**19 . REACTJS – HOL**

**Unit Testing and Mocking in React using Jest and Axios**

**1. Objectives**

* Understanding the need for isolation in testing
* Understanding the concept of mocking
* Using Jest for unit testing and mocking

**2. Learning Outcomes**

In this hands-on lab, you will learn how to:

* Unit test modules in isolation
* Create and configure mocks and spies using Jest

**3. Prerequisites**

The following tools are required:

* Node.js
* NPM
* Visual Studio Code

**5. Steps to Complete**

**Step 1: Create React App**

npx create-react-app gitclientapp

cd gitclientapp

code .

**Step 2: Install Axios**

npm install axios

**Step 3: Create GitClient Module**

**Create GitClient.js inside the src/ folder:**

// src/GitClient.js

import axios from 'axios';

class GitClient {

async getRepositories(user) {

const response = await axios.get(`https://api.github.com/users/${user}/repos`);

return response.data.map(repo => repo.name);

}

}

export default GitClient;

**Step 4: Modify App Component**

Update src/App.js:

javascript

CopyEdit

import React, { useEffect, useState } from 'react';

import GitClient from './GitClient';

function App() {

const [repos, setRepos] = useState([]);

const gitClient = new GitClient();

useEffect(() => {

gitClient.getRepositories('techiesyed').then(setRepos);

}, []);

return (

<div>

<h1>Repositories</h1>

<ul>

{repos.map((repo, index) => (

<li key={index}>{repo}</li>

))}

</ul>

</div>

);

}

export default App;

**Step 5: Run the Application**

npm start

**Step 6: Create Test File**

Create src/GitClient.test.js:

javascript

CopyEdit

import axios from 'axios';

import GitClient from './GitClient';

jest.mock('axios');

describe('Git Client Tests', () => {

test('should return repository names for techiesyed', async () => {

const mockData = {

data: [

{ name: 'RepoOne' },

{ name: 'RepoTwo' }

]

};

axios.get.mockResolvedValue(mockData);

const client = new GitClient();

const repos = await client.getRepositories('techiesyed');

expect(repos).toEqual(['RepoOne', 'RepoTwo']);

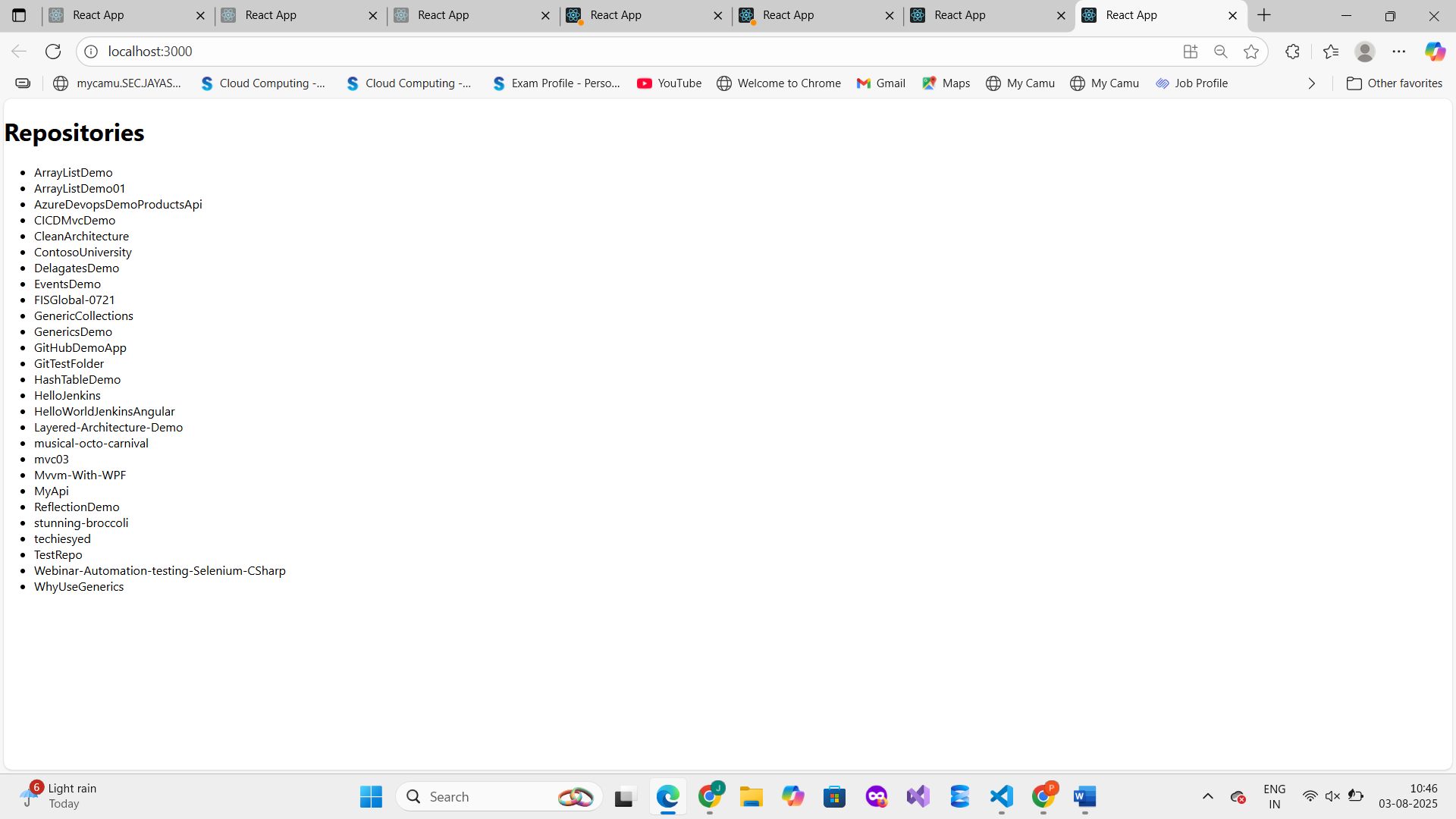
});

});

**7. Run Tests and get output**

npm test

**Output:**



**8. Conclusion**

In this lab, you built a React app to fetch GitHub repositories and used Jest to write isolated unit tests. By mocking Axios, you tested logic without depending on actual API responses, making tests more reliable and faster.