

COMPLEX ENGINEERING PROBLEM 1

**SUBMITTED BY**

* **CS-21106**

WEB SERVICES USING REST API

#### Submitted to : Dr. Muhammad Ismail

#### Date : 27th Nov, 2024

1. **INTRODUCTION**

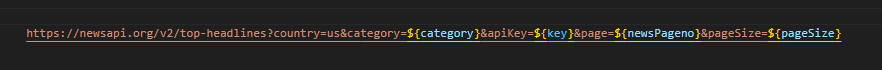
The objective of this project is to exemplify the employment of RESTful APIs in a contemporary web application, bringing to the fore integration of a third party News API with a React based front end. This project is based on the service “https://newsapi.org”, which allows to obtain the most important news based on the selected category of news and pagination. In this perspective, it correlates with the aims and objectives of web services by offering practical knowledge in enhancing the use of APIs, front end development and real-time content display.

1. **OBJECTIVE**

The primary objective is to create an interactive and visually appealing news platform that allows users to:

* View top news headlines dynamically.
* Filter news based on categories such as Business, Entertainment, Health, Sports, etc.
* Navigate through multiple pages using pagination.
* Present data in a user-friendly and responsive interface.

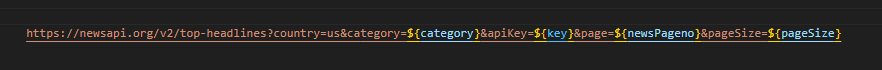
1. **TECHNICAL STACK**

* **Frontend Framework:** React.js
* **API Service :** 
* **Styling:** Tailwind CSS for responsive and modern design.
* **State Management:** React Hooks for managing API data, category selection, and pagination.

1. **PROJECT WORKFLOW**

**4.1 INCORPORATION OF API**

This application integrates news content by means of accessing the News API via the following endpoint:



This addresses the UI as the frontend of the application takes into account dynamic aspects such as category, newsPageno and pageSize making it economical and extensible.

These parameters can be fetched through fetch news function which takes newsPageno as argument and category, pagesize as props passed to news component , here api key (the key assigned to particular user) is coming from .env.local file so that during production this key cannot be seen in the console or network tab.

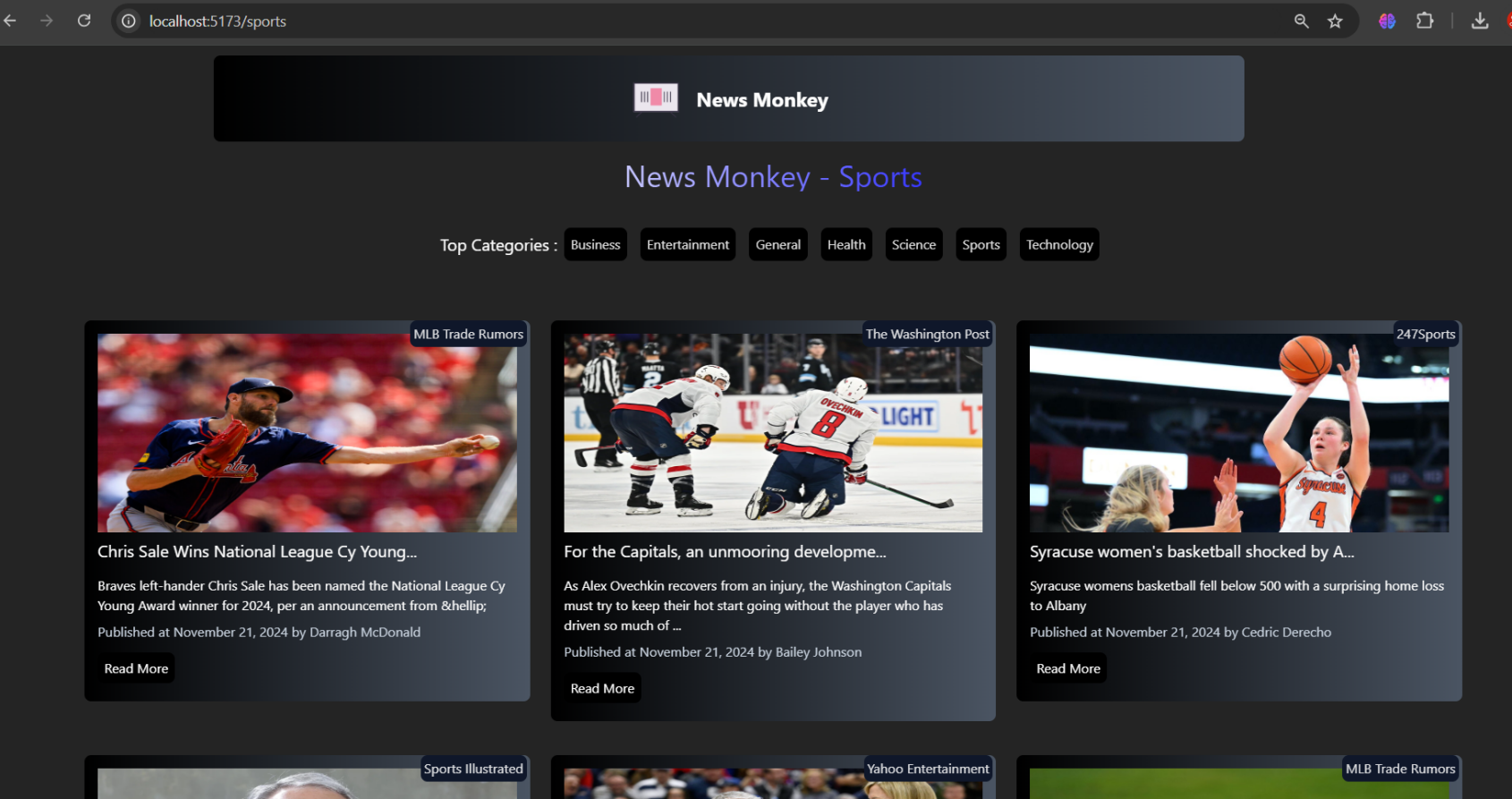


**4.2 FRONTEND INTERFACE**

The user interface features a clean and professional design with dark- themed styling, ensuring readability and visual appeal.

News articles are displayed in cards, each containing:

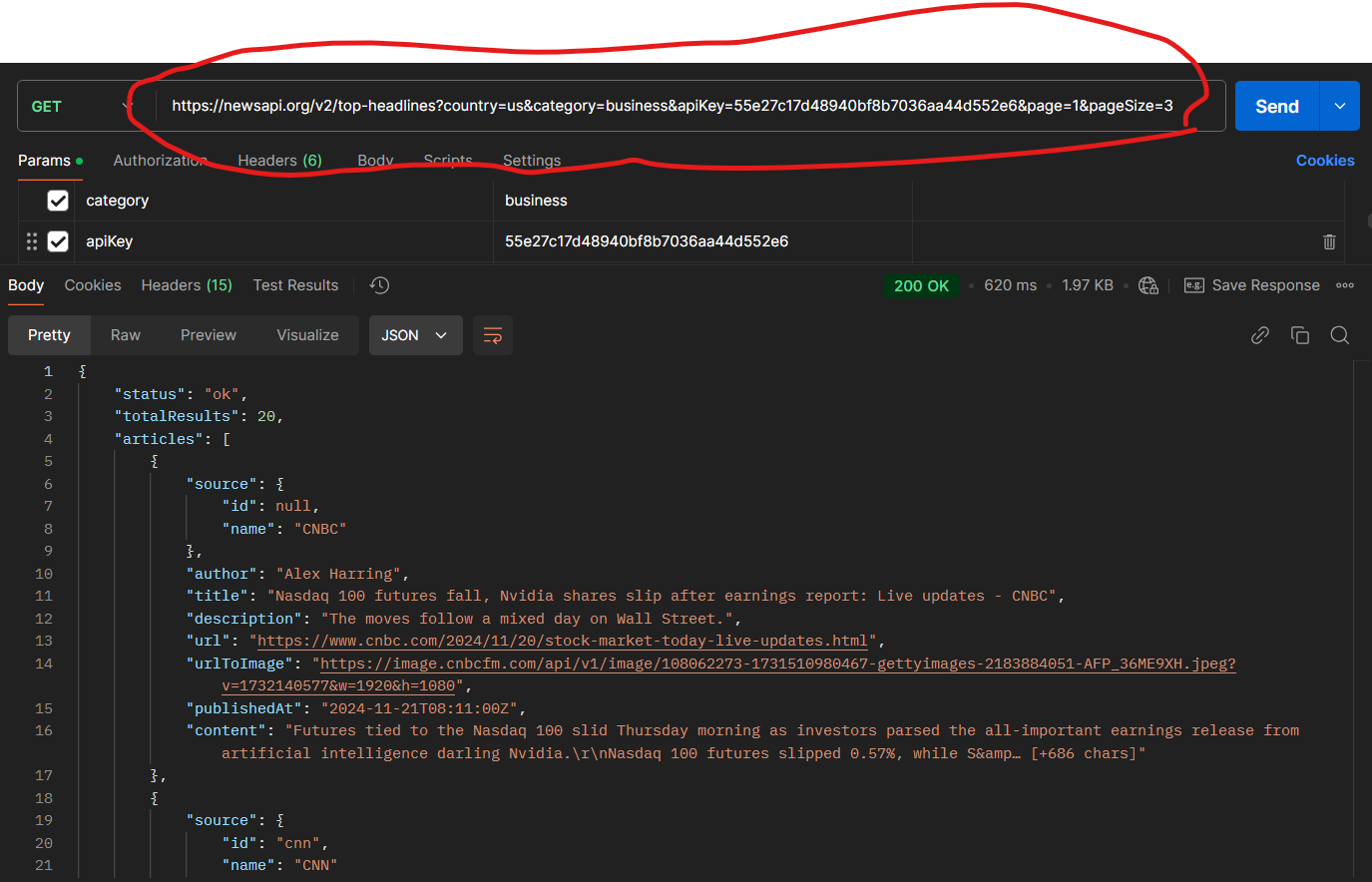
* Article image
* Title and description
* Source Information
* News belonging to specific category are filtered
* First ten news are fetched from the api at the start, as the user keeps scrolling new fetch of Api takes place so that storage is utilized efficiently of web browser.
* The Api calls on scrolling takes place by changing the page number in api discussed above , on the the increase of pageNo in the Api new page is fetched to showcase on the frontend so that the previous news and the newly fetched news are mapped together
* The pagesize in the Api determines the no of news articles to be displayed, the screenshot below gives better understanding of working of Rest Api:



Here the Sports Category is asked by the user so the sports related news is fetched from the api and the articles are giving news details.

**4.3 REST API FETCHING:**

The question arises here what details are the rest api giving, in order to understand that, let’s go through the json data given by rest apis.



Here data related to business category is asked through the Api and pageSize given is 3 means the user wants 3 articles of sports category news. Here on changing the page in the Api will display news of that particular page.

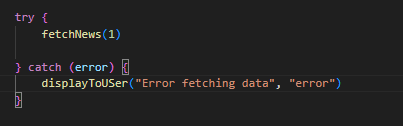
1. **KEY FEATURES DEMONSTRATED**

**5.1 DYNAMIC DATA RENDERING**

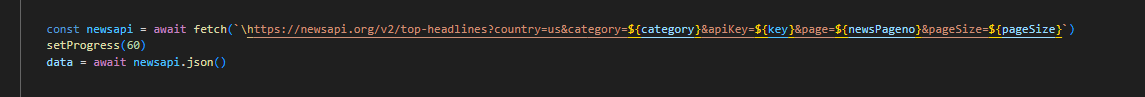
As seen from the previous example the app dynamically updates the news content based on user-selected categories, current page and page size which shows the pivelege of using rest api.

**5.2 ERROR HANDLING**

Basic error handling ensures users are notified if the API fails to fetch data. In the application it is achieved by using a try catch block in order to catch any error thrown by API.



In this way error handling is achieved quiet easily. The message here on error is given in development environment in order to test the working of REST API.



**5.3 ASYNCHRONOUS OPERATION**

The API is properly called using async/ await function which ensures proper handling of API response, since data will take some time in order to come through the API, making the API call using async/await will hold the further operation which are running in parallel, now the operation will run in sequence which will lead the code execution to move to next line only when the data from the API is properly fetched.

1. SUMMARY

This dissertation presents how the News API has been integrated within a React-based web application to create a very responsive, and up-to-date news site. The News API includes top headlines and stories divided into sections among them; Business, Sports and Entertainment. The application parameters – category, pageNo and pageSize - make it possible to download contents on demand making it fast growing and good for content delivery network. Delegates (API calls) are made to work asynchronously with async/wait, thus making it possible to get data and control the user at the same time. The exposed API key used in production is protected by keeping it inside .env.local file.

For the frontend part, a dark-themed responsive user interface built on Tailwind CSS has been used. The News articles are presented in card based layouts with images, titles, description, source etc. Where there is a lot of Information content, Paginations and infinite scrolling are used to ease the loading of data and improve the user experience. A safety net of error handling is in place to provide the desired level of functionality by intercepting and presenting errors arising from the functionality of the API. This project considers REST APIs to be important as they overcome the issues of latency, hardness and fixed expansion on increased usage, which are important in any web application today.