**Newsify: Navigate the News Landscape**

**(React Application)**

**Team Leader: Yazhini R**

**Team Members: Safrin Fathima K, Preethi K, Sai Sruthy G**

**Introduction**

Newsify is a modern web application designed to revolutionize how users engage with news. It offers a seamless experience, allowing users to explore real-time updates, trending stories, and in-depth articles across multiple categories. With an intuitive interface and smart filtering, Newsify ensures that staying informed is effortless and enjoyable.

**Description**

Welcome to Newsify!

Your one-stop platform for discovering,exploring, and consuming news in a whole new way. Newsify is crafted for news enthusiasts, journalists, and everyday readers who want instant access to the latest happenings worldwide.

With a clean design and smooth navigation, Newsify presents curated news feeds based on user preferences. Whether it's politics, technology, sports, or entertainment, you can personalize your news feed and explore stories that matter most to you.

Our vision is to provide a comprehensive, well-structured, and interactive platform that transforms the way people engage with information. Newsify not only delivers breaking news but also encourages discussion, content sharing, and community engagement.

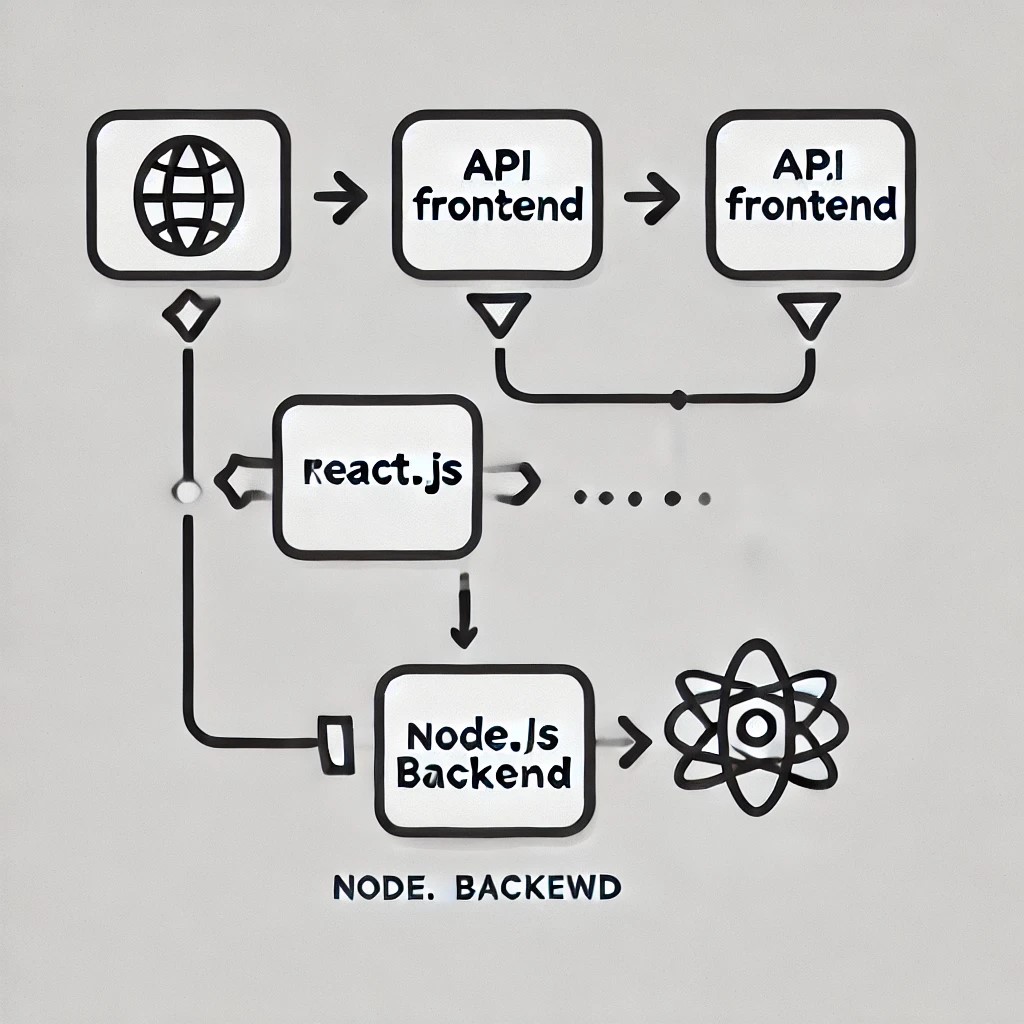
Stay connected with the world through a dynamic, AI-powered platform that brings you real-time news with accuracy and relevance.

**Scenario-Based Introduction**

Imagine you’re on your way home after a long day, and you need a quick update on the latest headlines. Instead of scrolling endlessly through social media, you open Newsify. Instantly, you see a personalized feed with breaking news, trending stories, and updates from your favorite sources. With just a few taps, you stay informed on the go.

**Technical Architecture**

Newsify is built as a single-page web application (SPA) using “React.js” for the frontend and “Node.js” with “Express.js” for the backend. It integrates with external APIs to fetch real-time news updates and ensures smooth performance with optimized data handling.



**Technology Stack**

* Frontend: React.js, React Router, Redux (for state management)
* Backend: Node.js, Express.js
* Database: MongoDB (for user preferences and saved news articles)
* API Integration: News API, RapidAPI
* Styling: Tailwind CSS / Bootstrap
* Version Control: Git & GitHub

**Project Goals and Objectives**

* User-Friendly Interface: Ensure a seamless and engaging user experience.
* Personalized News Feeds: Allow users to customize their content based on preferences.
* Efficient News Fetching: Integrate APIs for real-time news updates.
* Responsive Design: Optimize for both desktop and mobile users.

**Features of Newsify**

* Breaking News: Get real-time updates on the latest events.
* Category-Based News: Browse through specific categories like Business, Technology, Sports, and more.
* Advanced Search: Find news articles with an efficient search feature.
* User Profiles: Create personalized preferences and follow specific topics.

**Pre-Requisites**

To run Newsify locally, ensure you have the following installed:

* Node.js & npm : Install from [Node.js Official Website](<https://nodejs.org/en/download/>)
* React.js: npx create-react-app newsify
* Git: Download from [Git SCM](<https://git-scm.com/downloads>)
* IDE: Recommended - [VS Code](<https://code.visualstudio.com/download>)

**Project Setup & Installation**

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>

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: <https://drive.google.com/file/d/12f78YMPSw3hgbjDxgoviRWf6mS4Ncum6/view?usp=drivesdk>

cd news-app-react npm install

✔ Start the Development Server:

• To start the development server, execute the following command:

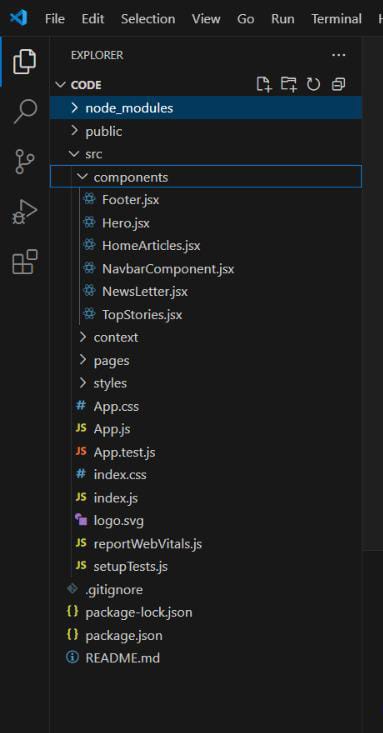
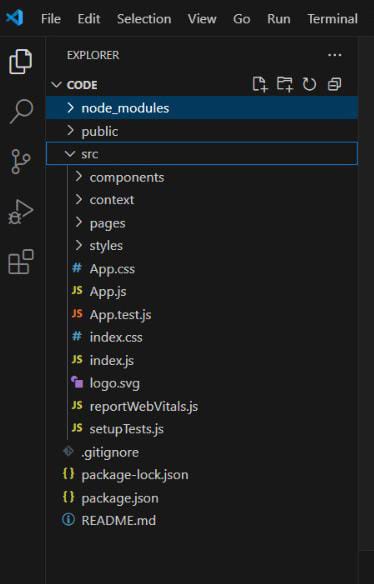
npm start

Access the App:

• Open your web browser and navigate to [React App](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.

**Project Structure:**

****

**Components**→ Contains reusable UI components.

**Pages**→ Includes different pages like Home, Categories, and Search Results.

**Context** → Manages global state (e.g., user preferences).

**Styles** → Stores CSS and styling files.

**Services** → Handles API calls and data fetching

**Milestone 1: Project setup and configuration.**

* **Installation of required tools**:

To build Newsify, 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.

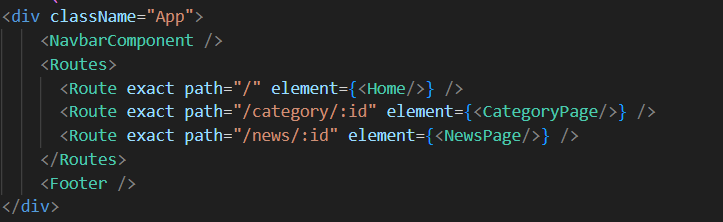
Open the project folder to install necessary tools. In this project, we use:

* React Js
* React Router Dom
* React Icons
* Bootstrap/tailwind css
* Axios
* For further reference, use the following resources
* <https://react.dev/learn/installation>
* <https://react-bootstrap-v4.netlify.app/getting-started/introduction/>
* <https://axios-http.com/docs/intro>
* <https://reactrouter.com/en/main/start/tutorial>

**Milestone 2: Project Development**

* Setup the Routing paths

Setup the clear routing paths to access various files in the application.

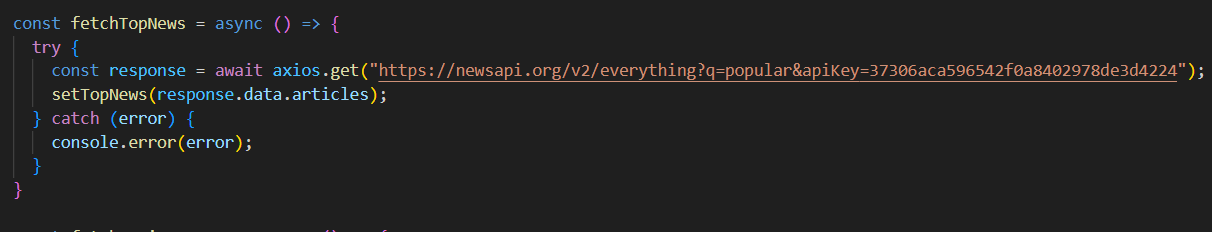


* Develop the Navbar and Hero components
* Code the popular categories components and fetch the categories from **newsapi**.
* Also, add the trending news in the home page.
* Additionally, we can add the component to subscribe for the newsletter and the footer.
* Now, develop the category page to display various news articles under the different categories.

**Important Code snips:**

* **Fetching Top/Trending news**

With the API request, we fetch the trending news articles.

****

The code snippet shows a function written in Python called fetchTopNews that fetches news articles from an API. Here's a breakdown of the code:

Async function fetchTopNews:

The code defines an asynchronous function named fetchTopNews. An asynchronous function is used to handle asynchronous operations, such as making API requests that take time to complete.

try...catch block:

* The try...catch block is used to handle the API request.
* The try block contains the code that attempts to fetch data from the API using axios.get.
* axios is an external Python library for making HTTP requests. If you don't already use Axios in your project, you'll need to install it using a package manager like pip.
* The .get method makes a GET request to the specified URL.

Error Handling (catch block):

The catch block handles any errors that might occur during the API request. If there's an error, it's logged to the console using console.error(error).

Setting State (then block not shown):

The .then method (not shown in the code snippet) is likely used to process the fetched data after a successful API request.

In this case, it likely updates a state variable named topNews (based on the function name fetchTopNews) with the fetched news articles. This state variable might be used to display the news articles in a user interface.

* **Fetching news by search/category**

With the specific category or search keyword, we use API request to fetch all the news articles related to that.

****

The code snippet shows a function called get\_news\_articles that fetches news articles from a news API. Here's a breakdown of the code:

Imports:

The code starts by importing the requests library. The requests library is a popular Python library for making HTTP requests. If you don't already have it installed in your project, you can install it using pip install requests.

API Key:

The line API\_KEY = 'YOUR\_API\_KEY' defines a variable named API\_KEY and assigns it a placeholder value 'YOUR\_API\_KEY'. You should replace this with a placeholder instructing users to replace it with their own API key obtained from the news API provider they want to use.

Function Definition (get\_news\_articles):

The code defines a function named get\_news\_articles that takes two parameters:

* query: This parameter is likely a string representing the search query for news articles.
* source: This parameter is likely a string representing the news source (e.g., 'bbc-news', 'cnn').

Building the API Request URL:

The line url = f'https://newsapi.org/v2/everything?q={query}&apiKey={API\_KEY}' constructs the URL for the API request using a formatted string literal (f-string).

The URL includes the following parts:

* Base URL: https://newsapi.org/v2/everything
* Query parameters:
* q: This parameter is set to the query argument passed to the function.
* apiKey: This parameter is set to the API\_KEY variable, which should contain the user's API key.

Making the API Request (requests.get):

The line response = requests.get(url) sends a GET request to the API URL constructed earlier. The requests.get function from the requests library is used to make the HTTP request. The response from the API is stored in the response variable.

Error Handling (try...except block):

* The try...except block is used to handle potential errors during the API request.
* The try block contains the code that attempts to fetch data from the API using requests.get(url).
* The except block handles any exceptions that might occur during the request, such as network errors or invalid API responses. In this case, it prints an error message to the console using print(f'Error fetching news articles: {e}').

.

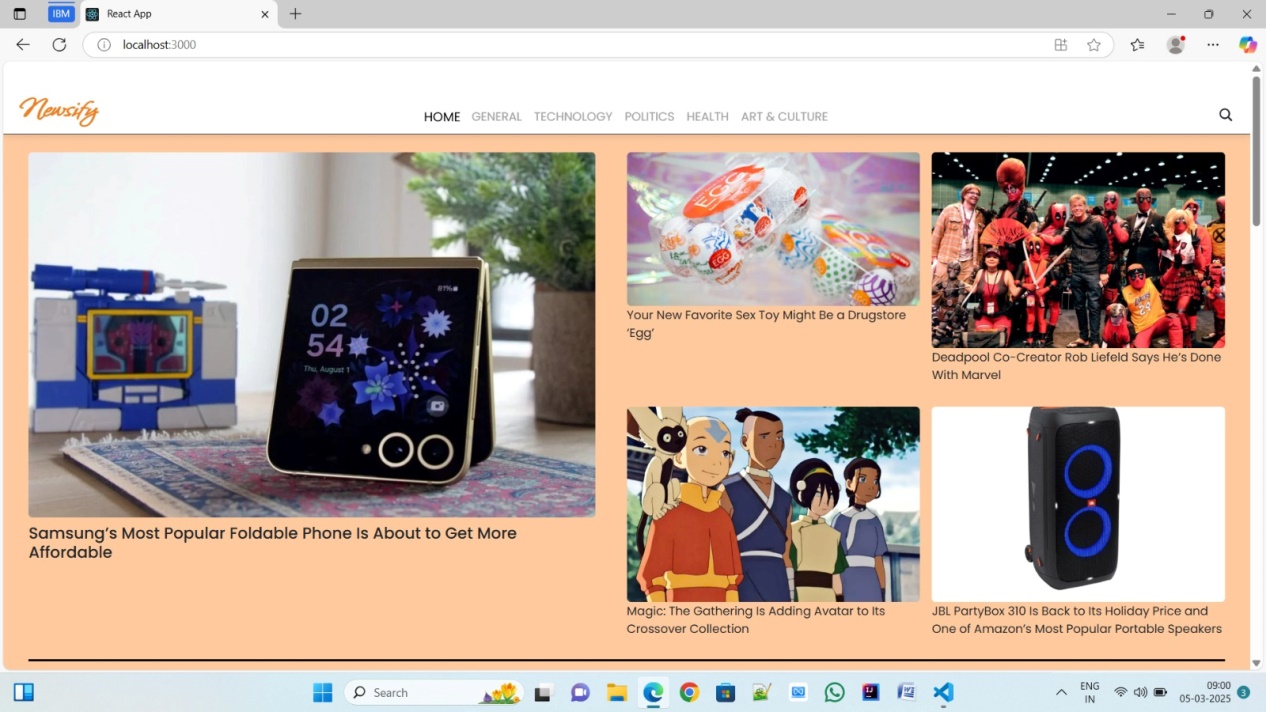
**Project Execution:**

After completing the code, run the react application by using the command “npm start” or “npm run dev” if you are using vite.js

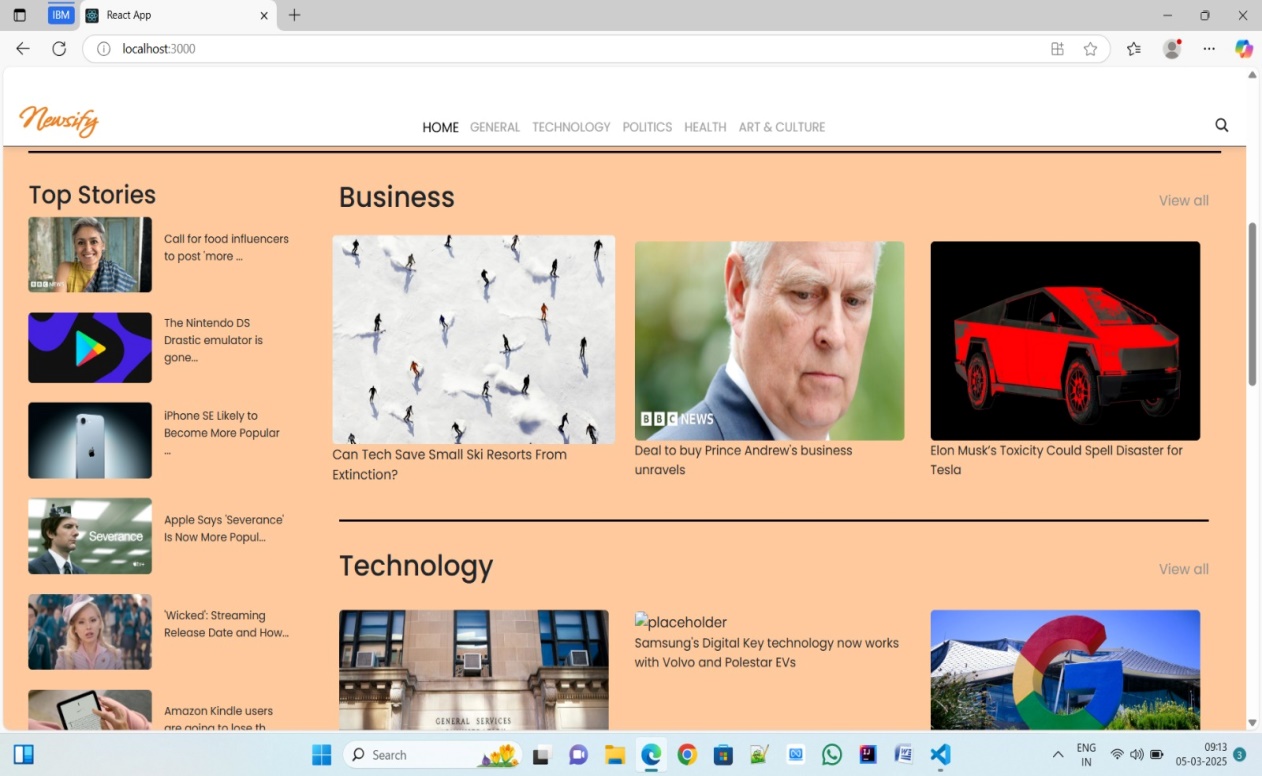
Here are some of the screenshots of the application.

* **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.

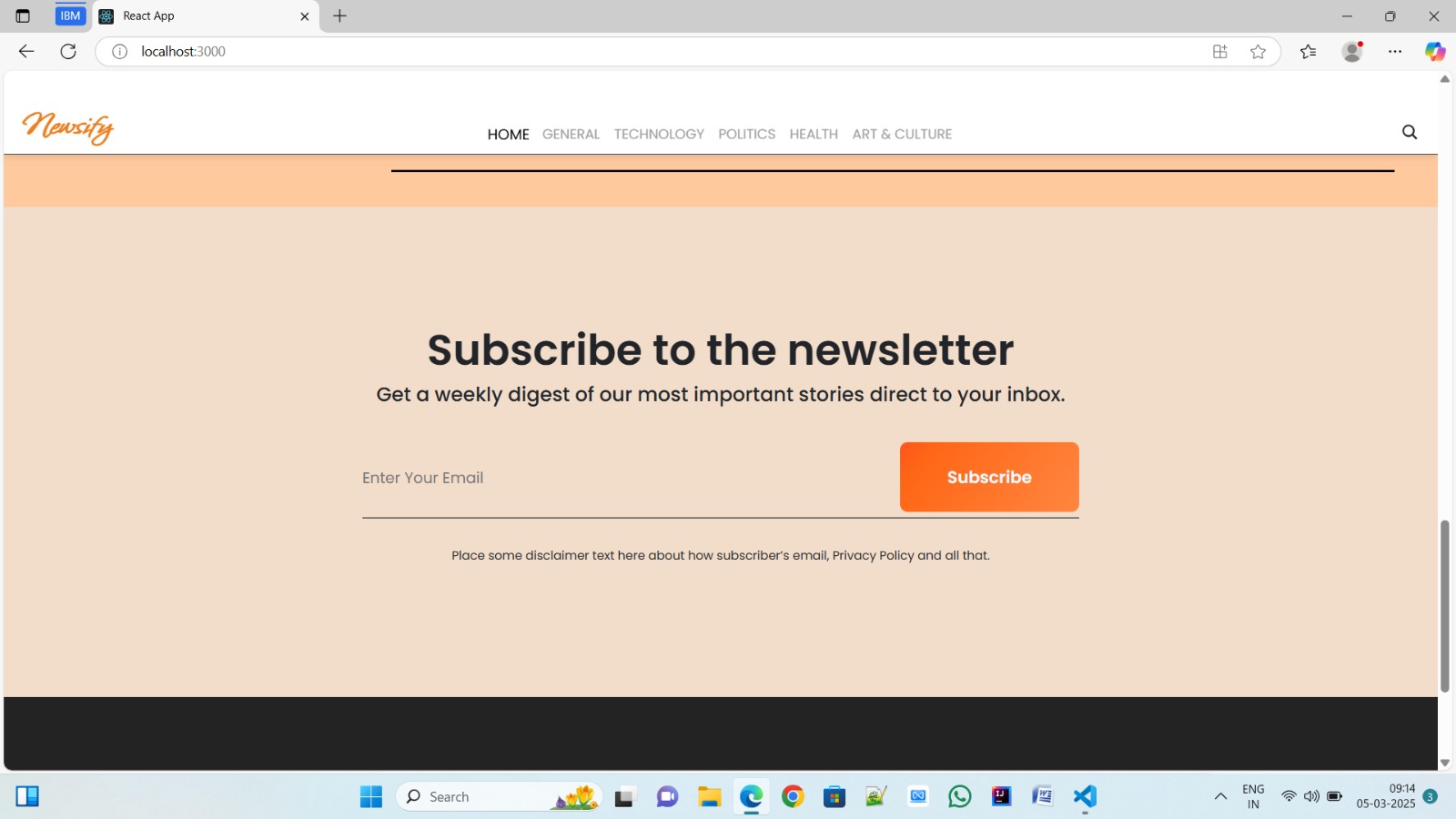
****

* **Popular categories**

****

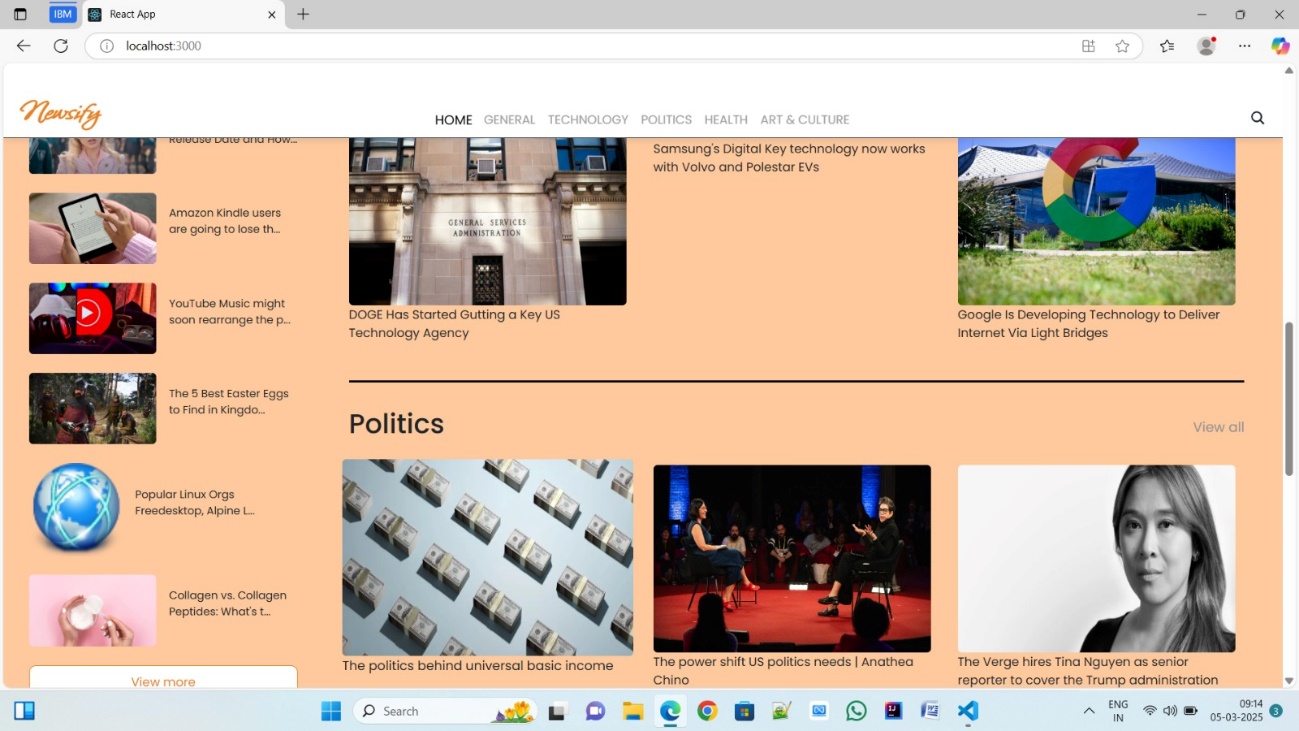
The **Popular Categories** feature in Newsify allows users to explore news effortlessly by browsing through well-organized sections like Business, Technology, Sports, and more. It highlights trending topics, ensuring quick access to relevant articles. With a user-friendly layout and smart filtering, staying updated on preferred subjects becomes seamless.

* **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 ‘Newsify’ community and receive curated news delivered straight to their inbox

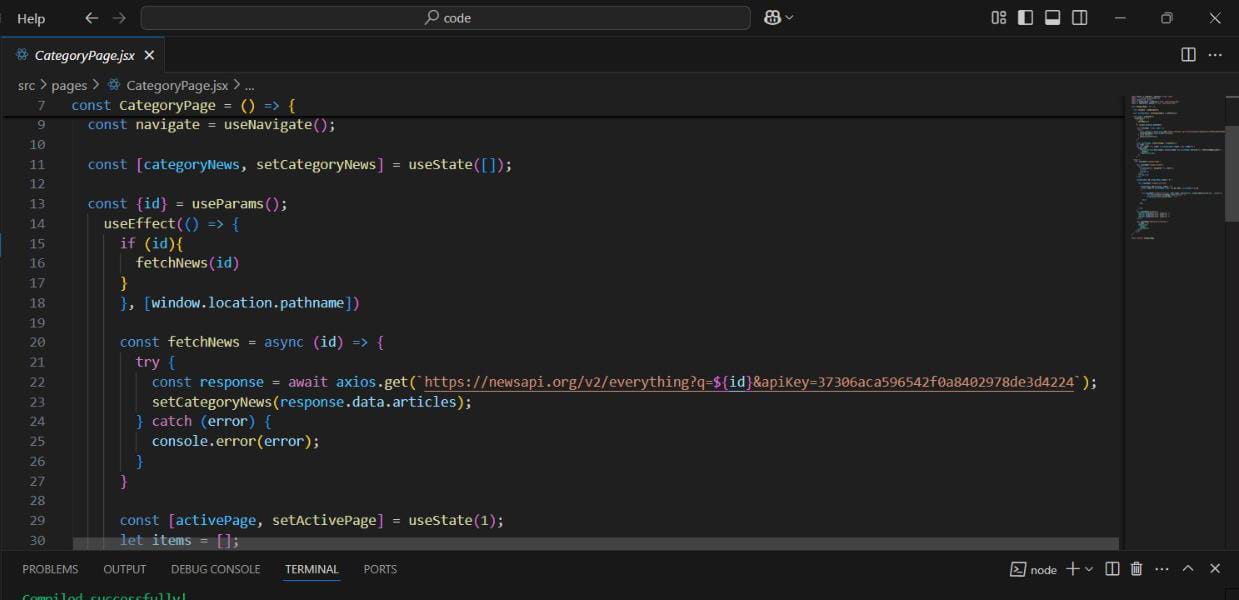
* **Category/Search result page**



Finding the news you crave is effortless withNewsify. 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. Imagine narrowing down , source, keyword, or other relevant criteria to pinpoint exactly what you're looking for.

results by date

**Project Development Milestones**



**Project Demo**

Demo:[https://drive.google.com/file/d/12byLf-hE\_uKZLhb6FmlxO7BDJyZfOwNC/view?usp=drivesdk](https://drive.google.com/file/d/12byLf-hE_uKZLhb6FmlxO7BDJyZfOwNC/view?usp=drivesdk%20)

**Conclusion**

Newsify aims to provide an innovative and engaging news consumption experience. With real-time updates, personalized feeds, and intuitive design, it empowers users to stay informed efficiently. Future enhancements will include AI-powered recommendations, voice search, and multilingual support.

***Stay updated, stay ahead – with Newsify*!!!**