Introduction

- Project Title: Insight Stream: Navigate the News Landscape

- Team Members:

    - [KARTHIKA V]

    - [DHIVYA DHARSHINI]

    - [ANULAKSHMI]

    - [NIVEDHA Y]

Project Overview

Purpose:

The purpose of this project is to develop a modern and user-friendly news application that provides users with a seamless and engaging experience for staying up-to-date with current events. The application aims to deliver relevant and timely news content, leveraging the latest web technologies to ensure a fast, responsive, and intuitive interface.

Goals:

The primary goals of this project are to:

- Provide users with a clean and intuitive interface for browsing news articles

- Deliver relevant and timely news content, categorized by topic and interest

- Ensure a fast and responsive user experience, optimized for various devices and browsers

Key Features:

- News Feed: A curated feed of news articles, categorized by topic and interest

- Search Functionality: A robust search feature that allows users to find specific news articles and topics

- Responsive Design: A responsive design that ensures a seamless user experience across various devices and browsers

- Article Details: Detailed views of news articles, including images, text, and other multimedia content

Architecture

Component Structure:

- Components Folder: Organized into subfolders for each component type (e.g., Header, Footer, NewsCard, SearchBar)
- Major Components:
  - App.js: The main application component
  - Header.js: The header component with navigation and search bar
  - NewsFeed.js: The news feed component that displays a list of news articles
  - NewsCard.js: The individual news article component
  - SearchBar.js: The search bar component
- Component Interactions:
  - App.js renders the Header and NewsFeed components
  - NewsFeed.js renders a list of NewsCard components
  - SearchBar.js interacts with the NewsFeed component to filter news articles

State Management:

- Context API: Used for state management to share data between components
- State Variables:
  - newsArticles: An array of news articles fetched from the API
  - searchQuery: The current search query
  - selectedCategory: The currently selected news category
- State Management Flow:
  - The App component wraps the entire application with the Context provider
  - Components access and update state variables using the useContext hook

Routing:

- React Router: Used for client-side routing
- Route Structure:
  - /: The home page route that displays the news feed
  - /category/:categoryName: The category page route that displays news articles for a specific category
  - /search/:searchQuery: The search results page route that displays news articles matching the search query
- Routing Flow:
  - The App component uses the BrowserRouter to define routes
  - Components use the Link component to navigate between routes

Setup Instructions

Prerequisites:

- Node.js: A JavaScript runtime environment (version 16 or higher)

- npm: The package manager for Node.js (version 8 or higher)

- Git: A version control system for cloning the repository

Installation:

1. Clone the Repository:

   - Open a terminal or command prompt and navigate to the directory where you want to clone the repository.

   - Run the command: git clone https://github.com/[username]/insight-stream.git

   - Replace [username] with the actual GitHub username or organization name.

2. Install Dependencies:

   - Navigate to the cloned repository directory: cd insight-stream

   - Run the command: npm install

   - This will install all the required dependencies listed in the package.json file.

3. Configure Environment Variables:

   - Create a .env file in the root directory of the project.

   - Add the following environment variables:

     - REACT_APP_API_KEY: The API key for the News API

     - REACT_APP_BASE_URL: The base URL for the API requests

   - Replace the placeholder values with your actual API key and base URL.

4. Start the Development Server:

   - Run the command: npm start

- This will start the development server, and you can access the application at http://localhost:3000

Environment Variables:

- REACT_APP_API_KEY: The API key for the News API

- REACT_APP_BASE_URL: The base URL for the API requests

Troubleshooting:

- If you encounter any issues during the installation process, try deleting the node_modules directory and running npm install again.

- If you encounter any issues with the development server, try checking the console logs for errors or restarting the server.

Folder Structure

Client:

The React application is organized into the following folders:

- components: Reusable UI components used throughout the application
  - Header.js: The header component with navigation and search bar
  - Footer.js: The footer component
  - NewsCard.js: The individual news article component
  - SearchBar.js: The search bar component
- pages: Components that represent individual pages or routes
  - Home.js: The home page component that displays the news feed
  - Category.js: The category page component that displays news articles for a specific category
  - SearchResults.js: The search results page component that displays news articles matching the search query
- assets: Static assets used in the application
  - images: Folder for image assets
  - fonts: Folder for font assets
- utils: Utility functions and custom hooks used in the project
  - api.js: A utility function for making API requests
  - helpers.js: A utility function for formatting dates and other helper functions
- context: Context API files for state management
  - NewsContext.js: The context API file for managing news articles state

Utilities:

- Helper Functions:
  - formatDate: A function for formatting dates
  - truncateText: A function for truncating text
- Custom Hooks:
  - useNews: A custom hook for fetching news articles
  - useSearch: A custom hook for handling search queries

Running the Application

To start the frontend server locally and access the InsightStream application, follow these steps:

1. Navigate to the Client Directory: Open a terminal or command prompt and navigate to the client directory of the project: cd client

2. Start the Frontend Server: Run the command npm start to start the development server.

3. Access the Application: Open a web browser and navigate to http://localhost:3000 to access the InsightStream application.

Server Status:

- The development server will be running on http://localhost:3000

- You can view the application in your web browser and interact with it as needed

Component Documentation

Key Components:

- Header Component:
  - Purpose: Displays the navigation bar with search functionality
  - Props: title (string), onSearch (function)
- NewsFeed Component:
  - Purpose: Displays a list of news articles
  - Props: newsArticles (array), onArticleClick (function)
- NewsCard Component:
  - Purpose: Displays individual news article details
  - Props: article (object), onClick (function)
- SearchBar Component:
  - Purpose: Allows users to search for news articles
  - Props: onSearch (function), placeholder (string)

Reusable Components:

- Button Component:
  - Purpose: A reusable button component with customizable styles
  - Props: label (string), onClick (function), variant (string)
- Loader Component:
  - Purpose: A reusable loader component to display loading state
  - Props: size (string), color (string)

Component Configurations:

- NewsCard Component:
  - Displays article title, description, and image
  - Handles click events to navigate to article details
- SearchBar Component:
  - Handles search input and submits search query
  - Displays search results in the NewsFeed component

State Management

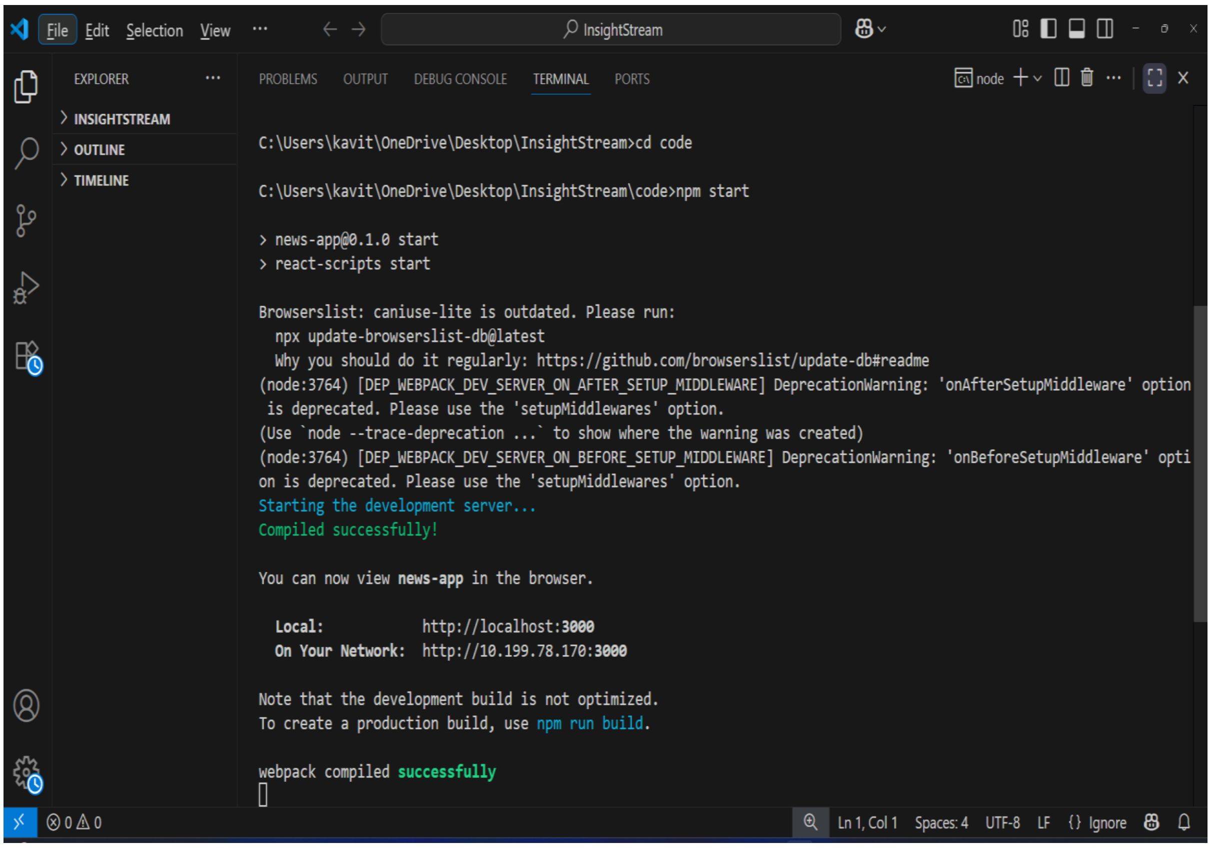
Global State:

- Context API: Used for global state management to share data between components
- State Variables:
  - newsArticles: An array of news articles fetched from the API
  - searchQuery: The current search query
  - selectedCategory: The currently selected news category
- State Flow:
  - The App component wraps the entire application with the Context provider
  - Components access and update state variables using the useContext hook

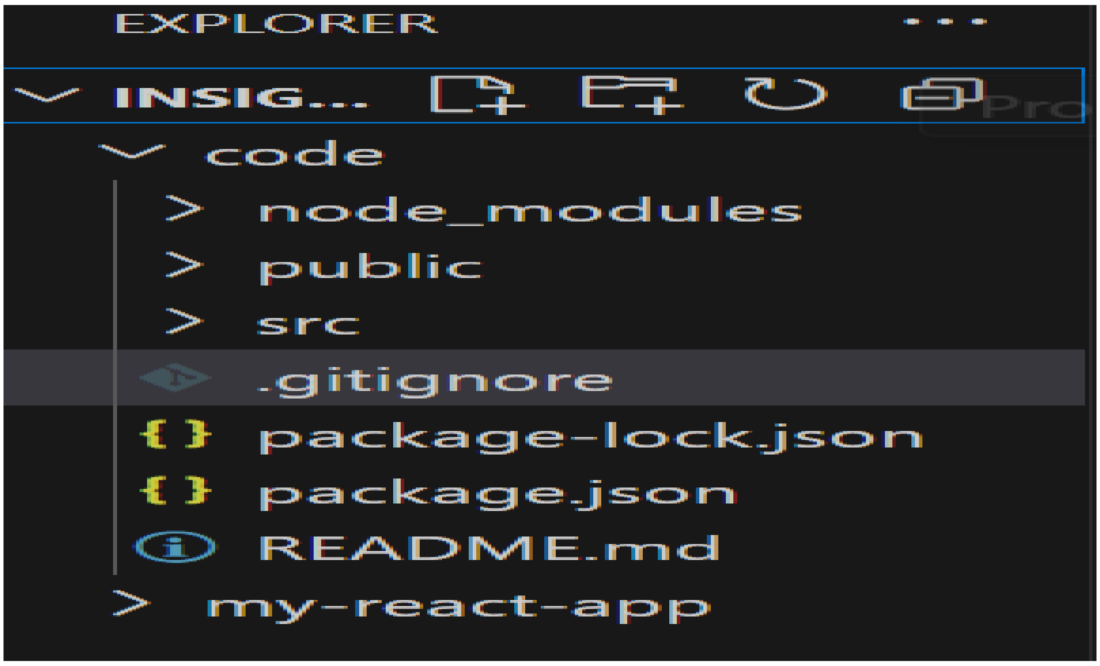
Local State:

- useState Hook: Used for managing local state within components
- Local State Variables:
  - searchInput: The current search input value
  - isLoading: A boolean indicating whether data is being fetched
- Local State Flow:
  - Components use the useState hook to initialize and update local state variables
  - Local state is used to manage component-specific state and behavior

User Interface



```
C:\Users\kavit\OneDrive\Desktop\InsightStream>cd code

C:\Users\kavit\OneDrive\Desktop\InsightStream\code>npm start

> news-app@0.1.0 start
> react-scripts start

Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
(node:3764) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option
 is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:3764) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' opti
on is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view news-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.199.78.170:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



```
INSIG...
  code
    node_modules
    public
    src
    .gitignore
    package-lock.json
    package.json
    README.md
  my-react-app
```

Styling

CSS Frameworks/Libraries:

- Bootstrap: Used as the primary CSS framework for styling and layout

- Styled-Components: Used for component-level styling and theming

Pre-processors:

- Sass: Used for writing modular and reusable CSS code

Theming:

- Custom Design System: Implemented a custom design system to ensure consistency across the application

- Theming: Used Styled-Components' theming feature to allow for easy switching between light and dark modes

- Color Palette: Defined a custom color palette to match the application's branding

Styling Approach:

- Modular CSS: Wrote modular CSS code using Sass to keep styles organized and reusable

- Component-level Styling: Used Styled-Components to style individual components and reduce CSS conflicts

- Responsive Design: Implemented responsive design principles to ensure the application looks great on all devices

Testing

Testing Strategy:

- Unit Testing: Used Jest and React Testing Library to write unit tests for individual components, focusing on functionality and rendering.

- Integration Testing: Used React Testing Library to write integration tests for components that interact with each other, ensuring seamless integration.

- End-to-End Testing: Used Cypress or similar tools to write end-to-end tests, simulating user interactions and verifying application behavior.

Testing Tools:

- Jest: Used as the testing framework for writing unit tests and integration tests.

- React Testing Library: Used to test React components in a more user-centric way.

- Cypress: Used for end-to-end testing, providing a more comprehensive testing experience.

Code Coverage:

- Jest Coverage: Used Jest's built-in coverage tool to ensure adequate test coverage, aiming for 80% coverage or higher.

- Code Review: Regular code reviews were conducted to ensure testing best practices and identify areas for improvement.

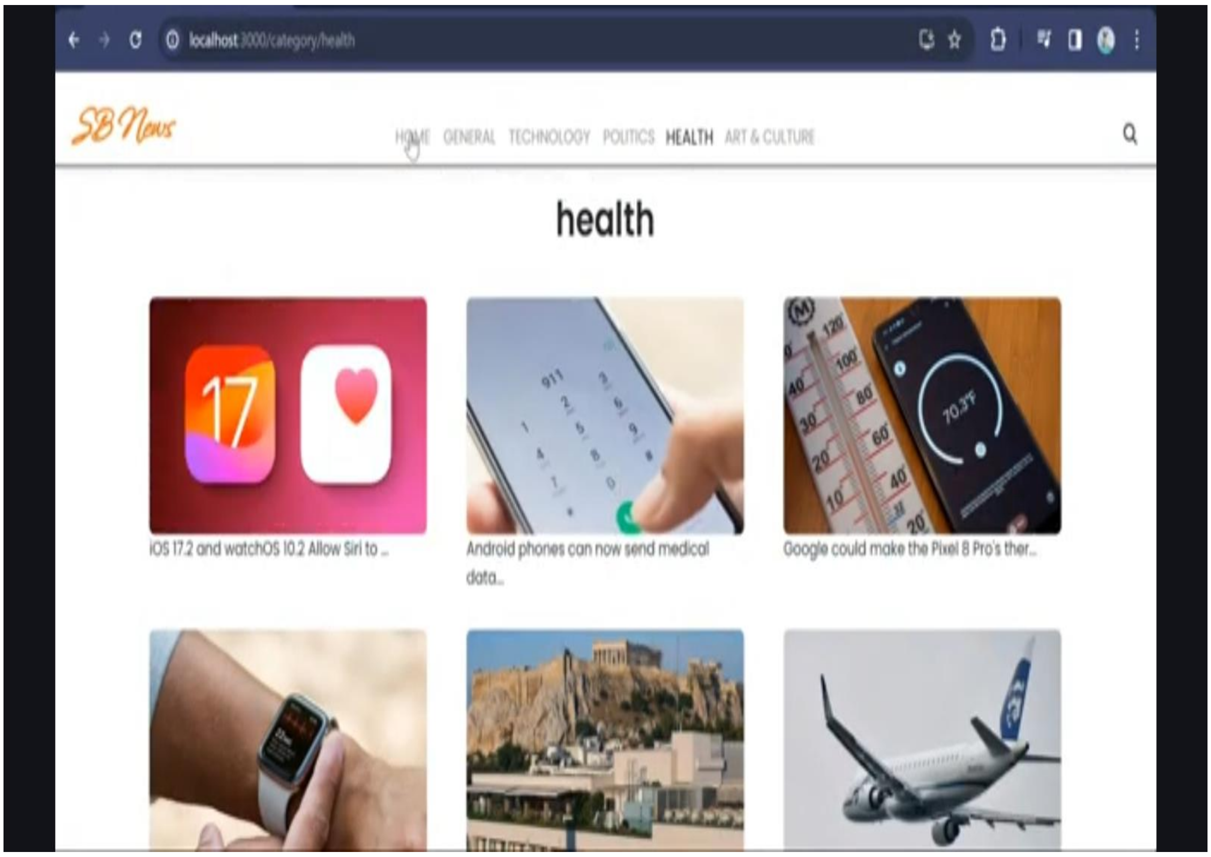
Testing Best Practices:

- Write Tests First: Followed a test-driven development (TDD) approach, writing tests before implementing component functionality.

- Keep Tests Simple: Ensured tests were simple, focused, and easy to maintain.

- Test User Interactions: Tested user interactions and edge cases to ensure the application behaves as expected.

Screenshots

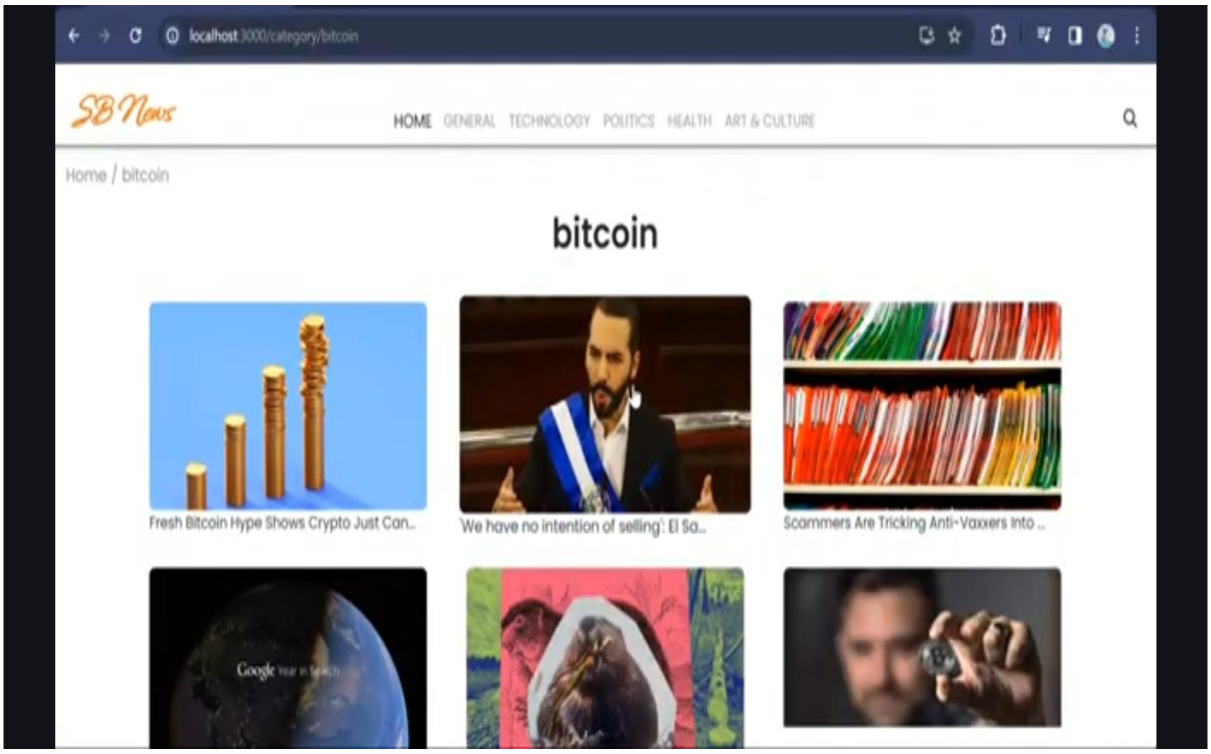

localhost:3000/category/health
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE

health

iOS 17.2 and watchOS 10.2 Allow Siri to ...

Android phones can now send medical data...

Google could make the Pixel 8 Pro's ther...


localhost:3000/category/bitcoin
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
Home / bitcoin

bitcoin

Fresh Bitcoin Hype Shows Crypto Just Can...

'We have no intention of selling': El Sa...

Scammers Are Tricking Anti-Vaxxers Into ...

Known Issues

The following are known bugs or issues that users or developers should be aware of:

- Search Functionality: The search functionality may not work as expected when searching for special characters or punctuation.

- News Article Rendering: News articles with certain formatting or embedded content may not render correctly in the application.

- Responsiveness: The application may not be fully responsive on older devices or browsers, leading to layout issues.

Workarounds:

- For search functionality issues, try searching for keywords without special characters.

- For news article rendering issues, try refreshing the page or checking the article on the original source website.

- For responsiveness issues, try using a modern browser or device.

Fixes in Progress:

- The development team is working to improve the search functionality to handle special characters and punctuation.

- Efforts are being made to improve the rendering of news articles with complex formatting or embedded content.

- The application is being optimized for better responsiveness across devices and browsers.

Future Enhancements

The following are potential future features or improvements:

- New Components:
  - Recommended Articles Section: A section that suggests related articles based on user interests.
  - Author Profile Pages: Pages that showcase author bios, articles, and other relevant information.
- Animations and Interactions:
  - Smooth Scrolling Animations: Animations that enhance the scrolling experience, making it more engaging and smooth.
  - Interactive Visualizations: Interactive visualizations that help users better understand complex data or news stories.
- Enhanced Styling:
  - Dark Mode: A dark mode feature that allows users to switch to a darker color scheme for better readability.
  - Customizable Layout: A feature that allows users to customize the layout and design of the application.
- Performance Optimizations:
  - Lazy Loading: Implementing lazy loading to improve page load times and reduce bandwidth usage.
  - Caching: Implementing caching to improve application performance and reduce server load.
- Accessibility Improvements:
  - Screen Reader Support: Improving screen reader support to make the application more accessible to users with disabilities.
  - Keyboard Navigation: Improving keyboard navigation to make the application more accessible to users who prefer keyboard navigation.