|  |
| --- |
| **API Data Fetching App**  *“A React application that fetches and displays posts from a public API using useEffect and useState.”*  **Developed By: Gaurvi Paneri**  14 August, 2025.  BrainyBeam Placement Task |

# **Table of Contents**

1. **Introduction** *Overview and purpose of the Auth Routing*
2. **Objective** *What app aims to achieve and why*
3. **Features** *Description of key functionalities*
4. **Technology Stack** *Tools and libraries used in the project*
5. **Application Architecture** *Explanation of frontend structure and data flow*
6. **Implementation Details** *Key code logic, components, and state management*
7. **Usage Instructions***How to install dependencies, use of this application*
8. **Screenshots** *Screenshot displaying the Output of the project*
9. **References** *Resources and documentation referred to during development*
10. **Conclusion** *Summary of the project and learnings.*

## **1. Introduction**

This project is a simple React application that fetches and displays data from a public API. It demonstrates how to use React hooks such as useEffect and useState for making API requests, handling asynchronous data, and rendering the fetched content in the browser.

## **2. Objective**

To build a React application that retrieves data from a public API and presents it in a clean and readable format, while learning about API integration in frontend applications.

## **3. Features**

* Fetches data from the public API **(**[**https://jsonplaceholder.typicode.com/posts**](https://jsonplaceholder.typicode.com/posts)**)**.
* Displays all posts in sequential order with their ID, title, and body.
* Shows a loading message until the data is fully retrieved.

## **4. Technologies Used**

* **React with Vite:** For fast development and easy setup
* **API:** JSONPlaceholder (sample public API)

## 

## 

## 

## **5. Application Architecture**

## **Folder Structure**

src/

│── main.jsx # App entry point

│── App.jsx # Routes setup

│── components/

│ └── PostData.jsx # Fetches and displays data from the public API

## **6. Implementation / Working**

1. When the PostData component mounts, the useEffect hook sends a GET request to the public API endpoint.
2. While fetching, a **"Loading..."** message is displayed.
3. Once the API returns the data, it is stored in the posts state using useState.
4. The posts are then mapped and rendered in individual styled <div> elements containing the ID, title, and body.
5. If there is an error during fetching, it is logged to the console.

**7.Usage Instructions**

**Running the App Locally**

**●Prerequisites:**

* Make sure you have Node.js (v16 or higher) installed on your system.
* A modern web browser (Chrome, Firefox, Edge, etc.).

**●Installation:**

Clone the project repository to your local machine:

git clone React\_Task\_4\_GaurviPaneri

Navigate to the project folder:

cd React\_Task\_4\_GaurviPaneri/code/API\_Code

**Install dependencies:**

npm install

**●Start the Development Server:**

Run the app locally using Vite:

**npm run dev**

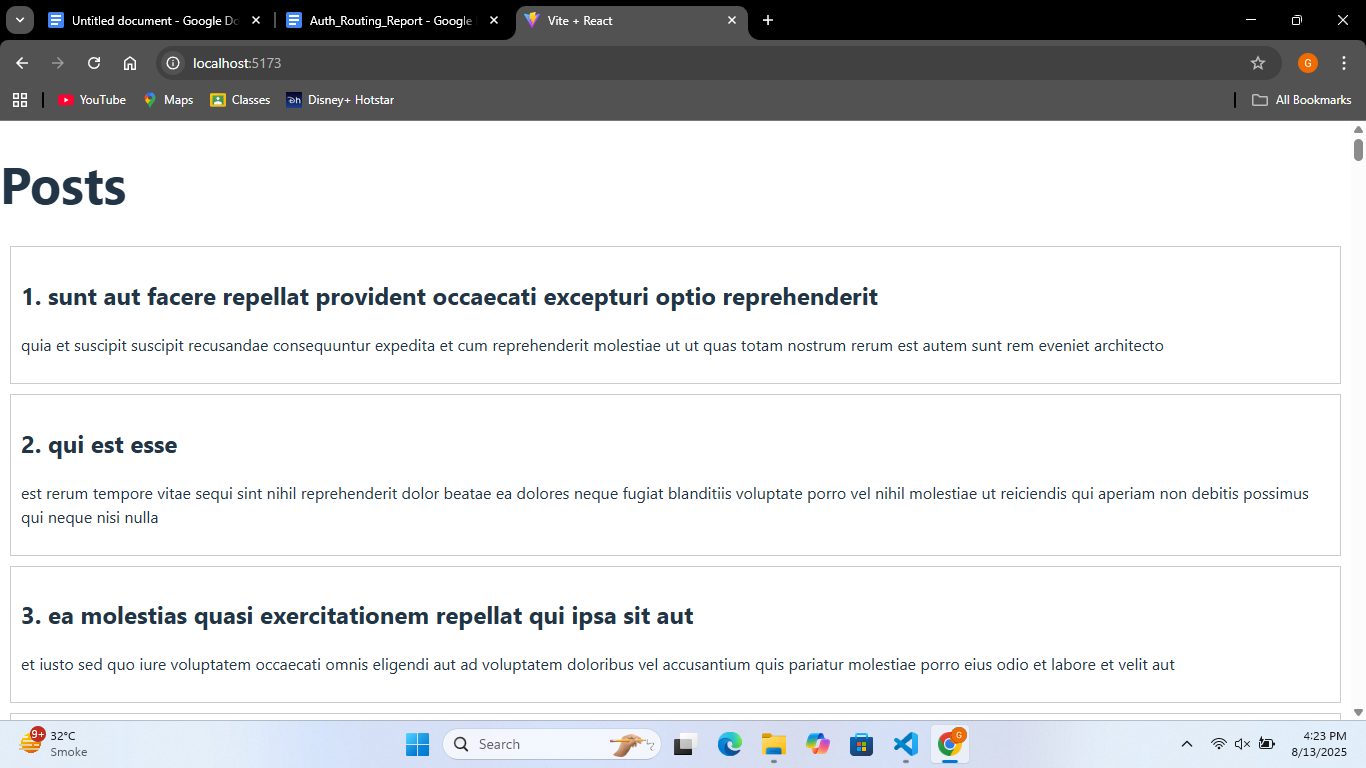
Open the URL printed in the terminal (usually http://localhost:5173) in your browser.

**●Using the App:**

**Open the application in a web browser.**

You will the list of all the data that was fetched from the given public API

**8. Output Screenshot**



## **9. Conclusion**

This project helped in understanding how to integrate public APIs into a React application, manage component state, and handle asynchronous operations. It also provided experience in rendering dynamic data and improving UI presentation.

## 

## 

## **10. References**

* **JSONPlaceholder API** –<https://jsonplaceholder.typicode.com>
* [**Vite Documentation**](https://vite.dev/guide/)
* [**React Documentation**](https://react.dev/)