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

**GitHubUser.js**

import React, { useState } from 'react';

import axios from 'axios';

*function* GitHubUser() {

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

*const* [userData, setUserData] = useState(null);

*const* [error, setError] = useState('');

*const* fetchGitHubUser = async (*e*) *=>* {

    e.preventDefault();

    try {

*const* response = await axios.get(`https://api.github.com/users/${username}`);

      setUserData(response.data);

      setError('');

    } catch (err) {

      setUserData(null);

      setError('User not found');

    }

  };

  return (

    <div>

      <h2>GitHub User Lookup</h2>

      <form onSubmit={fetchGitHubUser}>

        <input

          type="text"

          placeholder="Enter GitHub username"

          value={username}

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

        />

        <button type="submit">Fetch</button>

      </form>

      {error && <p style={{ color: 'red' }}>{error}</p>}

      {userData && (

        <div>

          <h3>{userData.name}</h3>

          <img src={userData.avatar\_url} alt="avatar" width="100" />

          <p>Followers: {userData.followers}</p>

          <p>Public Repos: {userData.public\_repos}</p>

        </div>

      )}

    </div>

  );

}

export default GitHubUser;

**GitHubUser.test.js**

import React from 'react';

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

import GitHubUser from '../components/GitHubUser';

import axios from 'axios';

import MockAdapter from 'axios-mock-adapter';

describe('GitHubUser component', () *=>* {

*const* mock = new MockAdapter(axios);

  it('fetches and displays user data', async () *=>* {

*const* mockData = {

      name: 'Adeline Joanna',

      avatar\_url: 'https://example.com/avatar.png',

      followers: 99,

      public\_repos: 20,

    };

    mock.onGet('https://api.github.com/users/adeline').reply(200, mockData);

    render(<*GitHubUser* />);

*const* input = screen.getByPlaceholderText(/enter github username/i);

*const* button = screen.getByRole('button', { name: /fetch/i });

    fireEvent.change(input, { target: { value: 'adeline' } });

    fireEvent.click(button);

    await waitFor(() *=>* {

      expect(screen.getByText('Adeline Joanna')).toBeInTheDocument();

      expect(screen.getByText(/followers: 99/i)).toBeInTheDocument();

      expect(screen.getByText(/public repos: 20/i)).toBeInTheDocument();

    });

  });

  it('shows error on invalid user', async () *=>* {

    mock.onGet('https://api.github.com/users/unknownuser').reply(404);

    render(<*GitHubUser* />);

    fireEvent.change(screen.getByPlaceholderText(/enter github username/i), {

      target: { value: 'unknownuser' },

    });

    fireEvent.click(screen.getByRole('button', { name: /fetch/i }));

    await waitFor(() *=>* {

      expect(screen.getByText(/user not found/i)).toBeInTheDocument();

    });

  });

});

**App.js**

import React from 'react';

import GitHubUser from './components/GitHubUser';

*function* App() {

  return (

    <div className="App">

      <*GitHubUser* />

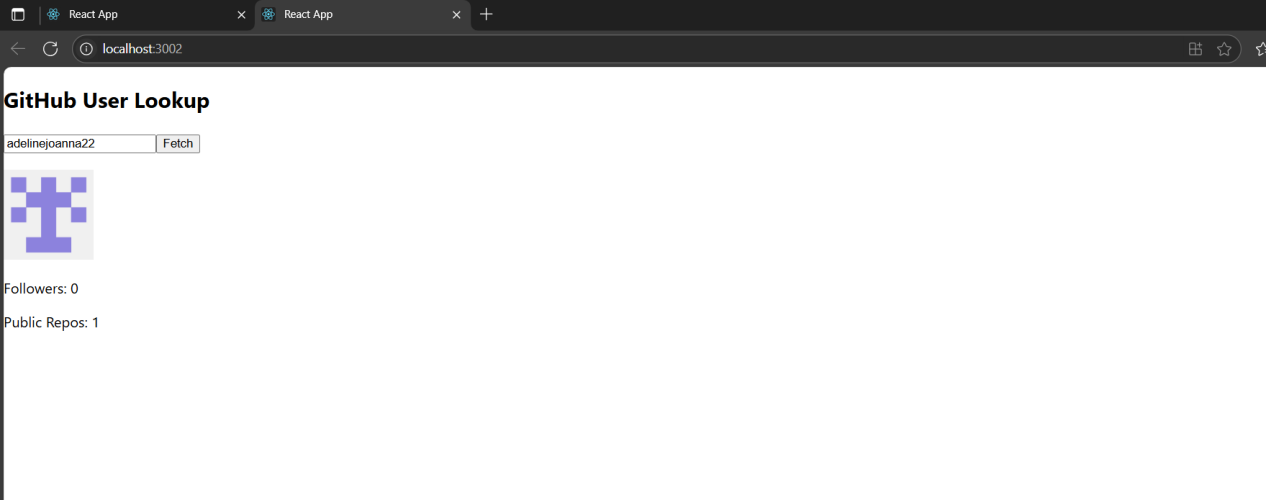
    </div>

  );

}

export default App;

**Output:**

****