InsightStream: Navigate the News Landscape

1.Introduction

**Team Members**

NAVITHA P (TEAM LEADER)

PAVITHRA J

POORNI SRI M

POOVEETHA S

**2.Project Overview**

**Purpose:**

InsightStream is a web application designed to help users navigate and explore news from multiple sources in an organized and insightful way. The platform aggregates news articles, provides search and filtering functionalities, and offers personalized recommendations based on user preferences.

**Features:**

Aggregation of news articles from various sources

Topic-based filtering and keyword search

User customization and saved preferences

Interactive dashboards with analytics

Responsive design for mobile and desktop

**3.Architecture**

**Component Structure**

App.js – Main entry point that includes routing and layout components

Header – Navigation bar with links to home, topics, and user profile

NewsFeed – Displays aggregated news articles with filtering options

ArticleCard – Individual news article component with metadata

SearchBar – Allows users to search articles by keywords

UserPreferences – Manages user settings and personalized feeds

**State Management**

The application uses Context API for global state management. User preferences, theme settings, and API data are shared across components using context providers.

**Routing**

Implemented using React Router v6 with routes defined for:

/ – Home page with news feed

/topics – Topic selection page

/profile – User preferences and saved articles

**4.Setup Instructions**

**Prerequisites**

Node.js version 18 or above

npm version 9 or above

**Installation**

Clone the repository:

git clone <https://github.com/palanisamynavitha450-tech/Landscape-.git>

Navigate to the client directory:

cd insightstream/client

Install dependencies:

npm install

Create a .env file with necessary API keys:

REACT\_APP\_NEWS\_API\_KEY=your\_api\_key\_here

Start the development server:

npm start

**5.Folder Structure**

client/

├── public/

├── src/

│ ├── components/

│ │ ├── ArticleCard.js

│ │ ├── Header.js

│ │ ├── NewsFeed.js

│ │ └── SearchBar.js

│ ├── context/

│ │ └── AppContext.js

│ ├── pages/

│ │ ├── Home.js

│ │ ├── Profile.js

│ │ └── Topics.js

│ ├── utils/

│ │ └── api.js

│ ├── App.js

│ ├── index.js

├── .env

├── package.json

└── README.md

**Utilities**

api.js – Handles API requests to the news service provider

custom hooks – useFetchArticles for fetching news and useTheme for managing themes

**6.Running the Application**

Run the following command in the client directory:

npm start

This will launch the application locally at <http://localhost:3000/>.

**7.Component Documentation**

**Key Components**

NewsFeed

Displays news articles

Accepts props: filter, searchQuery

ArticleCard

Displays a single article’s title, image, and summary

Props: article object

UserPreferences

Allows users to manage their news topics and themes

Props: none

**Reusable Components**

SearchBar

Handles keyword search input

Props: onSearch callback

Header

Navigation layout used across pages

Props: none

**8.State Management**

**Global State**

Managed using Context API

Holds user preferences, theme data, and API responses

**Local State**

Managed using useState inside components like SearchBar for query input and ArticleCard for likes and comment

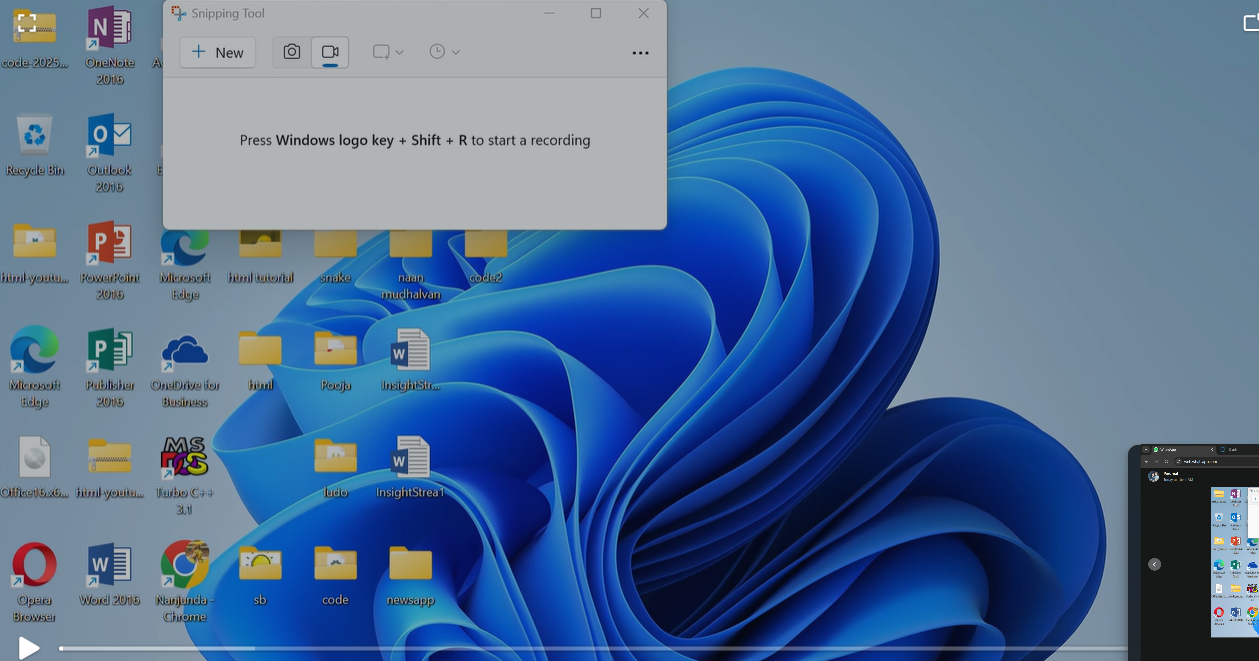
**9.User Interface**

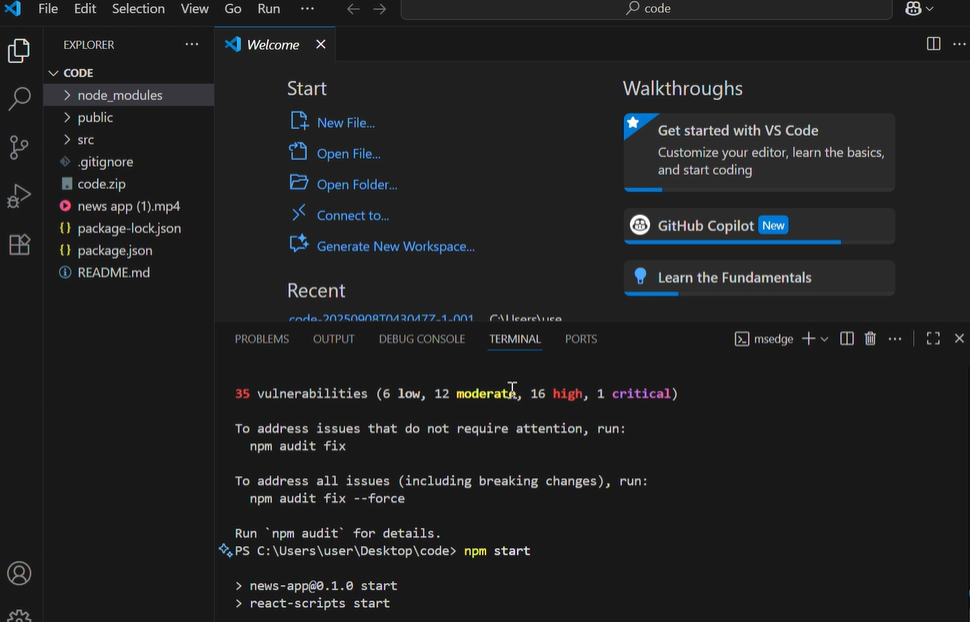
**Screenshots**

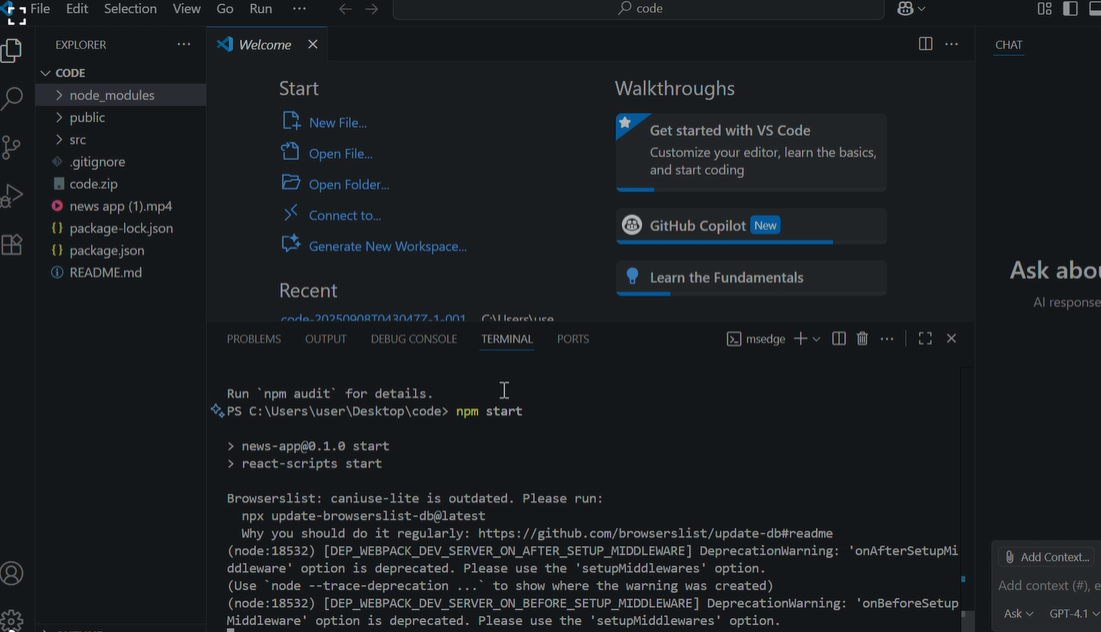
Home Page: Displays news feed with filters

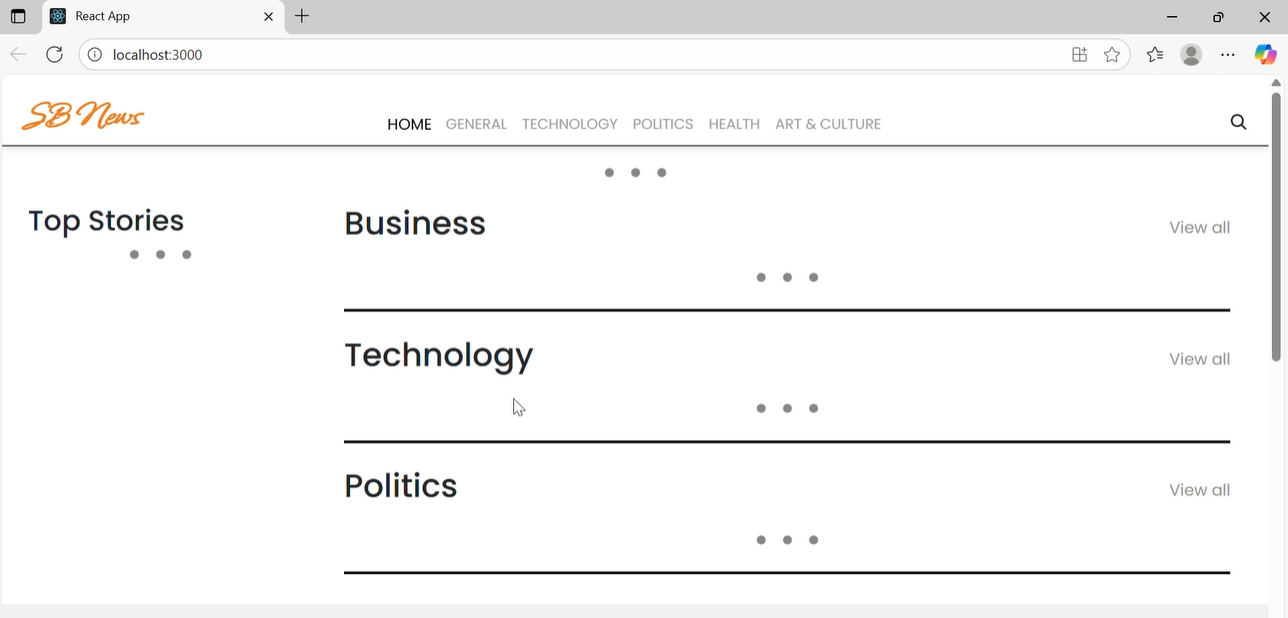
Topics Page: Allows selection of preferred topics

Profile Page: User preferences and saved articles









**10.Styling**

**CSS Frameworks/Libraries**

Uses Tailwind CSS for layout and styling

Custom styles for buttons, cards, and inputs

**theming**

Supports light and dark themes

Uses CSS variables and Context API for theme switching

**11.Testing**

**Testing Strategy**

Unit tests with Jest

Integration tests using React Testing Library

Mock API responses for end-to-end tests

**Code Coverage**

Ensured via Jest’s coverage reports

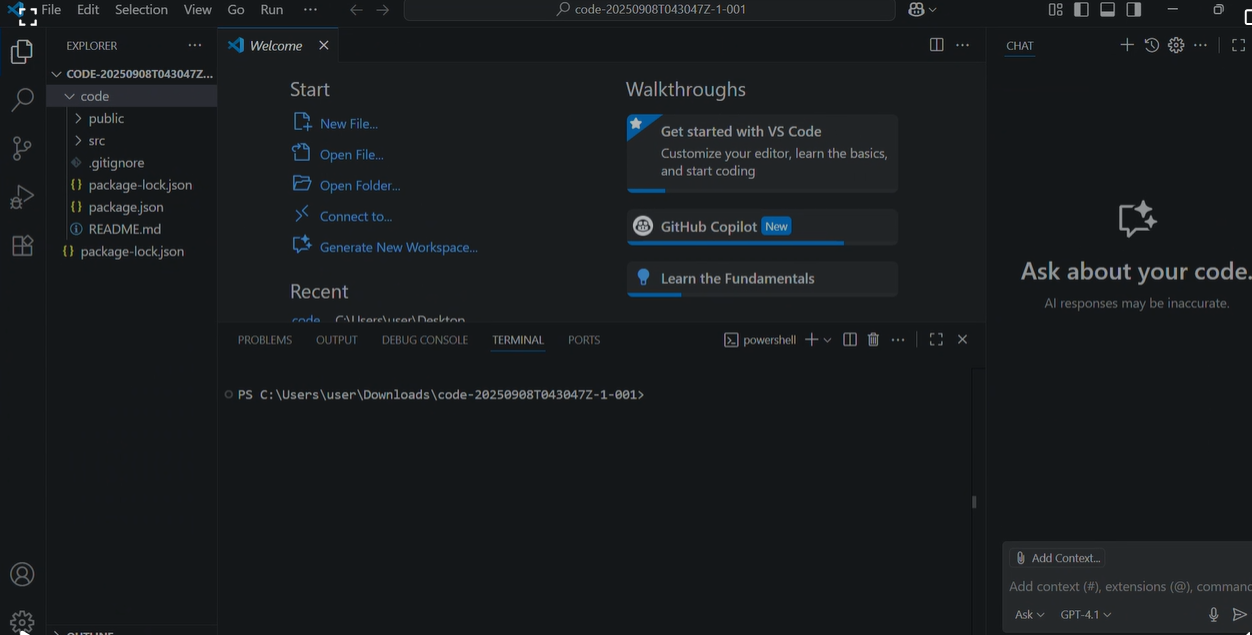
Focus on critical components like NewsFeed and SearchBar

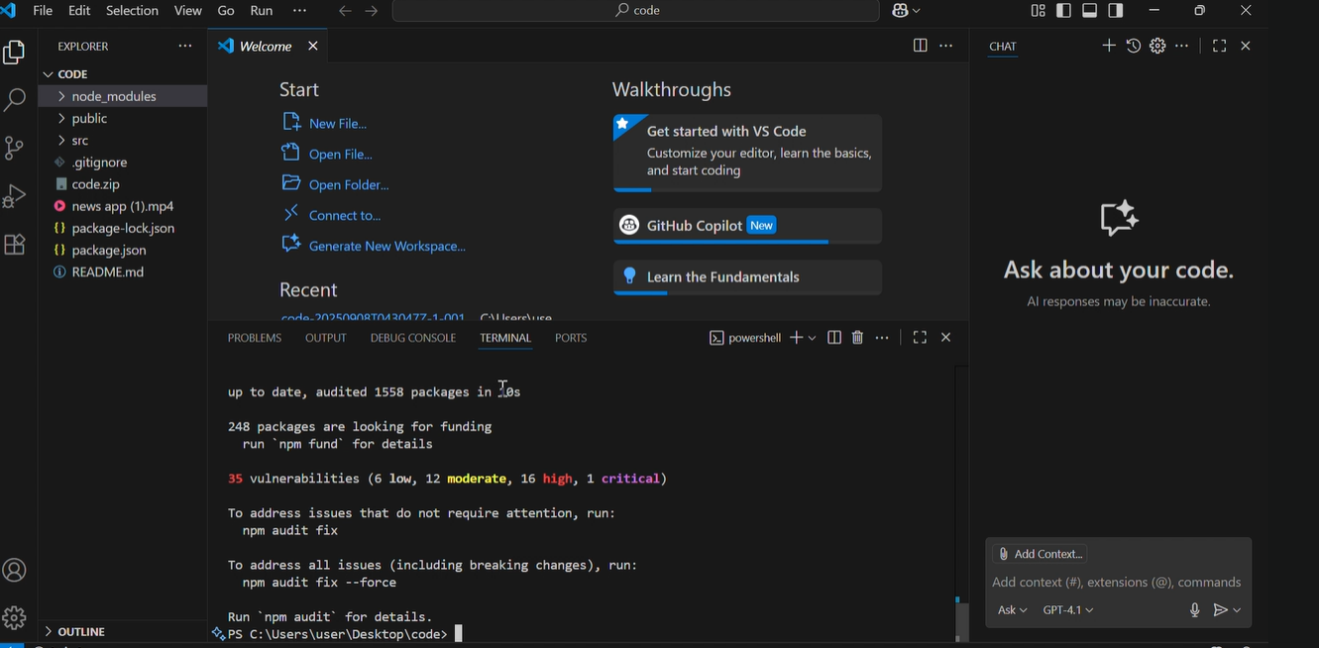
**12.Screenshots or Demo**

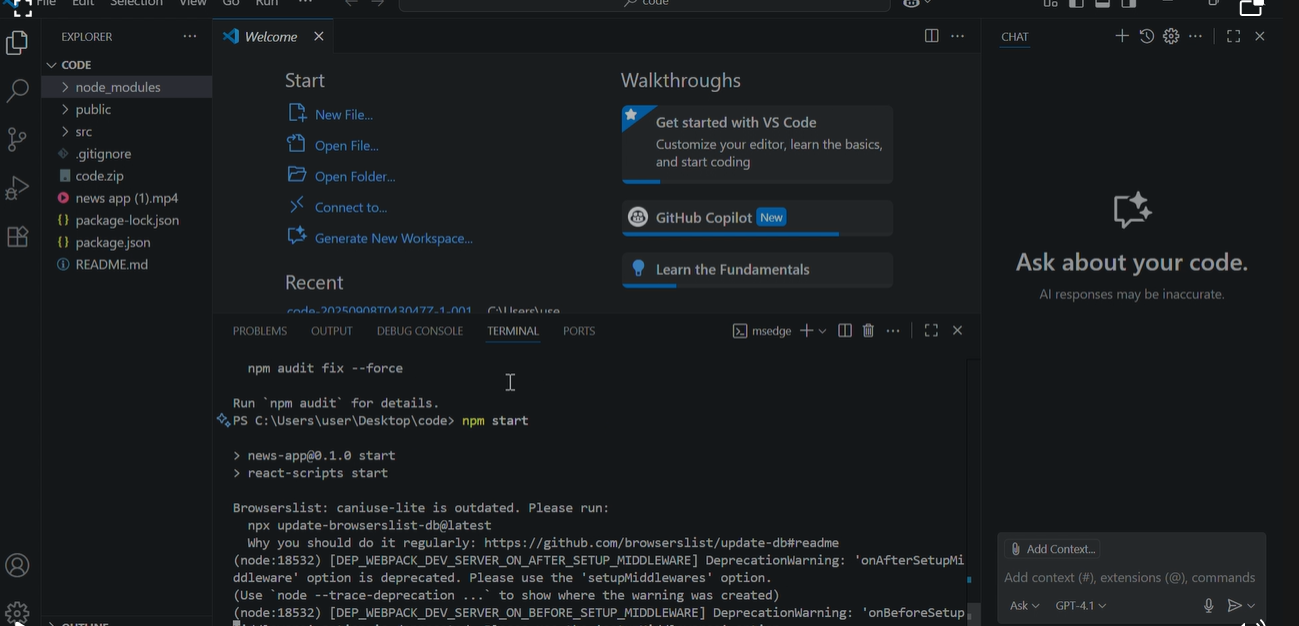
You can view the live demo at:

<https://drive.google.com/file/d/11uBQ3q6BK13SlFLz1ydlbM48QeqWecFL/view?usp=sharing>

<https://drive.google.com/drive/folders/1E9QxUsJfs_W1pTK3qt8lB_yWaYAecCWd?usp=drive_link>







13.Known Issues

Occasionally slow API responses due to third-party rate limits

Mobile responsiveness needs minor tweaks on smaller devices

Some saved preferences may not persist after browser refresh

14.Future Enhancements

Implement advanced filtering by date, location, and sentiment

Add user authentication and article bookmarking

Introduce animations and transitions for smoother interactions

Optimize API calls and caching for better performance