**App.js:**

import './App.css';

const GitClient = require('./GitClient');

import { useEffect, useState } from 'react';

function App() {

  const [repositories, setRepositories] = useState([]);

  useEffect(() => {

    GitClient.getRepositories('techieSyed')

      .then(r => setRepositories(r.data))

      .catch(err => console.error(err));

  }, []);

  return (

    <div className="App">

      <h1>Git repositories of User - TechieSyed</h1>

      {repositories.map(r => (

        <p key={r.name}>{r.name}</p>

      ))}

    </div>

  );

}

export default App;

**GitClient.js:**

const axios = require('axios');

class GitClient {

  static getRepositories(userName) {

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

    return axios.get(url);

  }

}

module.exports = GitClient;

**GitClient.test.js:**

const axios = require('axios');

const GitClient = require('./GitClient');

jest.mock('axios');

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

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

    const dummyData = [

      { name: "repo1" },

      { name: "repo2" }

    ];

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

    const response = await GitClient.getRepositories("techiesyed");

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

    expect(response.data).toEqual(dummyData);

  });

});

**App.test.js:**

import { render, screen } from '@testing-library/react';

import App from './App';

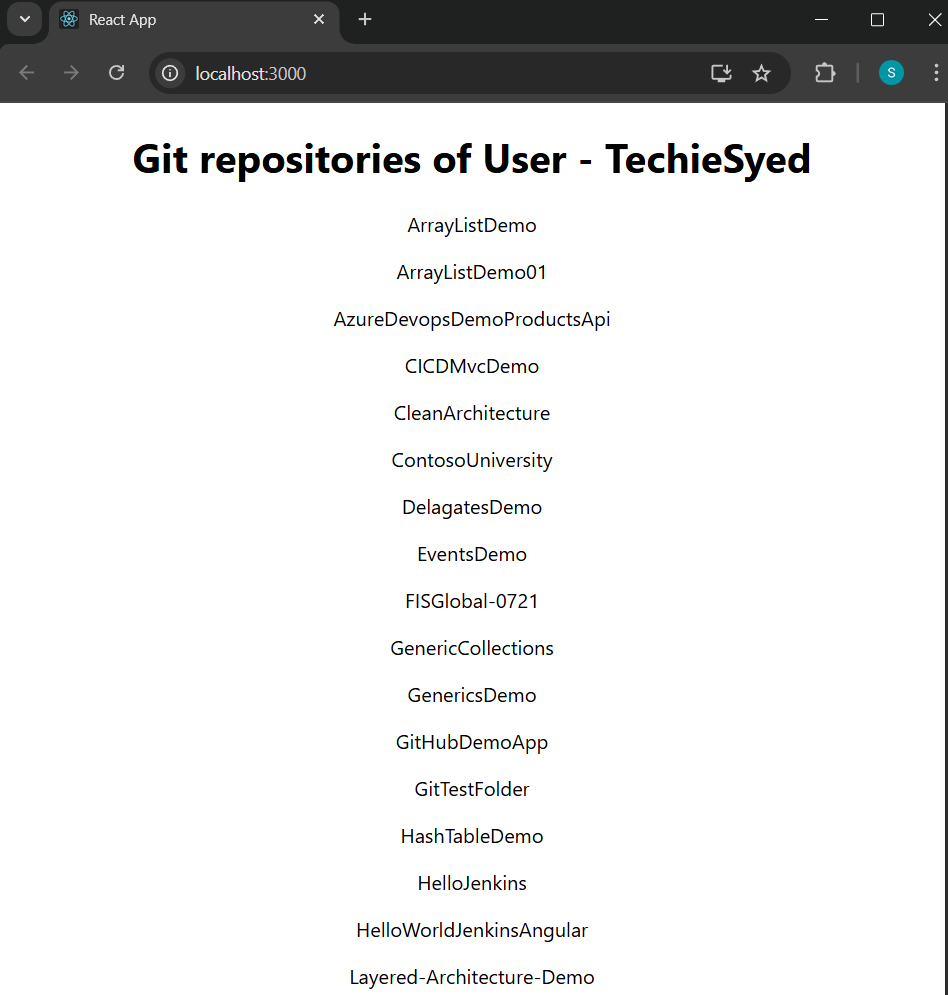
test('renders Git repositories heading', () => {

  render(<App />);

  const headingElement = screen.getByText(/Git repositories of User/i);

  expect(headingElement).toBeInTheDocument();

});



**After Testing:** npm test,

