**19.REACT**

**Index.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="apple-touch-icon" href="%PUBLIC\_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

    <title>Git Client App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>

</html>

**App.js**

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

import GitClient from './GitClient';

function App() {

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

    useEffect(() => {

        const gitClient = new GitClient();

        gitClient.getRepositories("techiesyed").then(data => setRepos(data));

    }, []);

    return (

        <div>

            <h1>Repositories of techiesyed</h1>

            <ul>

                {repos.map((repo, index) => <li key={index}>{repo}</li>)}

            </ul>

        </div>

    );

}

export default App;

**index.js**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

**GitClient.js**

import axios from 'axios';

class GitClient {

    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;

**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 mockedData = [

            { name: 'repo1' },

            { name: 'repo2' },

            { name: 'repo3' }

        ];

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

        const client = new GitClient();

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

        expect(repos).toEqual(['repo1', 'repo2', 'repo3']);

    });

});

**Output:**

