
});
```

2. Code Coverage

Ensuring that all components and logic are tested properly can be done using Jest's built-in coverage tool.

✅ Run Code Coverage Report

npm test -- --coverage

This will generate a report showing which parts of the code are covered by tests.

PROJECT DEMO LINK:

https://drive.google.com/file/d/1pvtZ9PuO5EIr7En_9f5aVwa8tVH920RI/view?usp=drivesdk

THANK YOU!