PROJECT DOCUMENTATION

INTRODUCTION

Project Title: NEWS INSIGHTS

TEAM MEMBERS	EMAIL ID
Jaishree G	jaishree@18112004@gmail.com
Deepika B	deepisuji1707@gmail.com
Charulatha D	rsai99075@gmail.com
Jenifer Mary S	jenifermaryjeni8@gmail.com

PROJECT OVERVIEW:

News Insights is an innovative web application designed for users to explore, organize, and create news. It caters to both beginner and professional insights, providing an intuitive user experience and a vast collection of news insights.

PURPOSE:

News insights aims to revolutionize the way users interact by offering a seamless platform for discovering, saving, and sharing culinary inspirations. The main goals are:

- **User-Friendly Experience** – Easy navigation for discovering and managing insights.
- **Comprehensive Recipe Management** – Advanced search and categorization for efficient organization.
- **Modern Tech Stack** – Utilizing React.js and Rapid API for enhanced functionality.

FEATURES:

- **News API Integration** – Fetches meals from the MealsDB API.
- **Visual Recipe Browsing** – Image-based navigation of categories.
- **Search Functionality** – Easily find recipes using keywords.
- **Interactive UI** – Built using modern design principles for a smooth experience.

ARCHITECTURE:

Component Structure

The application is divided into three main sections:

- **Pages** – Full-page components (Home, News Category, News Details).
- **Components** – Reusable UI elements (Navbar, Search Bar, News Cards, Category Filter).
- **Styles** – CSS and styling files (Global styles, Component-specific styles).

State Management

- **Global State:** Managed using React Context API.
- **Local State:** Controlled via React's useState for component-level updates.

Routing

Implemented using React Router to enable seamless navigation between pages.

SETUP INSTRUCTIONS:

Prerequisites

- **Node.js & npm** – Install from Node.js website.
    - **React.js** – Set up a new

    project using: npx create-

    react-app my-react-app

cd my-react-app npm

Start

Installation Steps

1. **Clone the repository**

   git clone [repository URL]

2. **Navigate into the project directory**

   cd recipe-app-react

3. **Install dependencies**

   npm install

4. **Set up environment variables** (if required) by creating a .env file and adding necessary API keys.

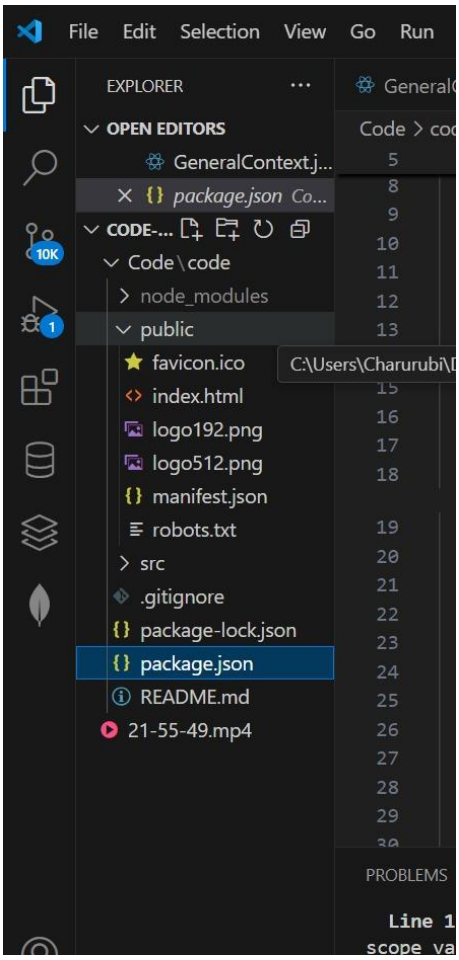
1. **Start the development server**

   npm start

2. **Access the application**
   Open http://localhost:3000 in your web browser.

FOLDER STRUCTURE

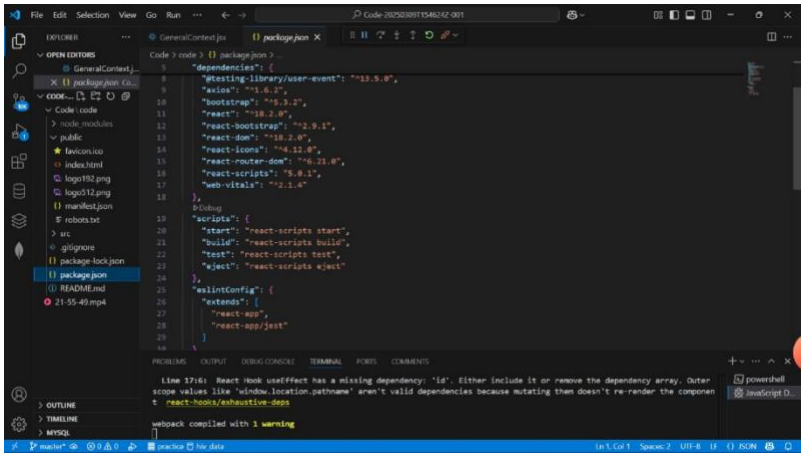
The project is structured into different directories for better organization and maintainability. Below is an overview of the folder structure.



Client

The **client** folder (inside src/) contains the core files of the frontend application. It consists of:

- **Components:** Houses all reusable UI components such as buttons, cards, and the navigation bar.
- **Pages:** Contains full-page components, including the homepage, recipe details page, and category listings.
- **Styles:** Includes all CSS or SCSS files to style the application.
- **Assets:** Stores images, icons, and other static files used throughout the app. Example structure inside src/:



RUNNING THE APPLICATION

To start the frontend server, run:

npm start

Then, open http://localhost:3000 in your browser.

COMPONENT DOCUMENTATION KEY COMPONENTS

Purpose:

Provides site-wide navigation to help users access different sections like Home, Categories, and Search**.**

Props:

- logo (string): Path to the logo image displayed in the navbar.

- menuItems (array): List of navigation items such as "Home", "Categories", and "Search".

- onSearchSubmit (function): A callback function triggered when the search form is submitted.

2. Hero Component (Hero.js)

Purpose:

- Displays an introduction to the site and a call-to-action (CTA) to engage users.

- Contains a button to encourage users to explore more content (e.g., "Explore Latest News").

Props:

- title (string): Main heading or introduction for the hero section.

- ctaText (string): Text on the call-to-action button (e.g., "Explore").

- ctaLink (string): URL or path for the CTA button to redirect users

3. News List Component (NewsList.js)

Purpose:

- Displays a list of news articles, filtered based on categories, search results, or user preferences.

Props:

- articles (array): List of news article objects (e.g., title, description, publication date).

4. News Card Component (NewsCard.js)

Purpose:

- Represents a single news article preview with a title, short description, image, and link to the full article.

Props:

- image (string): URL for the image associated with the article.

- title (string): Title of the news article.

- description (string): A brief description of the article.

- articleLink (string): URL to view the full article.

5. Categories Component (Categories.js)

Purpose:

- Displays a list of categories (e.g., Sports, Politics, Technology) to filter news articles based on users' interests.

Props:

- categories (array): List of categories (e.g., ["Politics", "Technology", "Sports"]).

- onCategorySelect (function): Callback function to filter news based on selected category.

REUSABLE COMPONENT

1. Search Bar Component (SearchBar.js)

Purpose:

- Allows users to search for news articles by keywords.

Props:

- onSearch (function): Callback function to handle the search query submitted by the user.
- placeholder (string): Placeholder text for the search input (e.g., "Search news articles...").

2. Button Component (Button.js)

Purpose:

- A reusable button that can be used for various actions across the site like submitting forms, triggering events, or navigating.

Props:

- text (string): The text that will appear on the button (e.g., "Submit", "Search").
- onClick (function): Callback function that executes when the button is clicked.
- style (string): A string for button style (e.g., "primary", "secondary").
- disabled (boolean): A flag to disable the button when set to true.

3.Card Component (Card.js)

Purpose:

- A reusable card layout for displaying content such as articles, products, or services.

Props:

- title (string): The title to be displayed on the card.
- image (string): The URL of the image to be displayed in the card.
- description (string): A short description of the content.
- link (string): A URL to link the user to the full content.

4. Modal Component (Modal.js)

Purpose:

- A reusable modal dialog to display content like confirmation, notifications, or forms without navigating away from the current page.

Props:

- isOpen (boolean): A boolean value that controls whether the modal is visible.

- title (string): The title to be displayed in the modal header.

- children (node): The content inside the modal (could be text, forms, or other components).

- onClose (function): A callback function to close the modal.

STATE MANAGEMENT

State management in News Insights (or any web application) refers to how the application handles and stores data that influences the user interface and other parts of the app. This includes managing things like user authentication, displayed articles, search results, State management in **NEWS INSIGHTS** application ensures efficient data handling across different components. The project incorporates both global state for shared data and local state for component-specific interactions.

GLOBAL STATE

In a modern web application like News Insights, global state management refers to storing and managing state that can be accessed across multiple components. This allows data to be shared between different parts of the application without the need to pass props manually through each component. For large-scale applications, global state management is essential to avoid prop-drilling and ensure better scalability and maintainability.

Global State Usage in News Insights

- User Authentication: Stores user login status (isAuthenticated) and user details (user).

- Search Results: Holds the searchQuery and articles fetched based on the query.
- Loading State:Tracks the loading status (loading) while fetching articles.

How Local State Works in Components:

Usage: Local state is typically used for things like form input values, toggle states, or visibility of UI elements.

Encapsulation: Local state is isolated within a component and not shared with other components.

State Updates: State is updated using the update function (e.g., setState).React re-renders the component when the state changes to reflect the updates in the UI.

User Interface

The User Interface (UI) of the News Insights project is designed to provide a clean, intuitive, and user-friendly experience for users interacting with the application. The UI is focused on accessibility, ease of navigation, and delivering the latest news in an organized and attractive layout.

1. Navigation Bar (Navbar)

- Purpose: Provides easy access to different sections of the app.
- Components:
  - Logo on the left side.
  - Navigation links: Home, Categories, Search, Favorites, Profile, Login/Logout.
  - Responsive design for mobile and desktop views.
  - Dropdown menu for categories or user profile options.

2. Hero Section

- Purpose: Serves as the main introduction or welcome area of the app.
- Components:
  - Headline text or call-to-action (CTA) inviting users to explore news.

- A search bar for users to quickly find news topics.
- Featured images or news headlines to draw attention to important stories.

3. Search Bar

- Purpose: Allows users to search for articles by keywords or topics.
- Components:
  - Input field for the search term.
  - Search button to trigger the query.

Auto-suggestions or recent searches for quick access.

4. Article List / News Feed

- Purpose: Displays a list of news articles based on search query or category.
- Components:
  - Card-like layout for each article displaying title, description, image, and a link to read more.
  - Pagination controls at the bottom to navigate through multiple pages of results.
  - "Load More" button for infinite scrolling or fetching more articles.

5. Article Detail Page

- Purpose: Provides in-depth content for a selected article.
- Components:
  - Article title, author, and publication date.
  - Full article content with images, videos, and text.
  - Social sharing buttons (Facebook, Twitter, etc.).
  - Option to mark as a favorite or save the article.

STYLING

Styling plays a key role in enhancing the visual appeal and user experience of the News Insights project. The goal is to create a clean, modern, and responsive design that provides easy

navigation, smooth interactions, and an intuitive layout. Below is an overview of how the styling is applied in the project.

1. General Styling Approach

- CSS Frameworks:

- Bootstrap or Tailwind CSS can be used to accelerate the development of responsive layouts and UI components.

- For custom designs, plain CSS or CSS-in-JS (e.g., styled-components) can be employed to fine-tune the application.

- Responsive Design:

- The app should be fully responsive, ensuring it works seamlessly across devices like mobile phones, tablets, and desktops.

- Media queries are used to adjust layout, font sizes, and other elements to fit different screen sizes.

- Color Scheme:

- The design should feature a modern, professional color palette with good contrast for readability. Examples:

  - Primary color: A bold color for the logo and key elements (e.g., blue, green).

  - Secondary colors: Lighter shades for backgrounds and borders (e.g., light grey, soft blue).

Text colors: Dark text for readability on light backgrounds and white text for dark sections.

2. Styling Key Components

a. Navbar Component:

- Purpose: Provides site-wide navigation.

- Styling:

  - Fixed position on top for easy access to the navigation links.

  - Background color can be dark with white text for high contrast.

  - Hover Effects on navigation links for better interactivity.

- Dropdown menus for categories or user profile, styled with hover transitions.

```
css
/* Navbar Styling */
.navbar {
  background-color: #003366;
  padding: 10px 20px;
  position: fixed;
  width: 100%;
  top: 0;
  z-index: 100;
}

.navbar a {
  color: #ffffff;
  margin: 0 15px;
  text-decoration: none;
  font-size: 18px;
}
.navbar a:hover {
  color: #ffcc00;
}
```

b. Hero Section:

- Purpose: Welcomes the user to the app with an introduction or CTA.
- Styling:
  - Full-width background image with a semi-transparent overlay to make the text stand out.
- Large text for the main headline and a prominent call-to-action (CTA) button.

- Centered content with good padding and spacing.

```
css
/* Hero Section Styling */
.hero {
  background-image: url('hero-image.jpg');
  background-size: cover;
  background-position: center;
  height: 400px;
  display: flex;
  justify-content: center;
  align-items: center;
  text-align: center;
  color: #ffffff;
  padding: 50px;
}

.hero h1 {
  font-size: 40px;
  margin-bottom: 20px;
}

.hero button {
  background-color: #ffcc00;
  color: #003366;
  padding: 15px 30px;
  border: none;
  cursor: pointer;
```

```
}

.hero button:hover {
  background-color: #003366;
  color: #ffffff;
}
```

TESTING

Testing is a crucial aspect of the News Insights project to ensure that all components and features work as expected, maintain functionality during updates, and provide a reliable experience for users. The testing process helps identify bugs, optimize performance, and verify that the system behaves as expected across various devices and use cases.

1. Unit Testing

- Purpose: To test individual functions, methods, or components in isolation, ensuring that each part works correctly.

- Tools:

  - Jest: JavaScript testing framework that allows for testing React components and functions.

2. Integration Testing

- Purpose: To test how different parts of the application work together (e.g., frontend and backend).

- Tools:

  - Jest + Enzyme: For testing React components and simulating interactions between them

3. Functional Testing

- Purpose: To validate the overall functionality of the application, ensuring that the system performs as expected from the user's perspective.

- Tools:

  - Cypress: For end-to-end (E2E) testing that mimics real user interactions.

Selenium: For automating browser interactions and checking functionality.

4. UI Testing

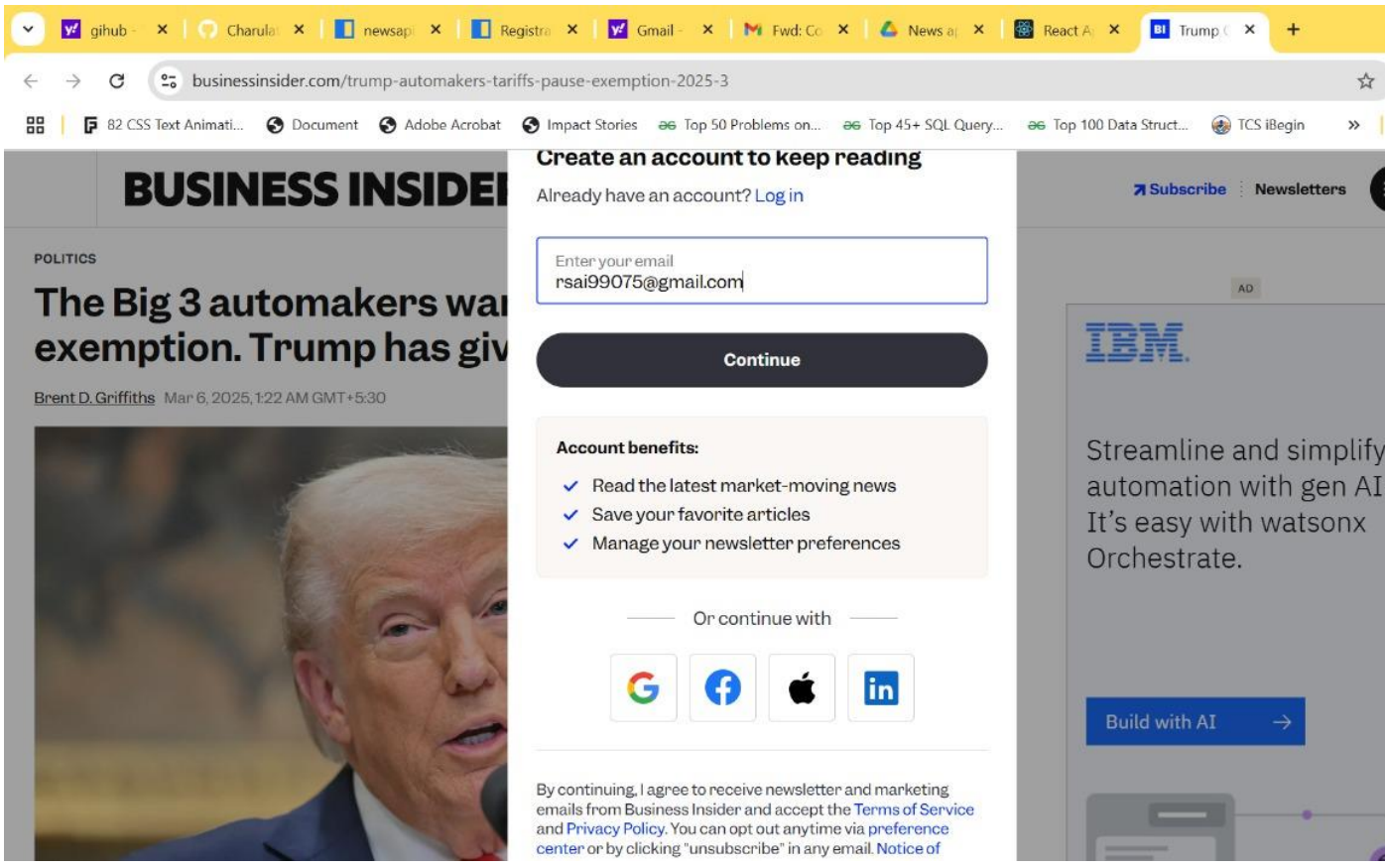
- Purpose: To verify that the user interface appears and functions as expected.

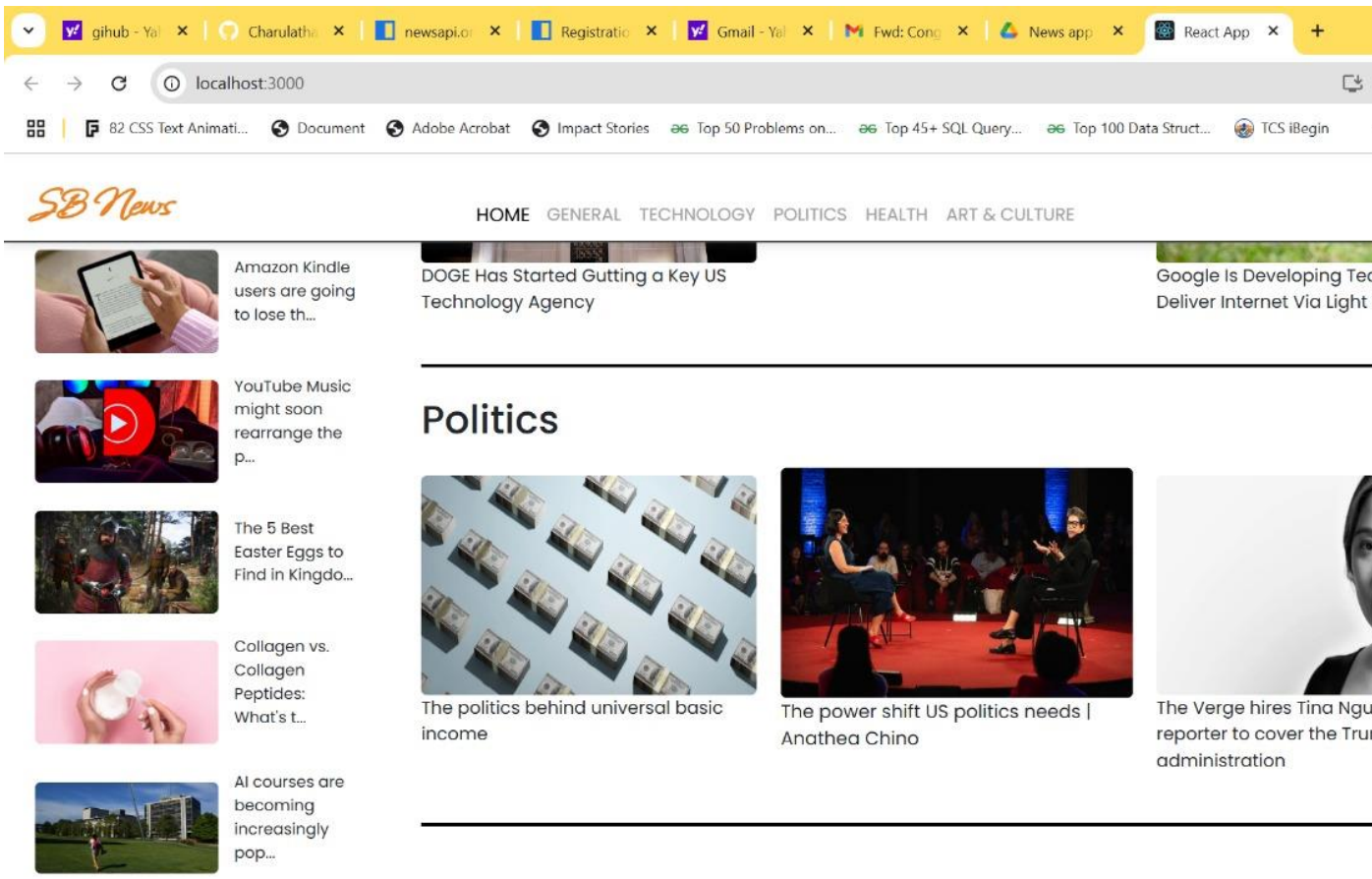
- Tools:

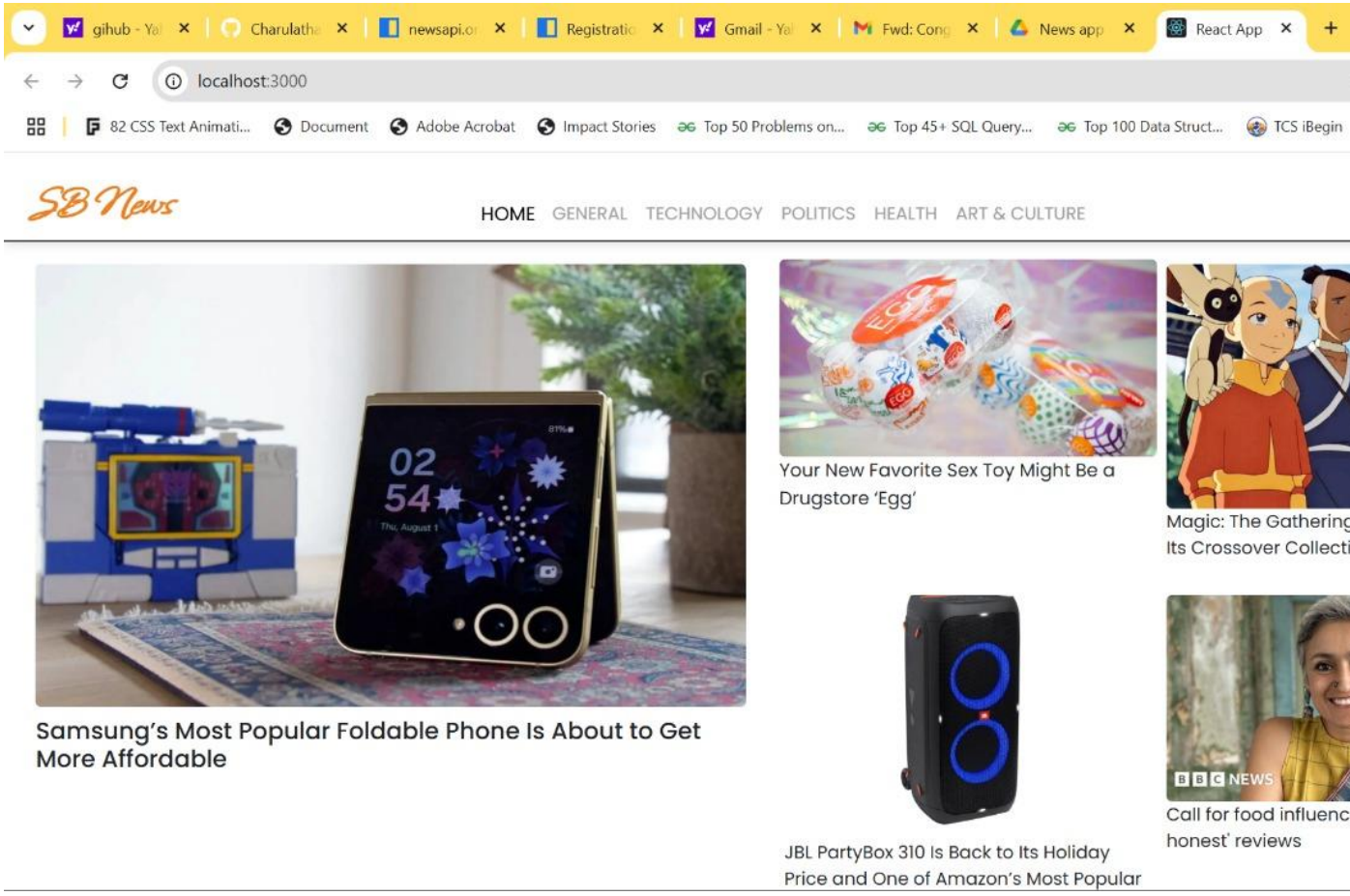
  - Jest + React Testing Library: For testing individual components and ensuring they render correctly.

  - Puppeteer: For browser-based UI testing.

SCREENSHOP AND DEMO




gihub - Ya
Charulatha
newsapi.or
Registratio
Gmail - Ya
Fwd: Cong
News app
React App
localhost:3000
82 CSS Text Animati...
Document
Adobe Acrobat
Impact Stories
Top 50 Problems on...
Top 45+ SQL Query...
Top 100 Data Struct...
TCS iBegin
SB News
HOME
GENERAL
TECHNOLOGY
POLITICS
HEALTH
ART & CULTURE
Amazon Kindle users are going to lose th...
YouTube Music might soon rearrange the p...
The 5 Best Easter Eggs to Find in Kingdo...
Collagen vs. Collagen Peptides: What's t...
AI courses are becoming increasingly pop...
DOGE Has Started Gutting a Key US Technology Agency
Google Is Developing Tec
Deliver Internet Via Light
Politics
The politics behind universal basic income
The power shift US politics needs | Anathea Chino
The Verge hires Tina Ngu
reporter to cover the Tru
administration


gihub - Ya
Charulatha
newsapi.or
Registratic
Gmail - Ya
Fwd: Cong
News app
React App
localhost:3000
82 CSS Text Animati...
Document
Adobe Acrobat
Impact Stories
Top 50 Problems on...
Top 45+ SQL Query...
Top 100 Data Struct...
TCS iBegin
SB News
HOME
GENERAL
TECHNOLOGY
POLITICS
HEALTH
ART & CULTURE
Samsung's Most Popular Foldable Phone Is About to Get More Affordable
Your New Favorite Sex Toy Might Be a Drugstore 'Egg'
Magic: The Gathering
Its Crossover Collecti
JBL PartyBox 310 Is Back to Its Holiday Price and One of Amazon's Most Popular
BBC NEWS
Call for food influenc
honest' reviews