ReactJS Mocking and Unit Testing Lab Report: GitClient Module

# Objectives

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

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

- Unit test modules in isolation  
- Create and configure mocks and spies

# Prerequisites

- Node.js  
- NPM  
- Visual Studio Code

# Step-by-Step Instructions

## Step 1: Create the React App

Open terminal and run:  
npx create-react-app gitclientapp  
cd gitclientapp  
code .

## Step 2: Install Axios

Install Axios using:

npm install axios

## Step 3: Create GitClient.js

Inside src/, create GitClient.js and paste:

import axios from 'axios';  
  
class GitClient {  
 static async getRepositories(username) {  
 const response = await axios.get(`https://api.github.com/users/${username}/repos`);  
 return response.data.map(repo => repo.name);  
 }  
}  
  
export default GitClient;

## Step 4: Modify App.js

Use the GitClient to fetch repositories and display them.

import React, { useEffect, useState } from 'react';  
import GitClient from './GitClient';  
  
function App() {  
 const [repos, setRepos] = useState([]);  
  
 useEffect(() => {  
 GitClient.getRepositories('techiesyed').then(data => setRepos(data));  
 }, []);  
  
 return (  
 <div className="App">  
 <h1>GitHub Repositories</h1>  
 <ul>  
 {repos.map((repo, index) => (  
 <li key={index}>{repo}</li>  
 ))}  
 </ul>  
 </div>  
 );  
}  
  
export default App;

## Step 5: Run the Application

Use the command: npm start

## Step 6: Create Unit Test for GitClient

Create src/GitClient.test.js and add:

import axios from 'axios';  
import GitClient from './GitClient';  
  
jest.mock('axios');  
  
describe('Git Client Tests', () => {  
 test('should return repository names for techiesyed', async () => {  
 const dummyData = {  
 data: [  
 { name: 'repo1' },  
 { name: 'repo2' },  
 { name: 'repo3' }  
 ]  
 };  
 axios.get.mockResolvedValue(dummyData);  
   
 const repos = await GitClient.getRepositories('techiesyed');  
 expect(repos).toEqual(['repo1', 'repo2', 'repo3']);  
 });  
});

## Step 7: Run the Tests

In terminal, run: npm test







