**19.ReactJs HOL**

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. Create a new React application using *create-react-app* tool and name it as “gitclientapp”.

**React App : gitclientapp**

**GitClient.js**

import axios from 'axios';

class GitClient {

    async getRepositories(username) {

        const url = `https://api.github.com/users/${username}/repos`;

        const response = await axios.get(url);

        return response.data.map(repo => repo.name); // Return repo names only

    }

}

export default GitClient;

**GitClient.test.js**

import axios from 'axios';

import GitClient from './GitClient';

jest.mock('axios');

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

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

        const dummyRepos = [

            { name: "repo1" },

            { name: "repo2" },

            { name: "repo3" }

        ];

        axios.get.mockResolvedValue({ data: dummyRepos });

        const client = new GitClient();

        const result = await client.getRepositories("techiesyed");

        expect(result).toEqual(["repo1", "repo2", "repo3"]);

        expect(axios.get).toHaveBeenCalledWith("https://api.github.com/users/techiesyed/repos");

    });

});

**App.js**

import React, { useState } from 'react';

import GitClient from './GitClient';

function App() {

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

    const [username, setUsername] = useState('');

    const [loading, setLoading] = useState(false);

    const fetchRepos = async () => {

        setLoading(true);

        try {

            const client = new GitClient();

            const result = await client.getRepositories(username);

            setRepos(result);

        } catch (error) {

            console.error("Error fetching repos:", error);

            alert("User not found or error in fetching data.");

        } finally {

            setLoading(false);

        }

    };

    return (

        <div style={{ padding: "20px" }}>

            <h2>GitHub Repositories</h2>

            <input

                type="text"

                placeholder="Enter GitHub username"

                value={username}

                onChange={e => setUsername(e.target.value)}

            />

            <button onClick={fetchRepos} disabled={loading}>

                {loading ? "Loading..." : "Get Repositories"}

            </button>

            <ul>

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

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

                ))}

            </ul>

        </div>

    );

}

export default App;

**Final Output:**

