**19.ReactJS- HOL**

**1.Understanding need for isolation in testing**

* Test one thing at a time: Isolation ensures that each test focuses on a specific function or component, not others.
* Avoid dependencies: It prevents outside factors (like network calls or databases) from affecting test results.
* Reliable and repeatable: Isolated tests always give the same result, making debugging and fixing easier.

**2.Understanding the concept of mocking**

* Fake versions of real functions: Mocking replaces real functions (like API calls or database access) with test-friendly versions.
* Control test behavior: Mocks let you simulate specific outputs, errors, or delays without depending on external systems.
* Used for unit testing: Helps test individual units in isolation by avoiding real dependencies.

**3.Using Jest for unit testing and mocking**

* Jest is a testing framework by Facebook, used to test JavaScript and React applications easily.
* Unit Testing with Jest: Use test(), expect(), and describe() to write and group test cases.
* Mocking with Jest: Use jest.fn() or jest.mock() to create and control mock functions for testing.

**React Application – gitclientapp**

As an intern at OpenAI you are assigned the task of creating and testing a React application which will fetch and display a list of repository names for a given user.

1. Create a new React application using *create-react-app* tool and name it as “gitclientapp”.
2. Open the application using VS Code
3. Go to terminal in VS Code and install the support for “axios” react library to make calls to the GitHub API.



*Figure 2: Install axios library*

1. Create a new file with the name as GitClient.js in src folder of the application
2. Add the following code to create a class named GitClient which will make calls to the *api.github.com* to fetch the repositories as follows.



*Figure 3: GitClient module*

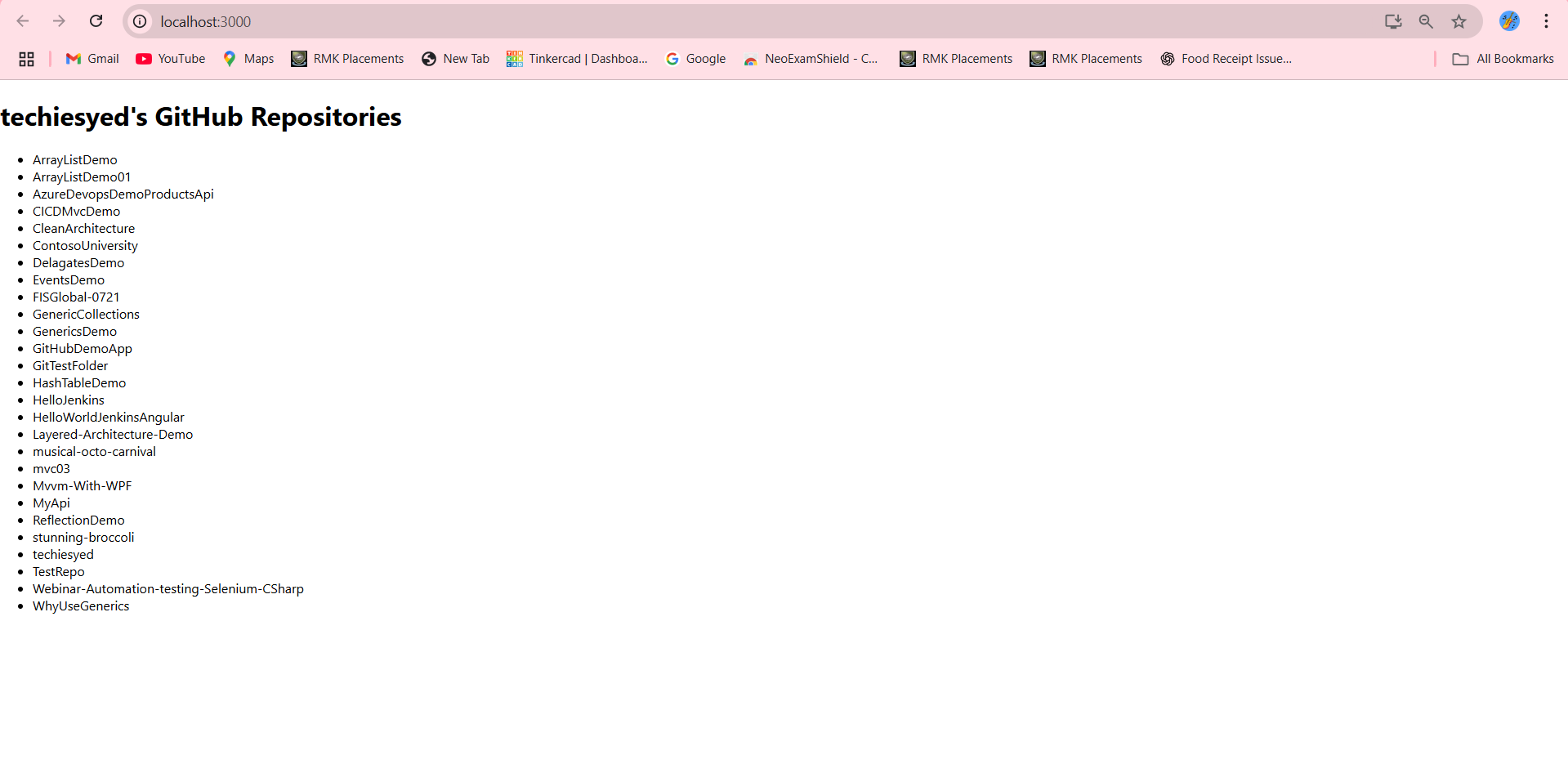
1. Modify the App component to use the declared module to fetch and display the repositories data as follows.



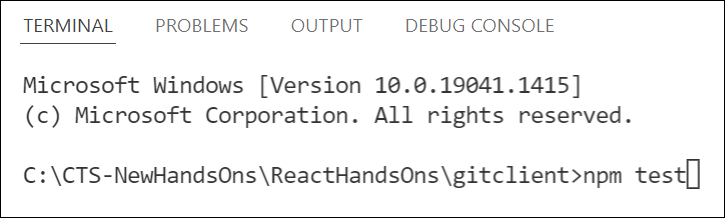
*Figure 4: App Component*

1. Build and Run the application using *npm start* command. The output should look similar to below.

*Figure 5: Application Output*

**

1. Create a new unit test file named as GitClient.test.js to unit test the newly created module.
2. Import axios and GitClient into the unit test file.
3. Describe the test name as “Git Client Tests”
4. Create a unit test using the test() and give the test name as “should return repository names for techiesyed”
5. Mock the axios object to return the dummy data
6. Invoke the getRepositories() method of GitClient and see it’s returning the mocked data instead of making an actual call to *api.github.com*
7. Run tests using *npm test* command.



*Figure 6: Run tests*

**Output:**

