## **Objectives**

* Explain how to consume REST APIs from React applications

1. **Definition**:  
   REST APIs allow communication between a React frontend and a backend server using HTTP methods like GET, POST, PUT, DELETE.
2. **Common HTTP Methods**:

* GET: Fetch data
* POST: Submit new data
* PUT/PATCH: Update data
* DELETE: Remove data

1. **Ways to Consume APIs in React**:

* Using **fetch()** (built-in JavaScript method)
* Using **axios** (external library for easier API handling)

1. **Steps to Consume an API**:

* Call API inside componentDidMount() (class) or useEffect() (functional)
* Use useState() or this.state to store the response
* Render the data using JSX
* Handle loading and error states

1. **Install Axios (if used)**:

* npm install axios

1. **API Call Example (Using Axios)**:
   * axios.get('https://api.example.com/data')
   * .then(res => setData(res.data))
   * .catch(err => console.log(err));
2. **Error Handling**:

Always handle errors using .catch() or try/catch blocks for better UX.

1. **Where to Place API Calls**:

* componentDidMount() – class components
* useEffect(() => {}, []) – functional components

1. **Best Practices**:

* Use async/await for clean code
* Show loading indicators
* Modularize API logic in separate service files

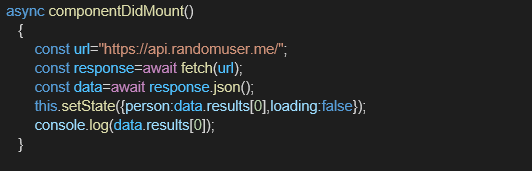
## **Notes**

Estimated time to complete this lab: **60 minutes.**

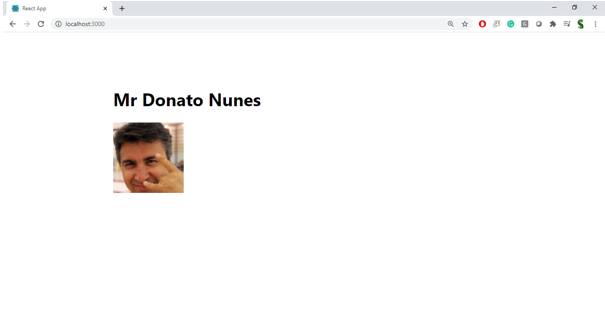
Create a React Application “fetchuserapp” which will retrieve the user details from <https://api.randomuser.me/> and display the title, firstname and image of a user.

Create a component named “Getuser” and in the asynchronous method “ComponentDidMount ()” invoke the URL using fetch method and the response can be displayed in the render method of the component.

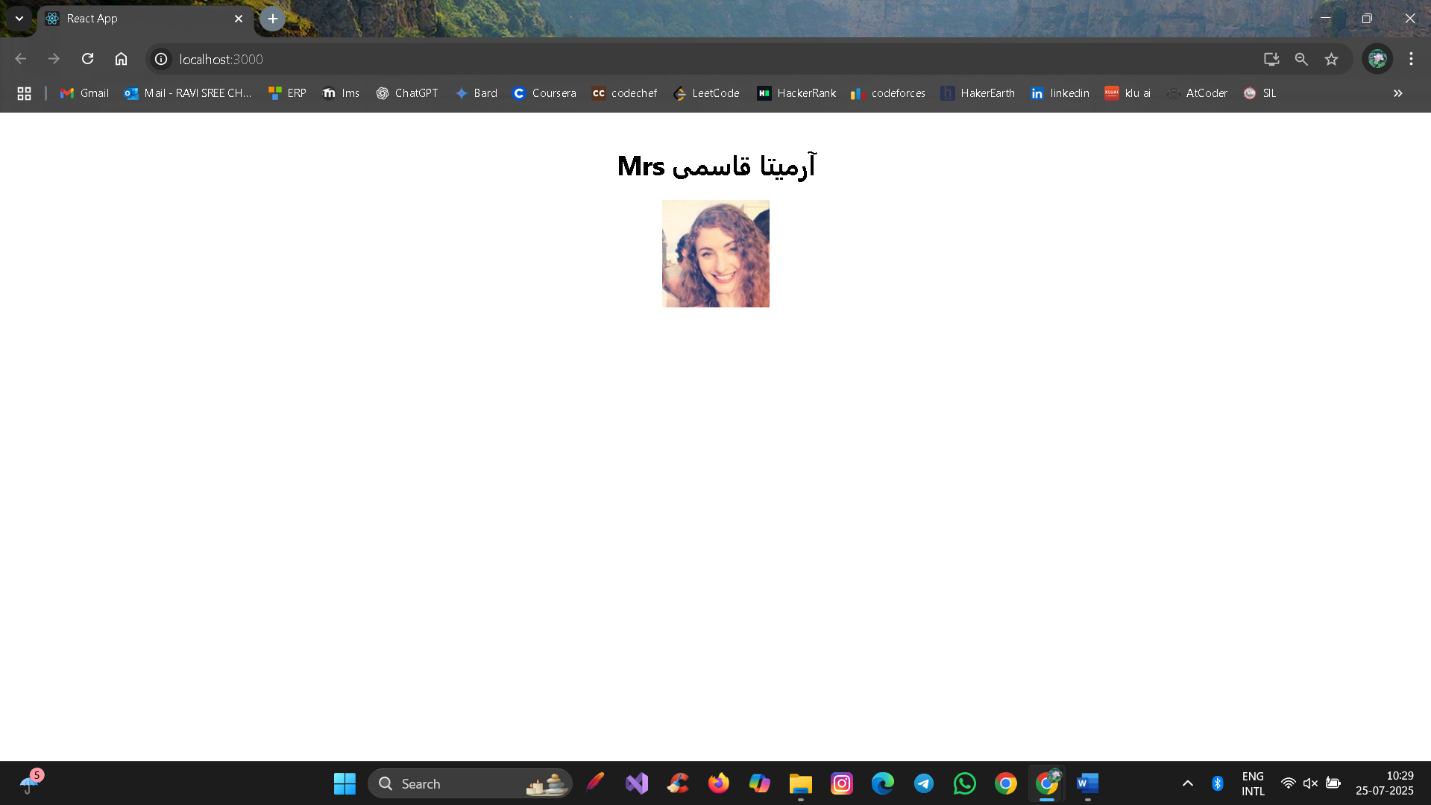
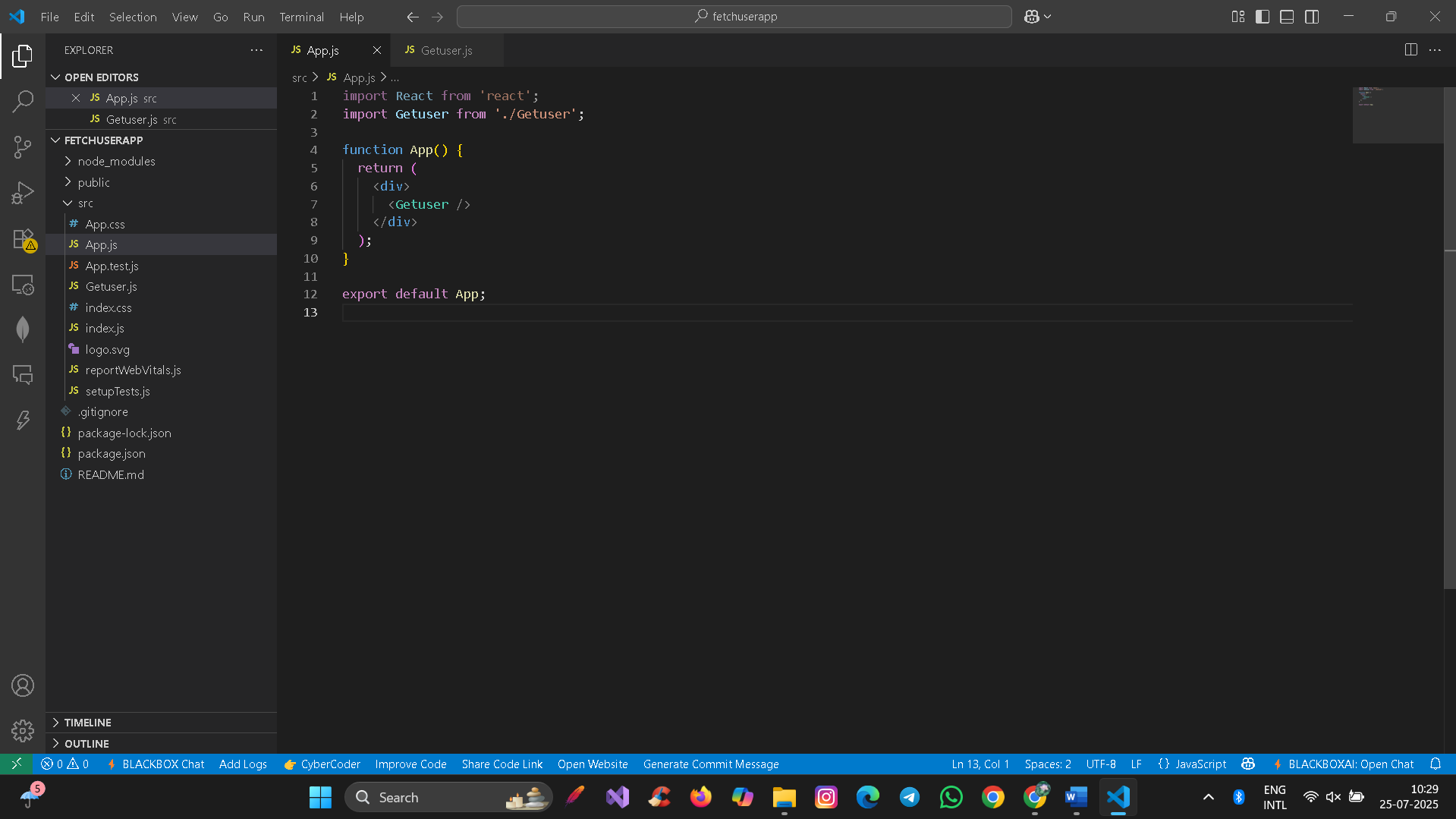
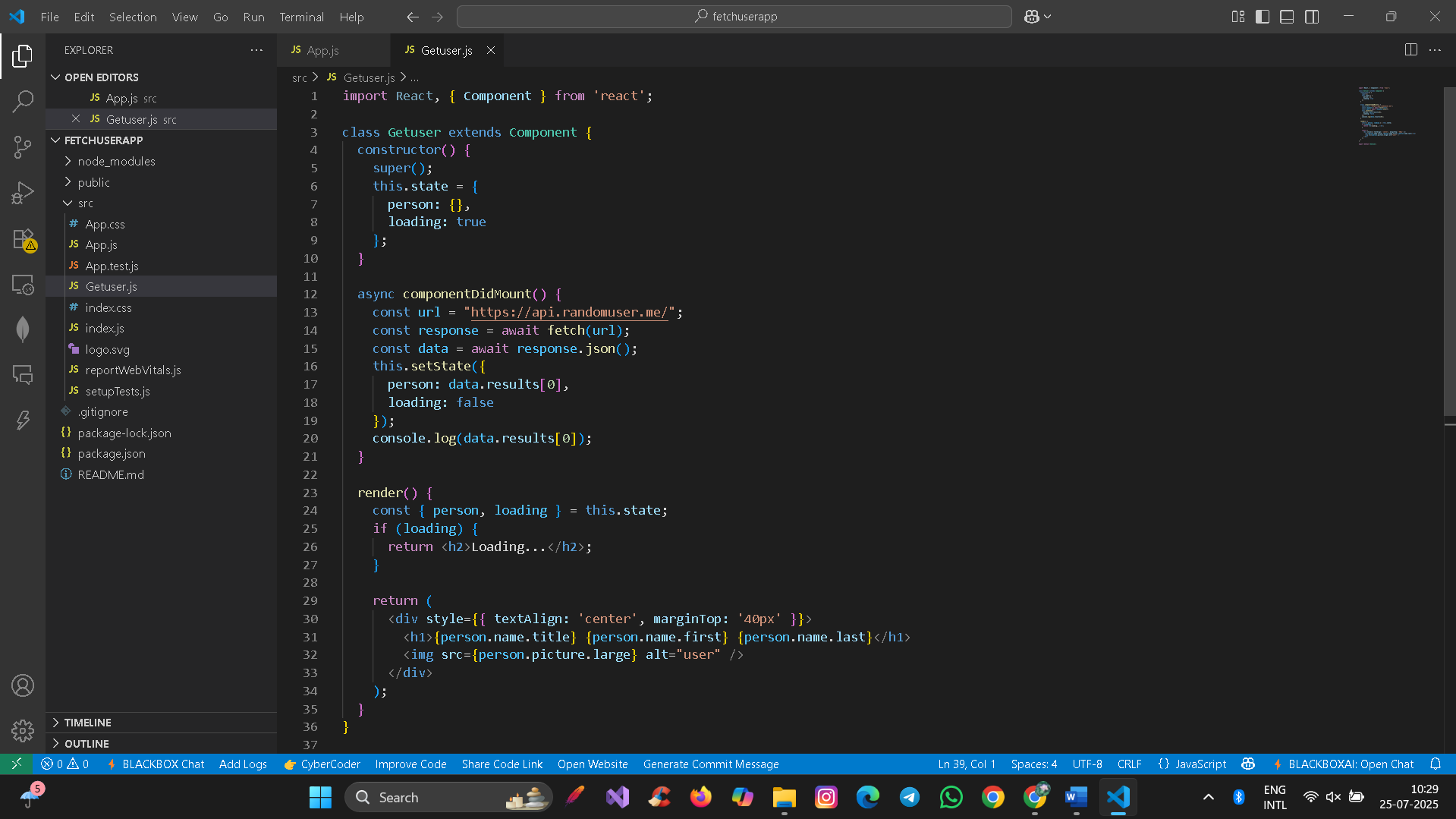
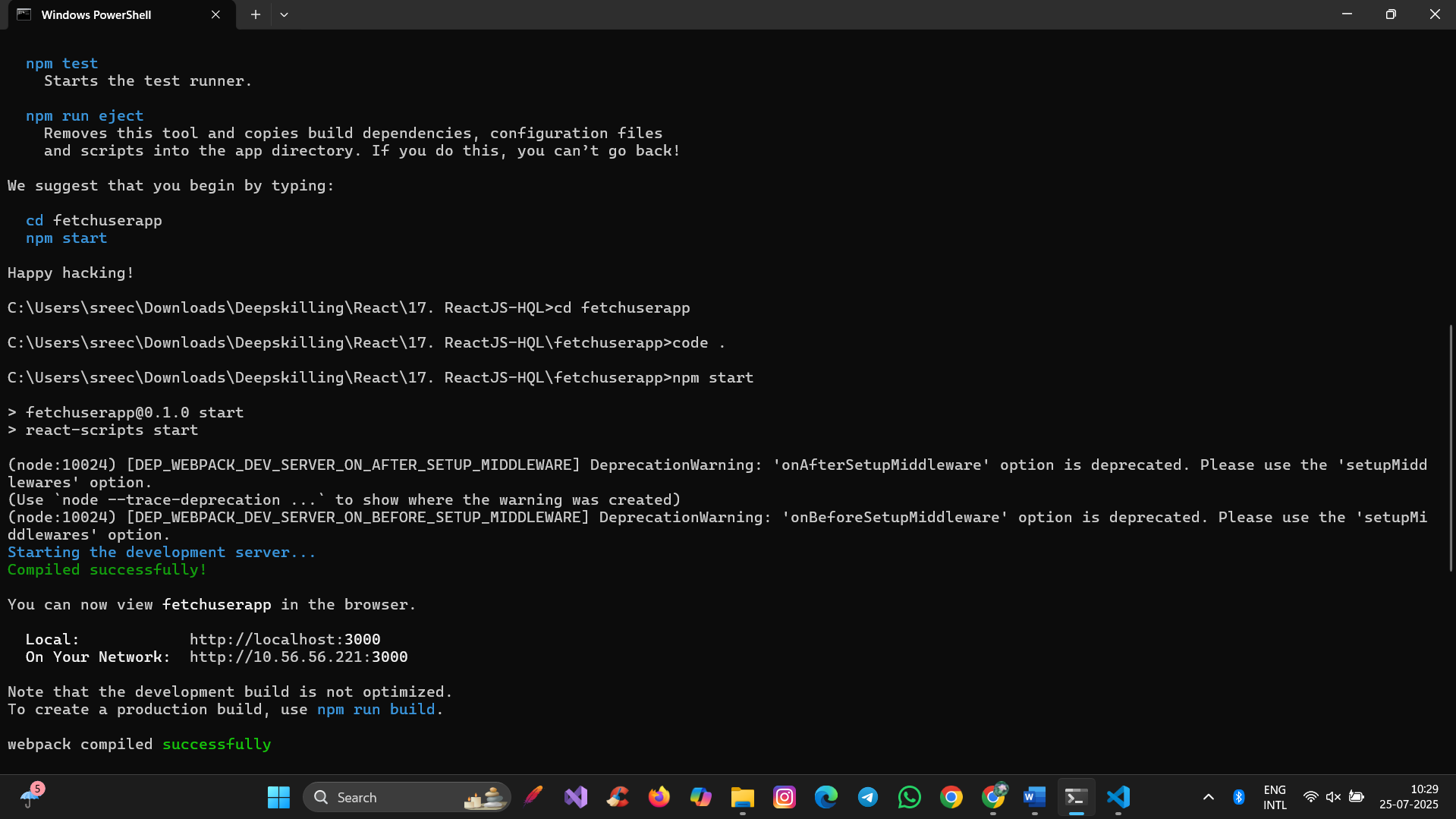
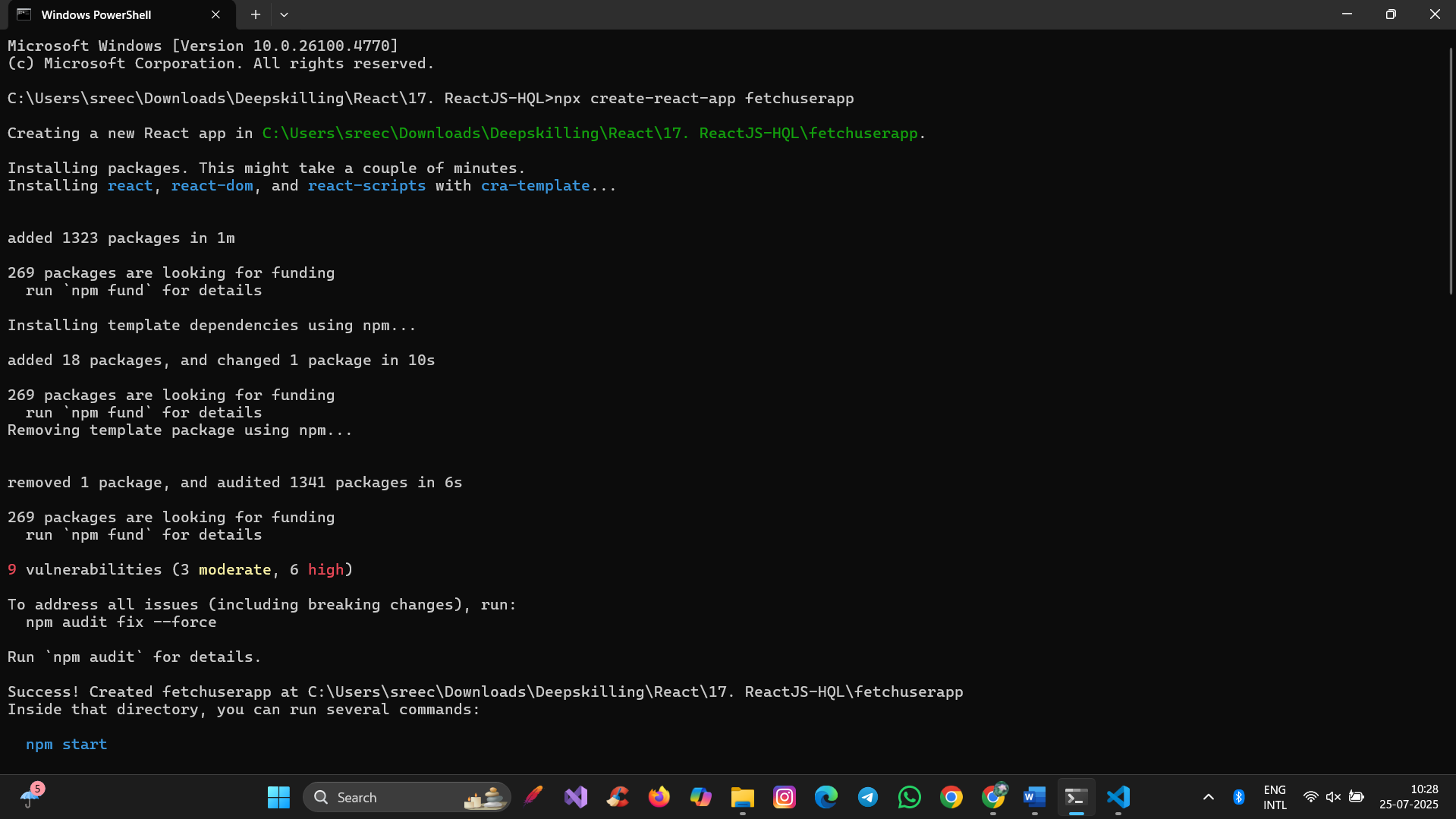
Code Snippet in Getuser Component:



**Expected Output:**



**Handson:**

****