News Aggregator

A PROJECT REPORT

Submitted by

# Gaurav Ojha



Lovely Professional University, Punjab

Name-Gaurav Ojha

Section- K24DJ

Registration Number – 1240419

Roll Number- RK24DJB36 Page-1

INDEX

|  |  |  |
| --- | --- | --- |
| NO. | CHAPTER NAME | PAGE NO. |
| 1. | Title Page | 1 |
| 2. | Introduction and Technology used | 4 |
| 3. | Workflow(flowchart) | 5 |
| 4. | Working of project | 6 |
| 5. | Code snippet (HTML) | 7-8 |
| 6. | Code snippet (CSS) | 8-10 |
| 7. | Code snippet(script.js) | 11-12 |
| 8. | Screenshots of website, webpage | 12-14 |
| 9. | conclusions | 15 |
|  | LINK TO OPEN THE WEBSITE |  |

## 

Page-2

INTRODUCTION

The **News Aggregator** project is a web-based application that dynamically fetches and displays the latest news articles from various topics. Built using HTML, CSS, and JavaScript, the aggregator utilizes the NewsAPI to provide real-time news content to users. This project serves as a one-stop platform where users can explore news across diverse categories like Politics, Sports, Education, Research, Finance, Household, Movies, National, International, Trade, and Geopolitics.

The news aggregator offers an intuitive interface that enables users to filter news by category and search for specific topics. Users can select any category to instantly view related articles, ensuring a seamless, tailored browsing experience. Additionally, the main page presents a default feed with trending articles across all categories, helping users stay informed on a broad range of topics. This project highlights the effective use of APIs in web development, allowing users to access constantly updated content without requiring manual data entry.

With its search functionality, responsive design, and real-time updates, the News Aggregator project demonstrates a robust example of how front-end development skills can be applied to create meaningful, user-centered applications. This project is ideal for users looking to stay current on global happenings in a personalized and streamlined way.

4o

Page-3

# TECHNOLOGY-USED

1. **HTML**:
   * Used to structure the content of the web application, including headers, navigation buttons, and sections for displaying news articles.
2. **CSS**:
   * Applied for styling the web page, making it visually appealing and responsive.
   * Includes custom styles for the layout, buttons, search bar, and news article formatting to ensure a user-friendly interface.
3. **JavaScript**:
   * Used for interactivity and functionality, including handling button clicks, fetching data from APIs, and dynamically updating the page content.
   * Manages API calls, parses data, and controls category filtering and search features.
4. **NewsAPI**:
   * An external API used to fetch real-time news articles from various sources.
   * Allows the app to display up-to-date content for different categories and search queries without manually updating the data.
5. **Fetch API**:
   * A JavaScript-based method for making asynchronous HTTP requests to retrieve news data from NewsAPI.
   * Ensures that news content is updated in real time by dynamically loading data from the server.
6. **Visual Studio Code (VS Code)**:
   * The development environment used for coding, testing, and running the project with Live Server.
   * Provides syntax highlighting, debugging, and an integrated terminal to streamline development.

These technologies come together to create a responsive, interactive, and real-time news aggregator that enhances the user’s ability to stay informed across multiple topics.

Page-4

WORKFLOW

User Interface Display:

The project starts by loading index.html, which provides the initial structure and layout of the page, including the header, search bar, category buttons, and an empty container where news articles will be displayed.

style.css is applied to make the interface user-friendly and responsive.

**Default News Loading on Page Load**:

When the page is first loaded, the DOMContentLoaded event triggers the fetchNews('general') function in script.js.

This function sends a request to the NewsAPI for general news articles, and the results are displayed in the main content area. This default feed provides a general overview of trending topics to engage users immediately.

**Category Selection for News Filtering**:

The navigation bar offers buttons for various categories, like Politics, Sports, Education, and others.

When a user clicks on a category button, it triggers the filterNews(category) function, which calls fetchNews() with the selected category.

The fetchNews() function dynamically loads articles that match the selected category, replacing the previous content in the news container.

**Search Functionality**:

Users can type keywords into the search bar to find articles on specific topics.

The searchNews() function captures the user input, sends a request to the NewsAPI with the query, and retrieves articles that match the search term.

If articles are found, they’re displayed in the news container; if not, a "No articles found" message appears to inform the user.

**Data Fetching Using Fetch API**:

Page-5

For both category selection and search functions, the Fetch API is used to send asynchronous requests to the NewsAPI.

JavaScript’s async and await syntax handles these asynchronous calls, ensuring a smooth and responsive experience for users.

Data from the API response is parsed and passed to the displayArticles() function to format and insert it into the HTML.

**Displaying Articles in the HTML Container**:

The displayArticles() function processes each article by creating individual elements with the title, description, and a "Read more" link to the full article.

These article elements are appended to the newsContainer section, dynamically updating the page based on user interactions.

**Error Handling and User Feedback**:

Error handling is integrated into the data-fetching functions to catch and display messages when the NewsAPI fails to respond or if the request exceeds the API limits.

The user is informed of the loading state, search results, or any errors directly in the newsContainer, maintaining a clear communication flow with the user.

**Continuous Updates and Real-Time News**:

With the NewsAPI providing fresh content, users always have access to the latest news articles in any category or search query.

The API-driven design ensures that content is up-to-date without requiring manual data management.

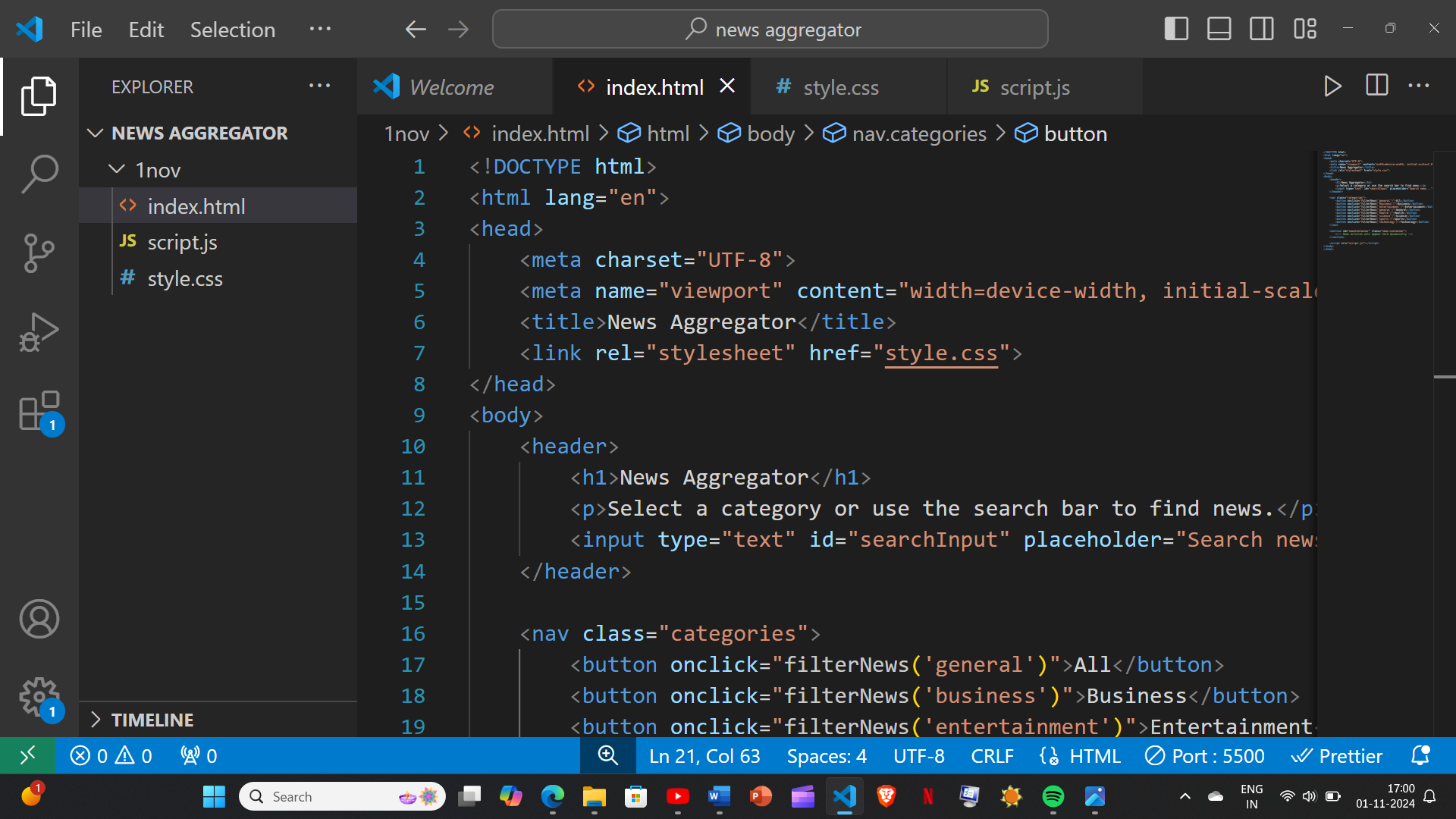
**Summary**

The project’s workflow provides a seamless experience by dynamically fetching and displaying news articles based on user input, category selection, or search queries. This workflow enables users to explore a variety of news topics efficiently and stay updated with real-time news content.

page-6

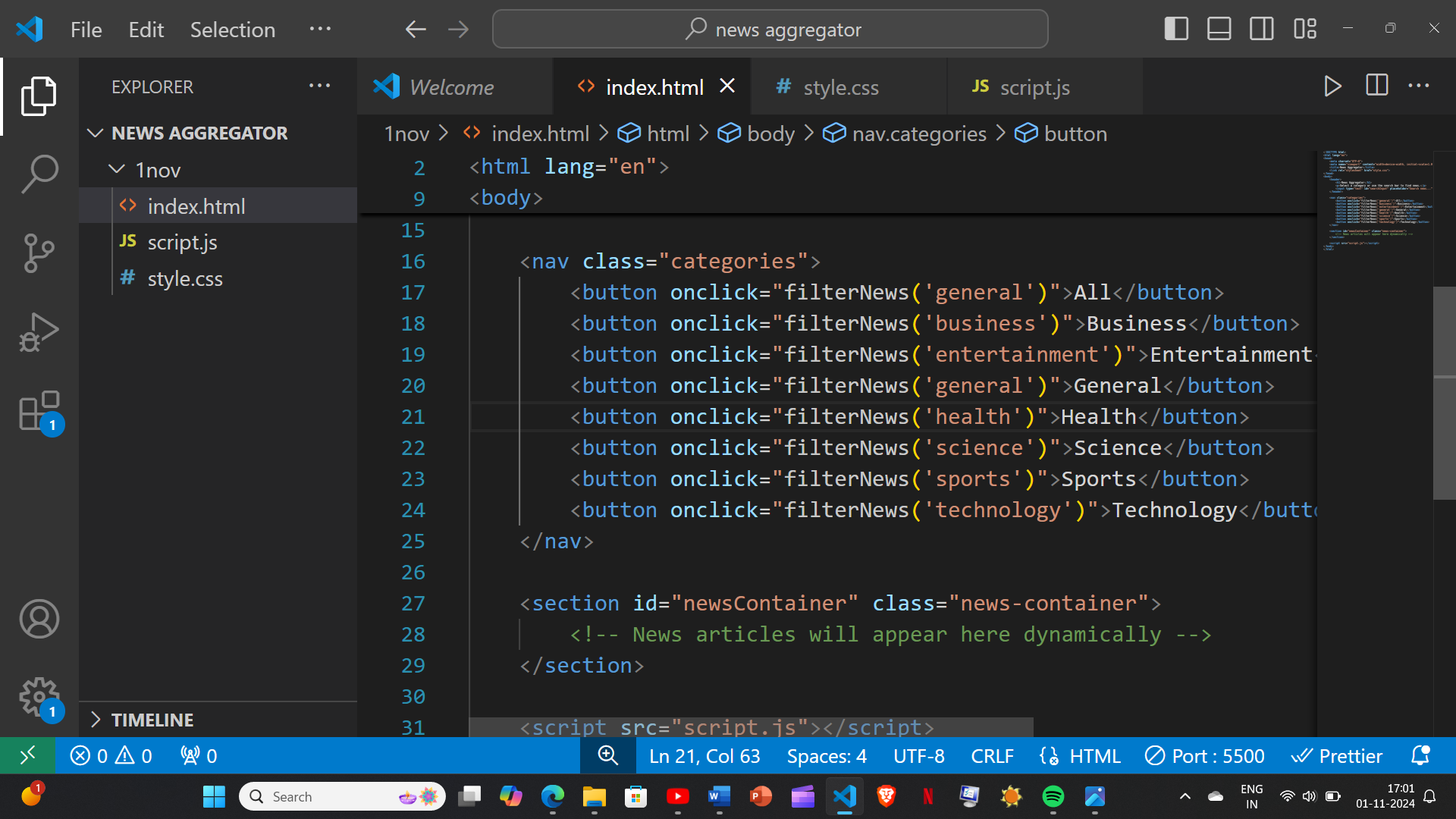
# 

## (INDEX.HTML)

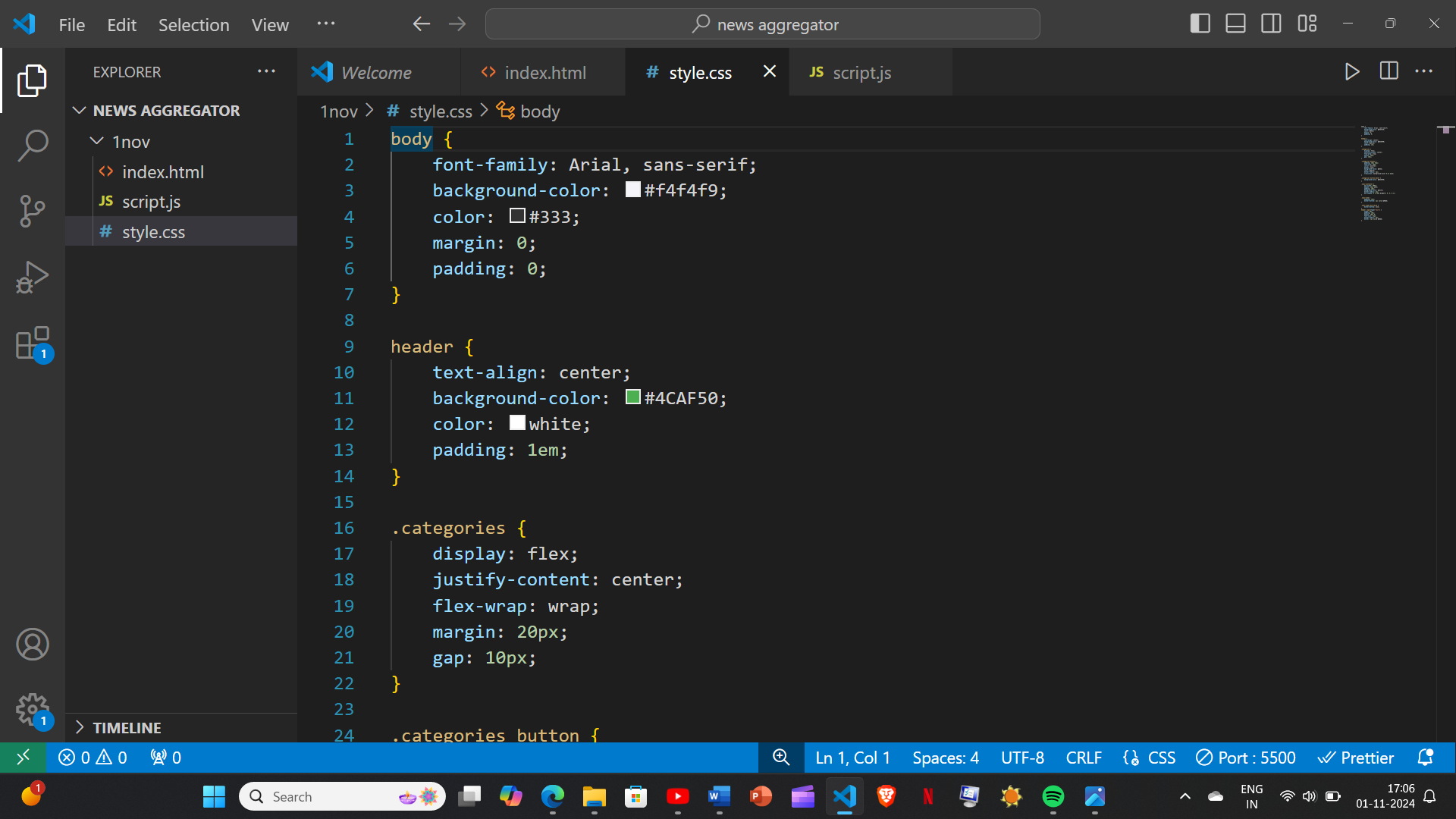


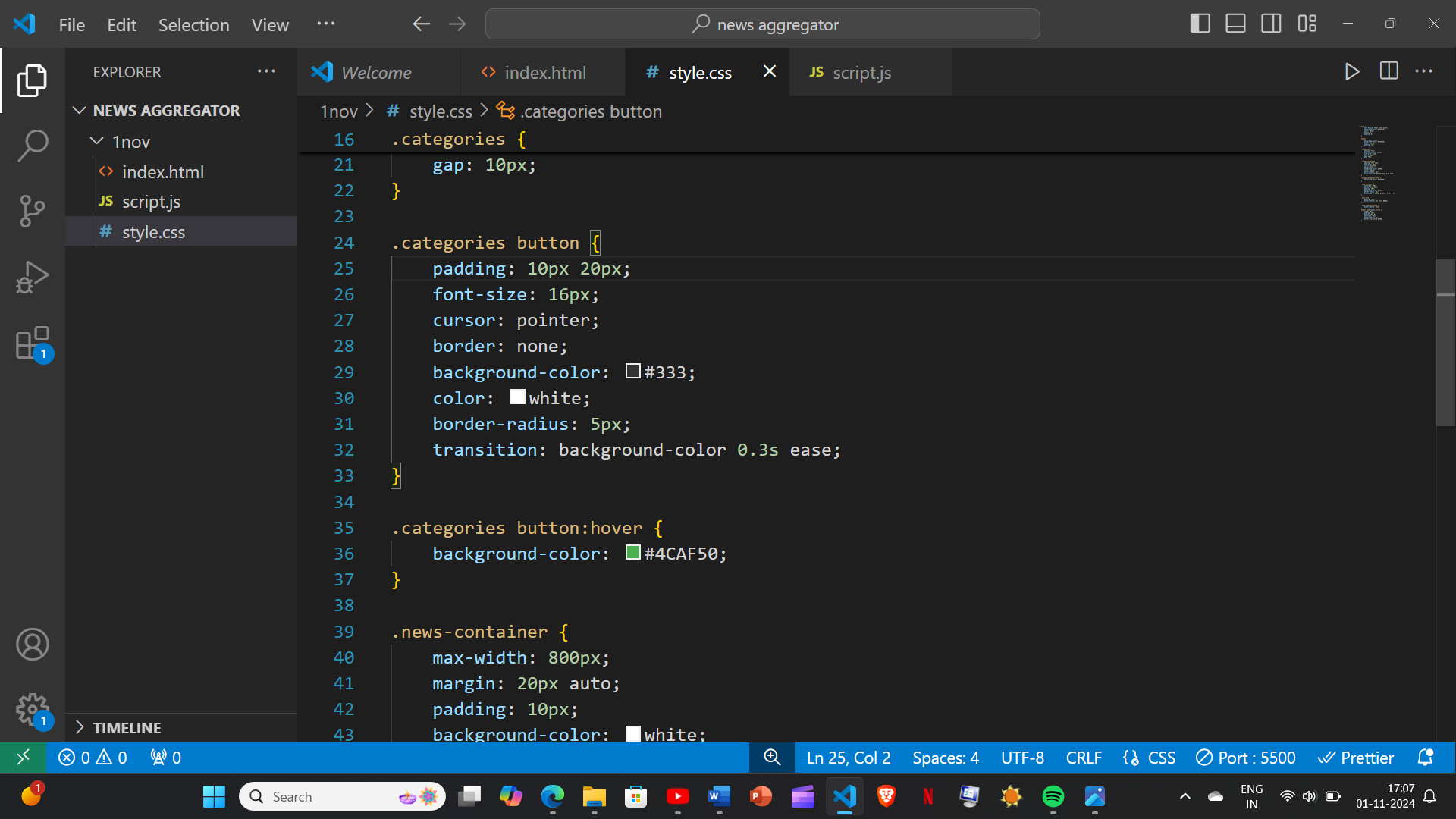
Page-7

Page-8

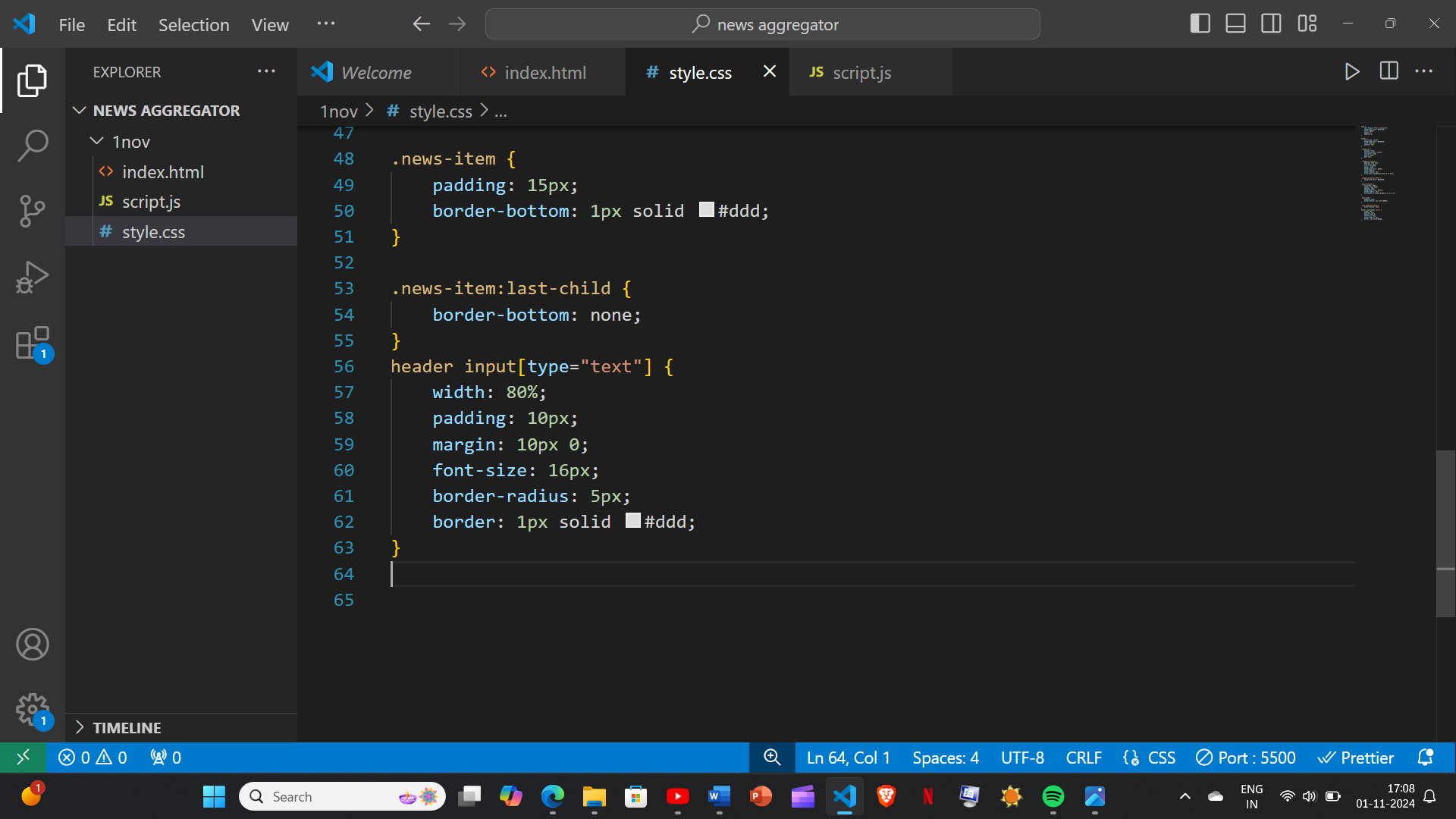


## Style.css



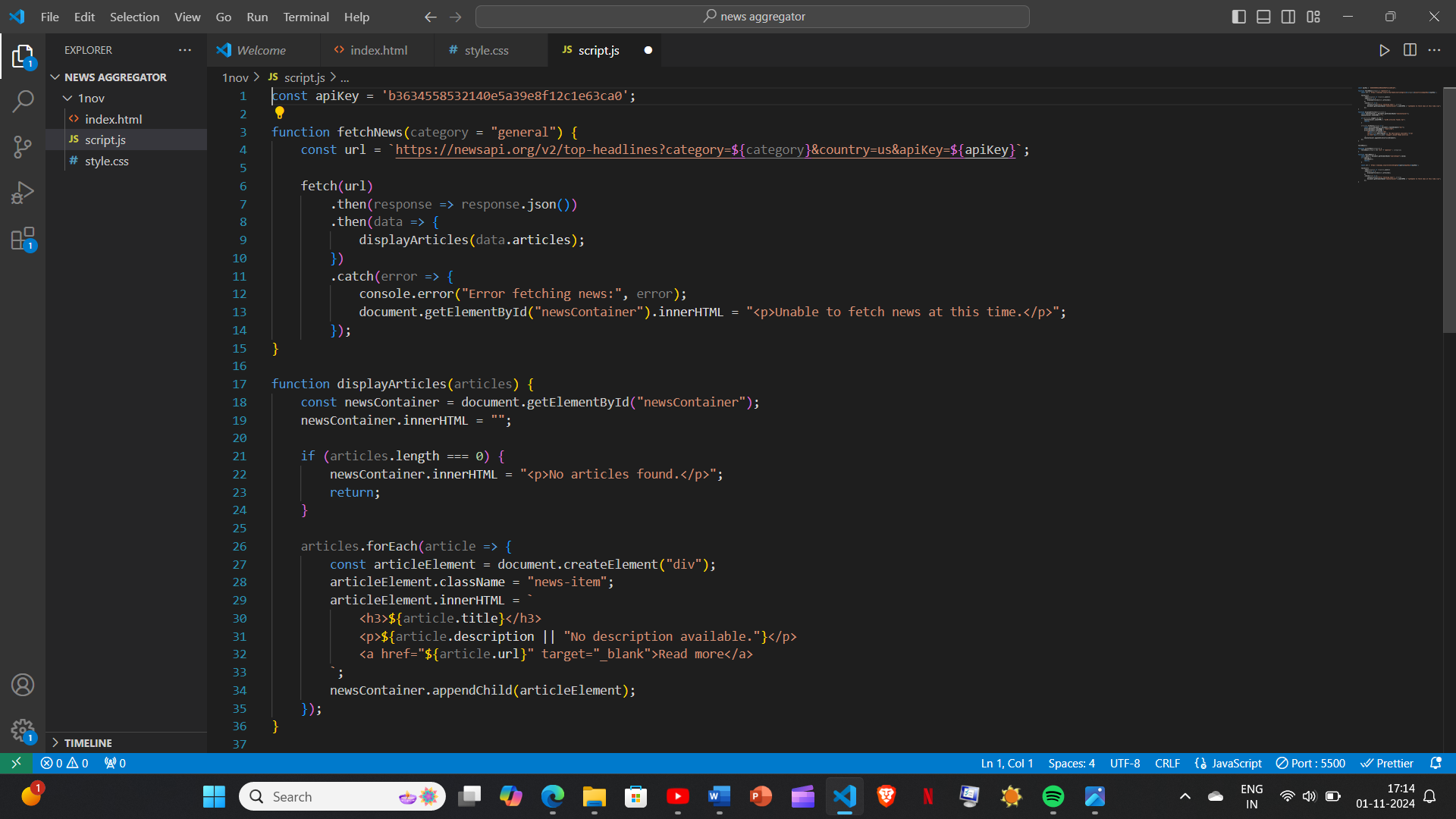


Page-9

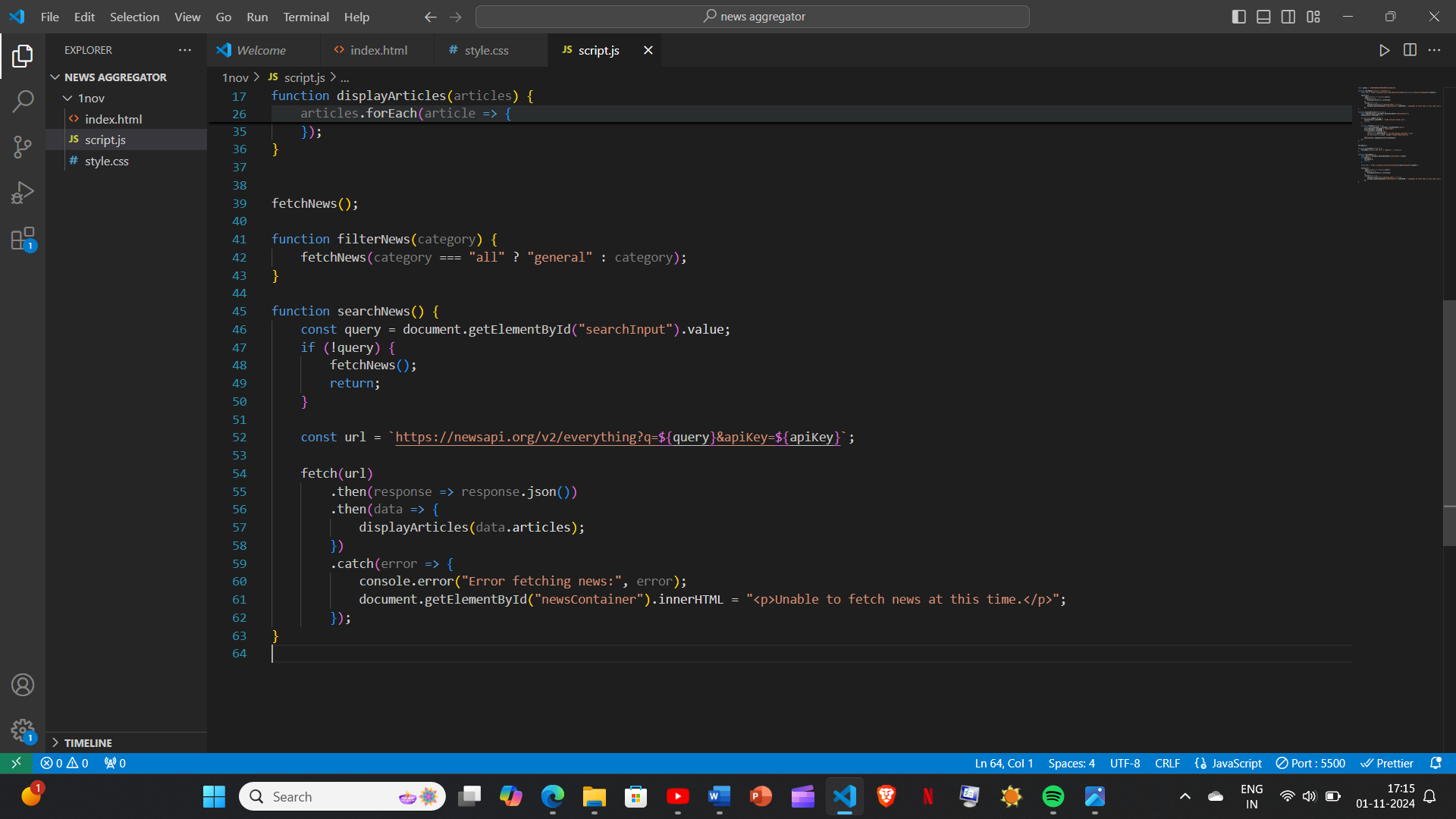


Page-10

## Script.js

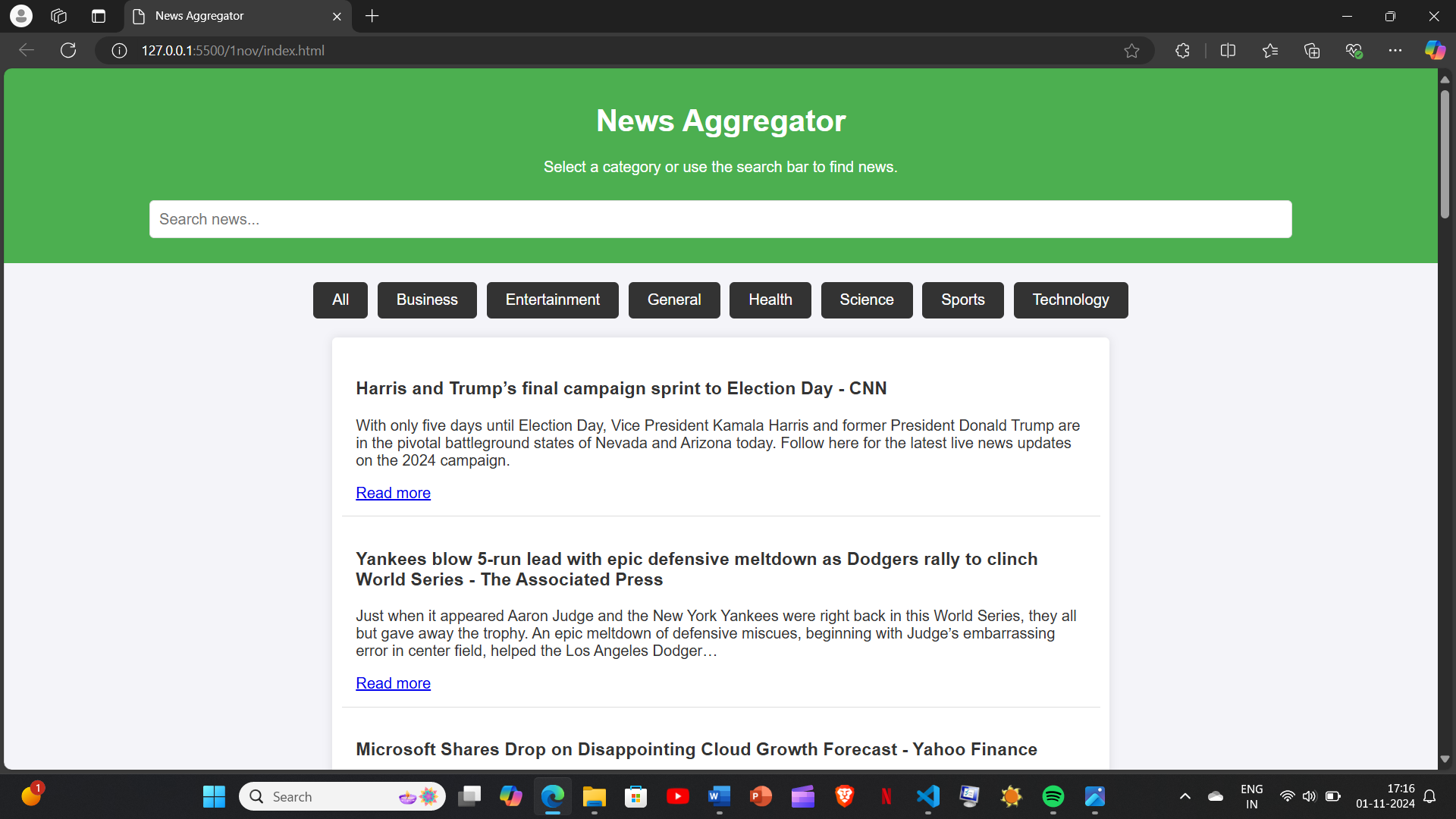


Page-11

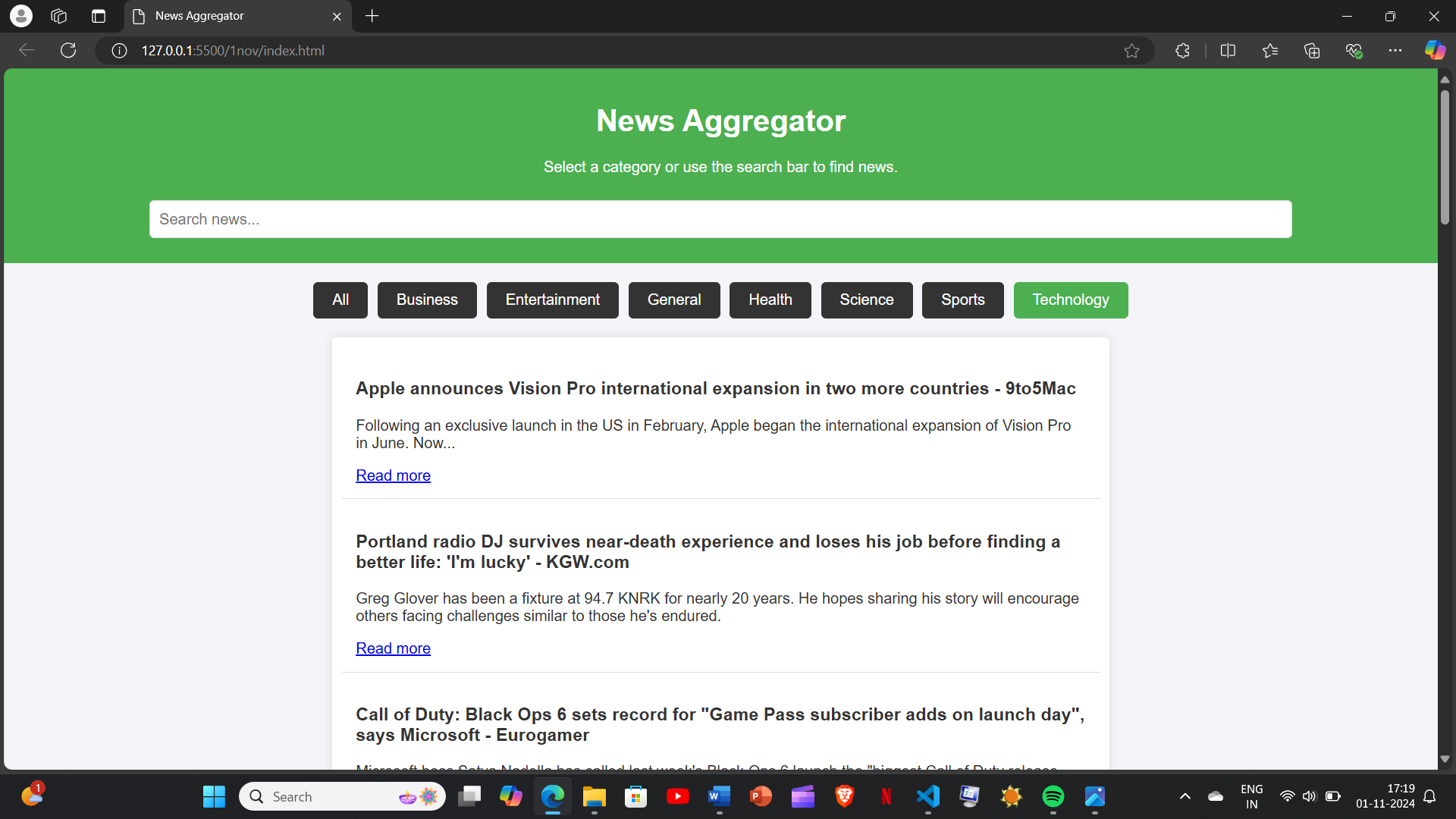


Page-12

SCREENSHOT OF WEBPAGE



Page-13



Page-14

Conclusion

The News Aggregator project is a practical application that effectively demonstrates the integration of front-end technologies—HTML, CSS, and JavaScript—with external data sources through an API. By fetching real-time news from NewsAPI, the project provides a dynamic and interactive platform for users to stay updated on a wide array of topics.

Through its category selection, search functionality, and user-friendly interface, the aggregator allows users to personalize their news consumption experience effortlessly. This project showcases the power of APIs in enriching web applications, allowing for up-to-date content without constant maintenance. Additionally, it demonstrates core development skills, such as asynchronous programming, error handling, and responsive design.

In conclusion, the News Aggregator is a valuable, functional project that not only meets the needs of users seeking timely news updates but also serves as an excellent example of how web development tools and technologies can be used to build engaging, content-driven applications. Page-15

ADDRESS OF THE WEBPAGE:

Click on the link below to visit the website:

THANK YOU