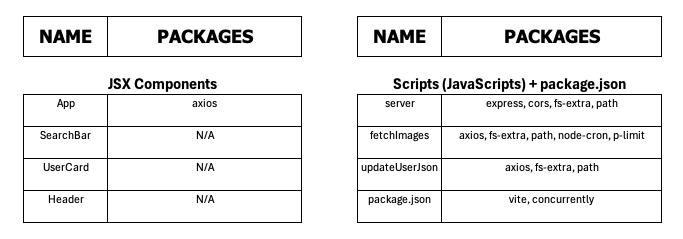
**AAALife Interview assignment**

(For Rafael Padron)

**Requirements**

MY ENVIRONMENT (main package dependencies):

* MacBook Pro: macOS Sequoia version 15.4
* NODE (node -v) version: v23.11.0
* NPX (npx -v) version: 10.9.2
* Vite (npm create vite@latest): v6.3.1
* Axios (npm install axios): v1.9.0
* Express (npm install express): v5.1.0
* Concurrently (npm install ..): v9.1.2
* Visual Studio Code version: 1.99.3

****

1. Create a dynamic reactJS app to accept the search string.

Done.

I am using VITE for being able to create a Project with “JSX”-App and -alike Components.

Thus to create this ReactJS Project, the command line:

$ npm create vite@latest

$ npm create vite@latest

Need to install the following packages:

create-vite@6.4.1

Ok to proceed? (y)

> npx

> create-vite

│

◇ Project name:

│ rocket-search

│

◇ Select a framework:

│ React

│

◇ Select a variant:

│ JavaScript

│

◇ Scaffolding project in /Users/rafaelpadron/JOB\_INTERVIEW\_TASK/rocket-search...

│

└ Done. Now run:

cd rocket-search

npm install

npm run dev

$ cd rocket-search

$ npm install

$ npm install

added 152 packages, and audited 153 packages in 9s

32 packages are looking for funding

run `npm fund` for details

found **0** vulnerabilities

$ npm run dev

$ npm run dev

> rocket-search@0.0.0 dev

> vite

**VITE** v6.3.3 ready in **411** ms

➜ **Local**: http://localhost:**5173**/

➜ **Network**: use **--host** to expose

➜ press **h + enter** to show help

And on Web Browser, go to URL 🡺 <http://localhost:5173>

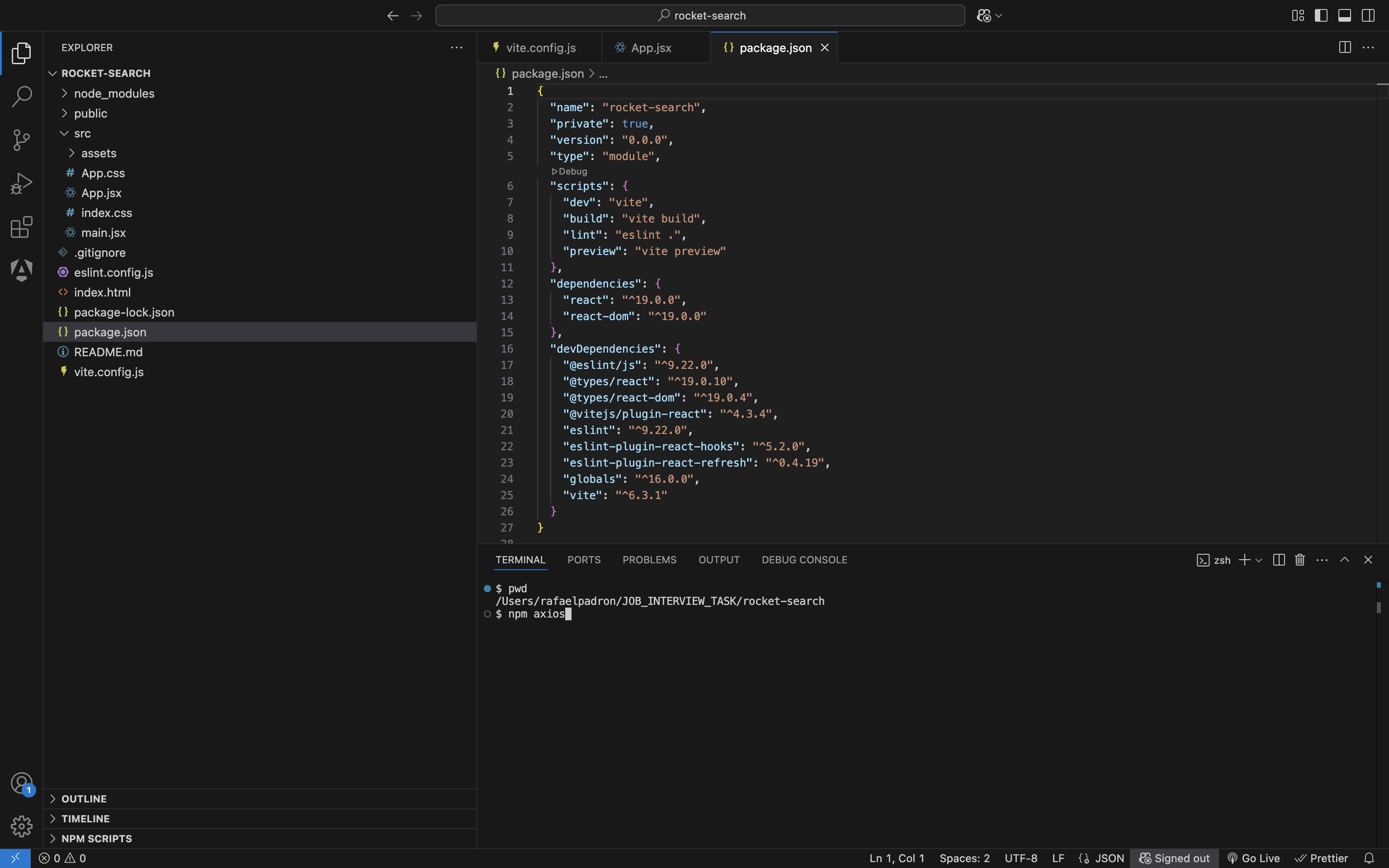
A screenshot of a computer

AI-generated content may be incorrect.

<CTRL-C>

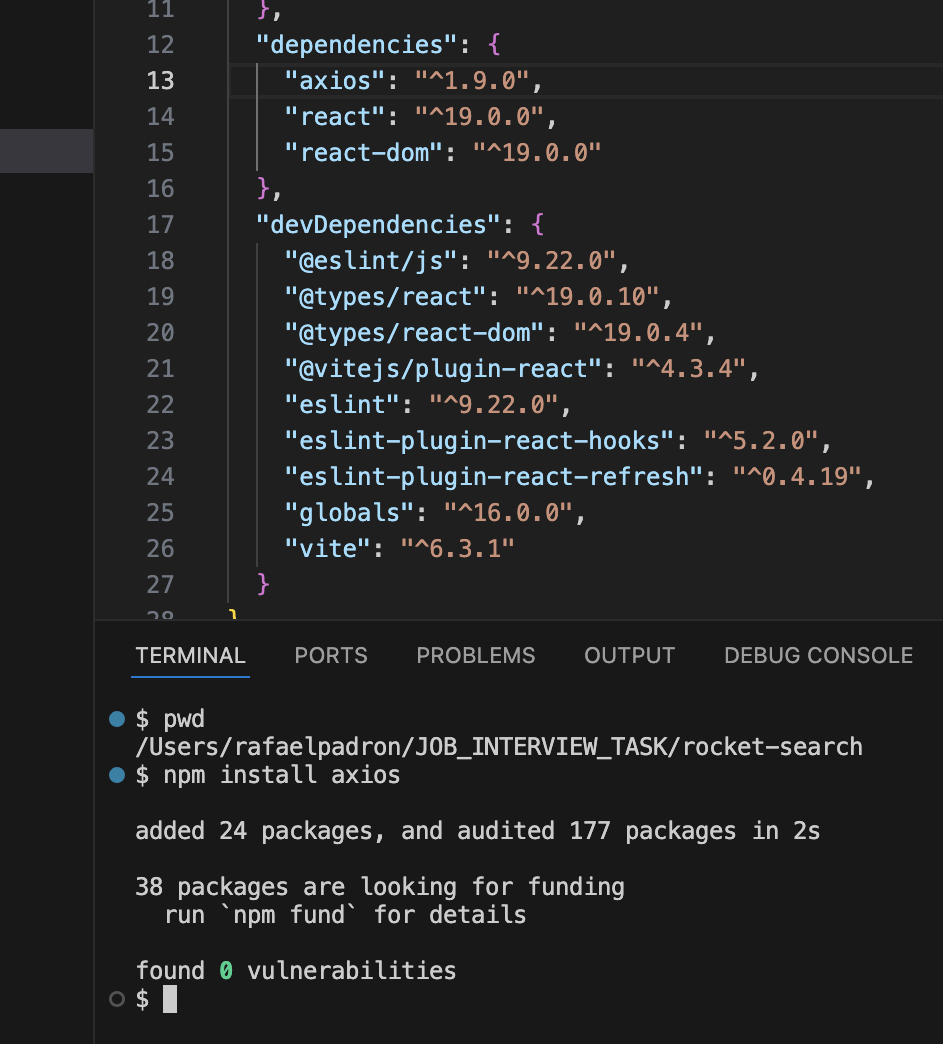
$ code .

🡺 “Yes, I trust the authors” clicking…



We open Visual Studio Code and use its Terminal and check the dependencies inside of the file, package.json.

$ npm install axios



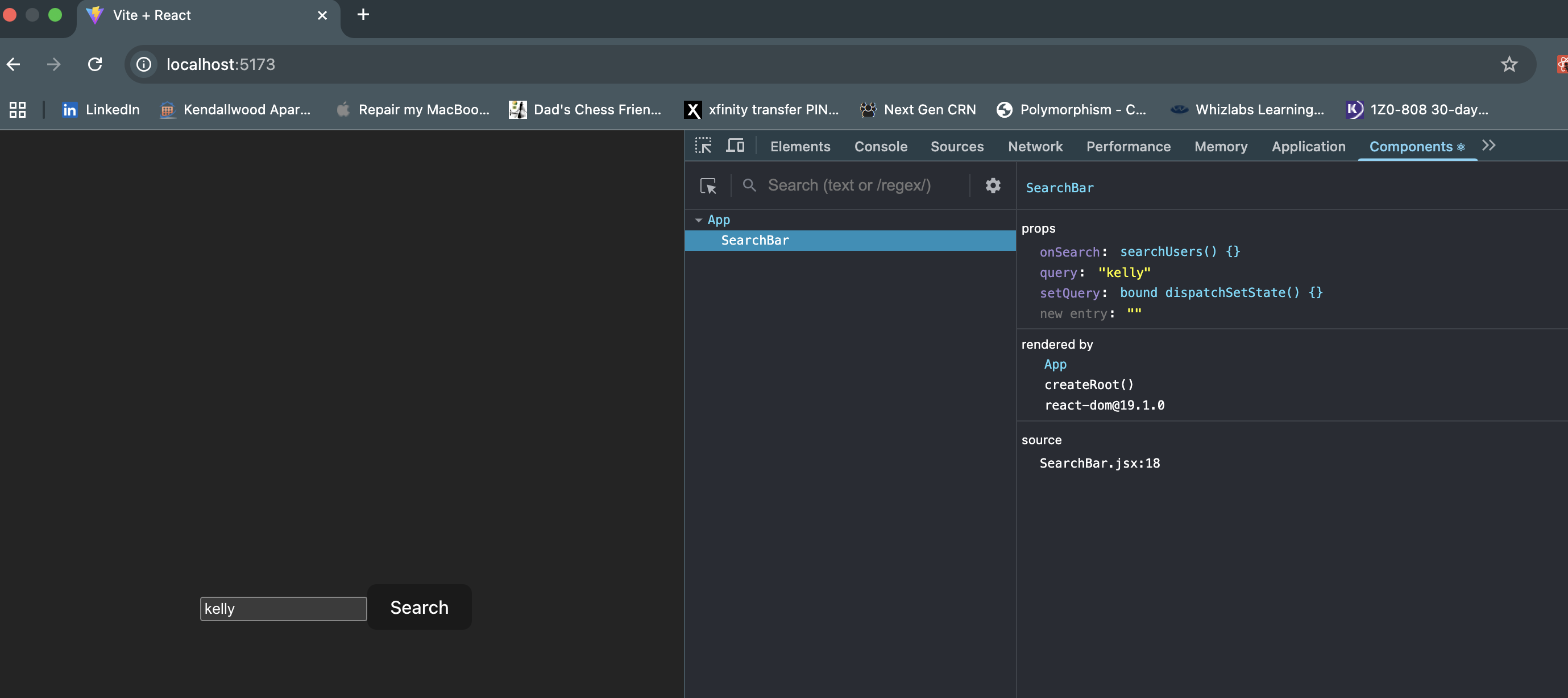
We begin by creating the file:

src/components/SearchBar.jsx

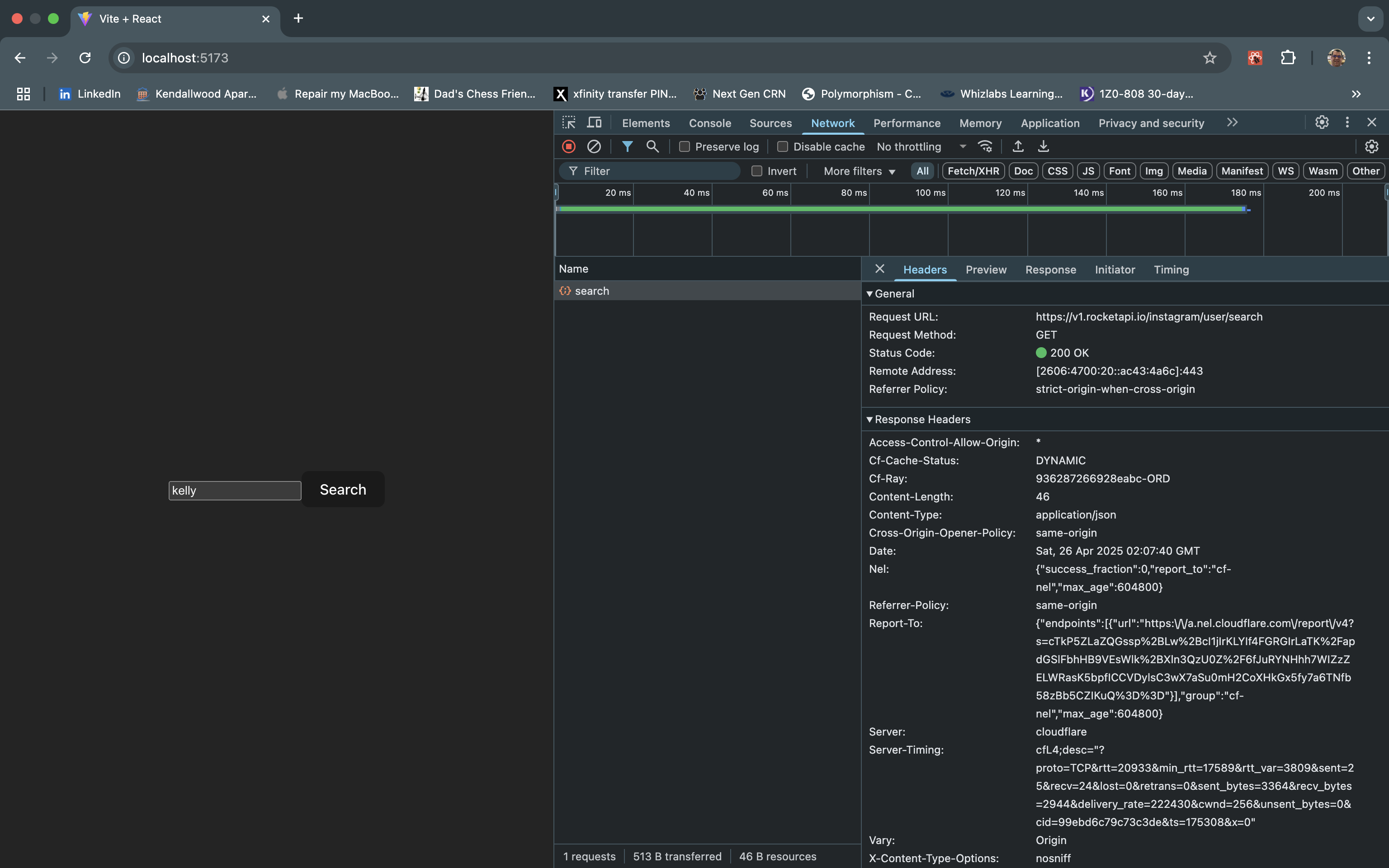
This marks the true start of the development process, where we introduce our first custom component, <SearchBar />. It will act as a child of the root component, <App />. In projects scaffolded with Vite, the entry point (main.jsx) is automatically set up to render the <App /> component, making main.jsx the parent of <App />.

Once the SearchBar component is created, the next step is to import and include <SearchBar /> within the JSX of the App component.

I checked “inspect” and the Chrome Extension, “React Developer Extension”:



Also, I see the HTTP Code status from the endpoint:



* 1. Allow only alphanumeric keys with max length of 30 .

For the event handler, “handleKeyDown”, I added this restriction that added the Hook setState, in the SearchBar a function Also, I see the HTTP Code status from the endpoint:

const handleKeyDown = (e) => {

if (!/^[a-zA-Z0-9]\*$/.test(e.key)) e.preventDefault();

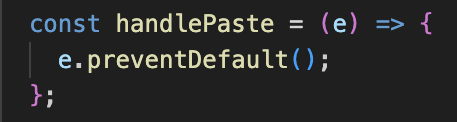
if (query.length >= 30) e.preventDefault();

};

* 1. Do not allow adding more characters to the search field if 30 characters are reached.

Done. See above.

* 1. Block pasting data.   
     Done.



1. Use the search string input parameter to search using the search users rocket api (details attached).

Done.

1. Show the results as a grid of cards.

Done. The UserCard JSX Component was created.

1. Feel free to use any styling.

Done. Respective CSS files were created for each new JSX Component

1. Only include username, fullname and add an image linking to the profile\_pic\_url on the api response.

Done.

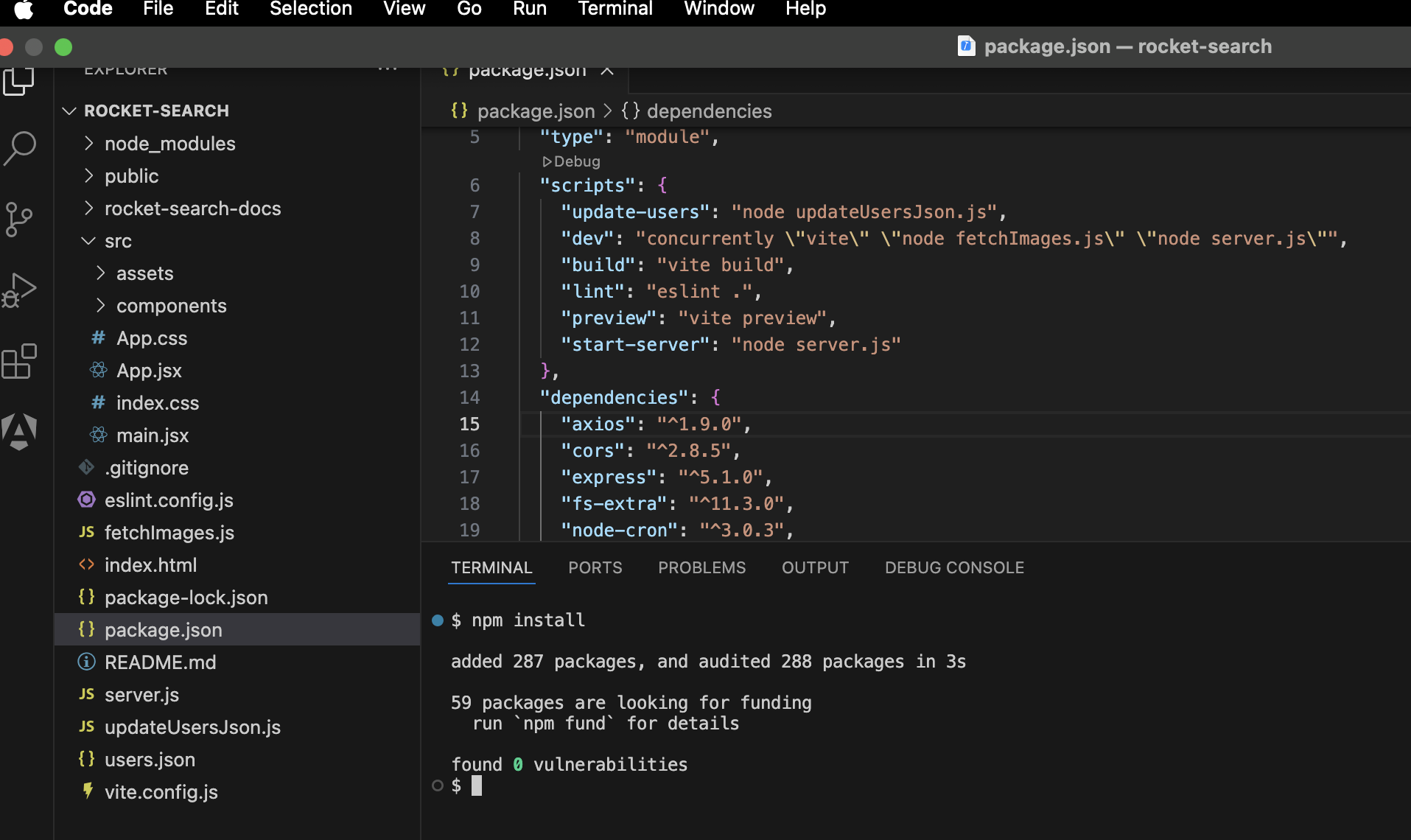
1. Share complete development workspace and share instructions on how to execute and host/run the app on any computer.

Done. Hereto and I also added and updated README.md file.

RESTARTING THE SERVICES AND RUNNING THE NEW REACT-JS APPLICATION:

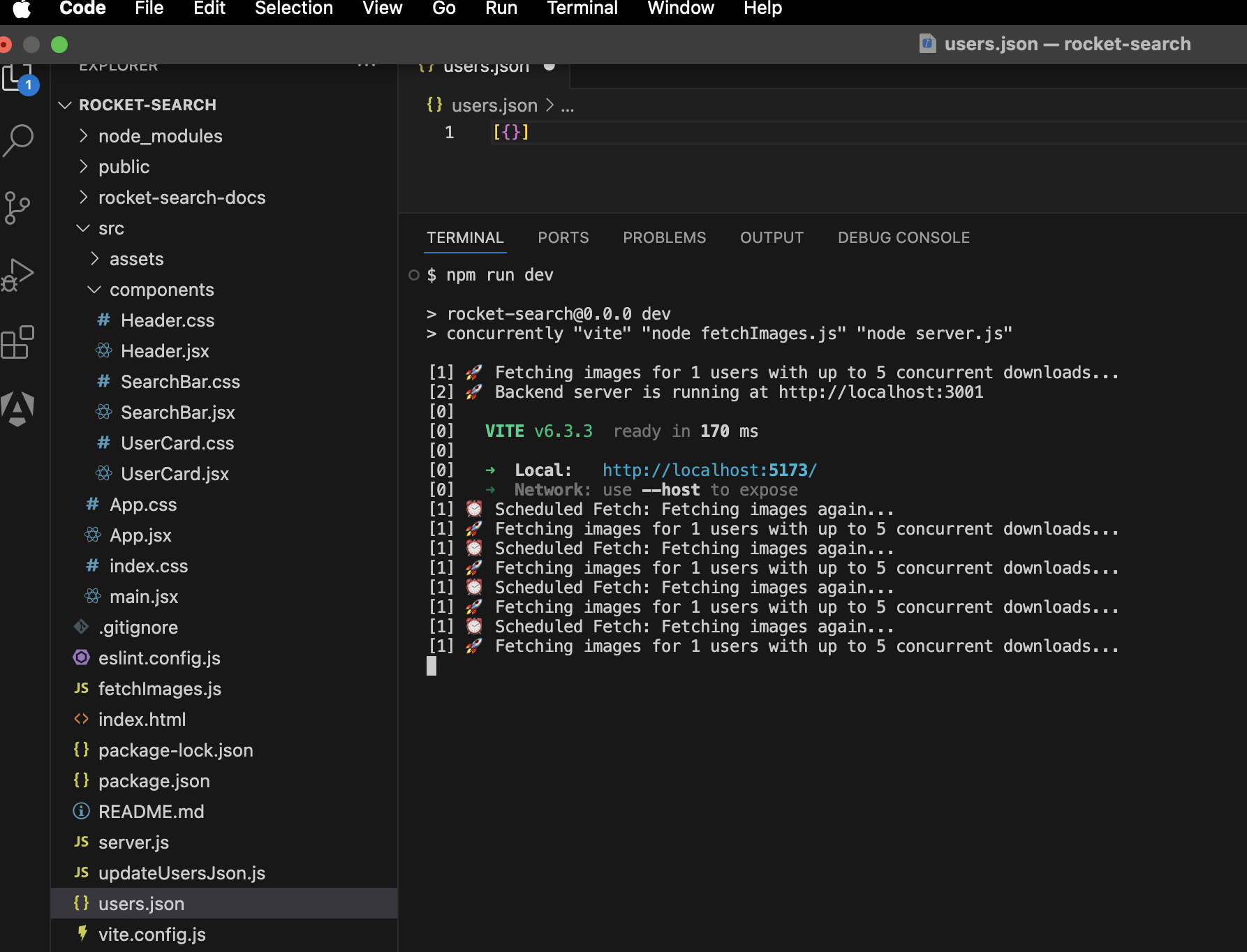
1. Run on the Visual Studio Code Terminal Console, the below command line to install all the dependency modules:

$ npm install



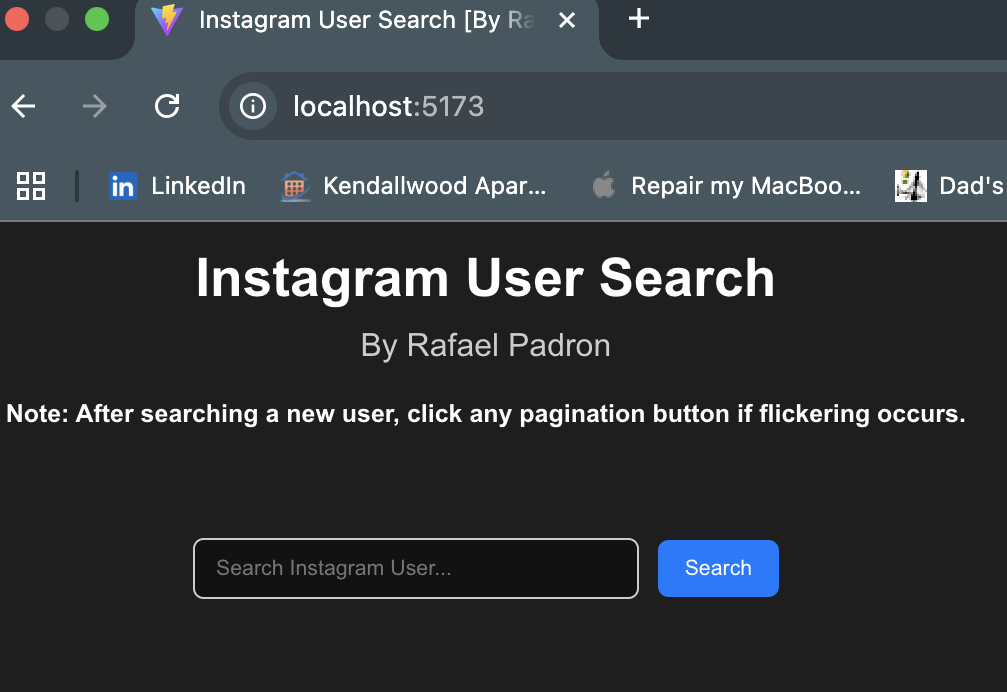
1. Likewise, run on the Visual Studio Code Terminal Console to restart all the services run:

$ npm run dev

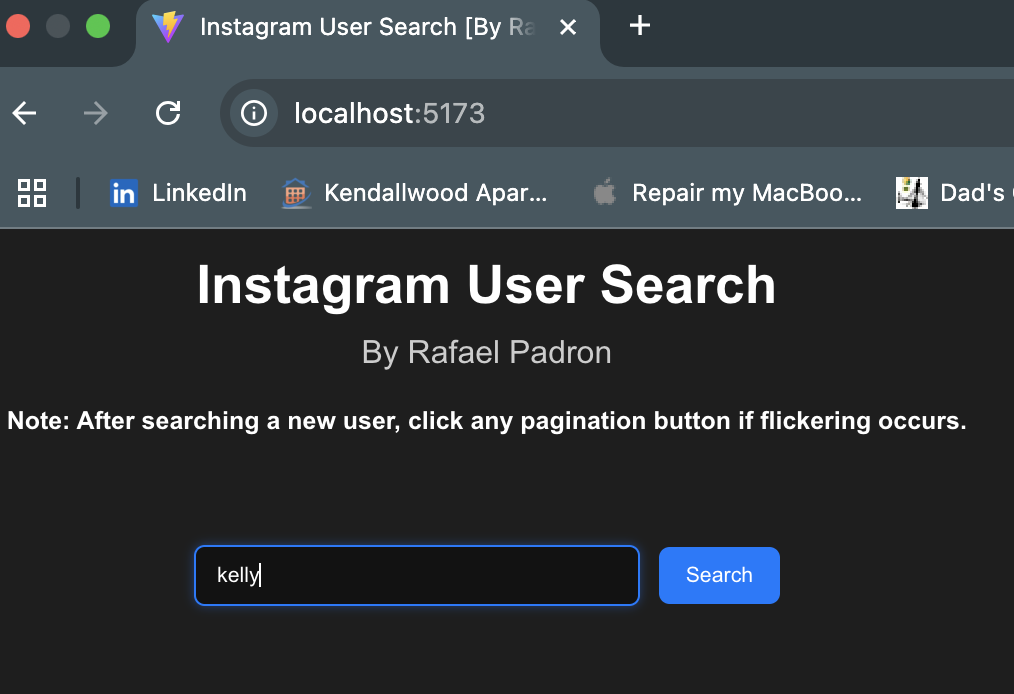


1. Access on Browser, preferebly using Chrome Web Browser:

http://localhost:5173



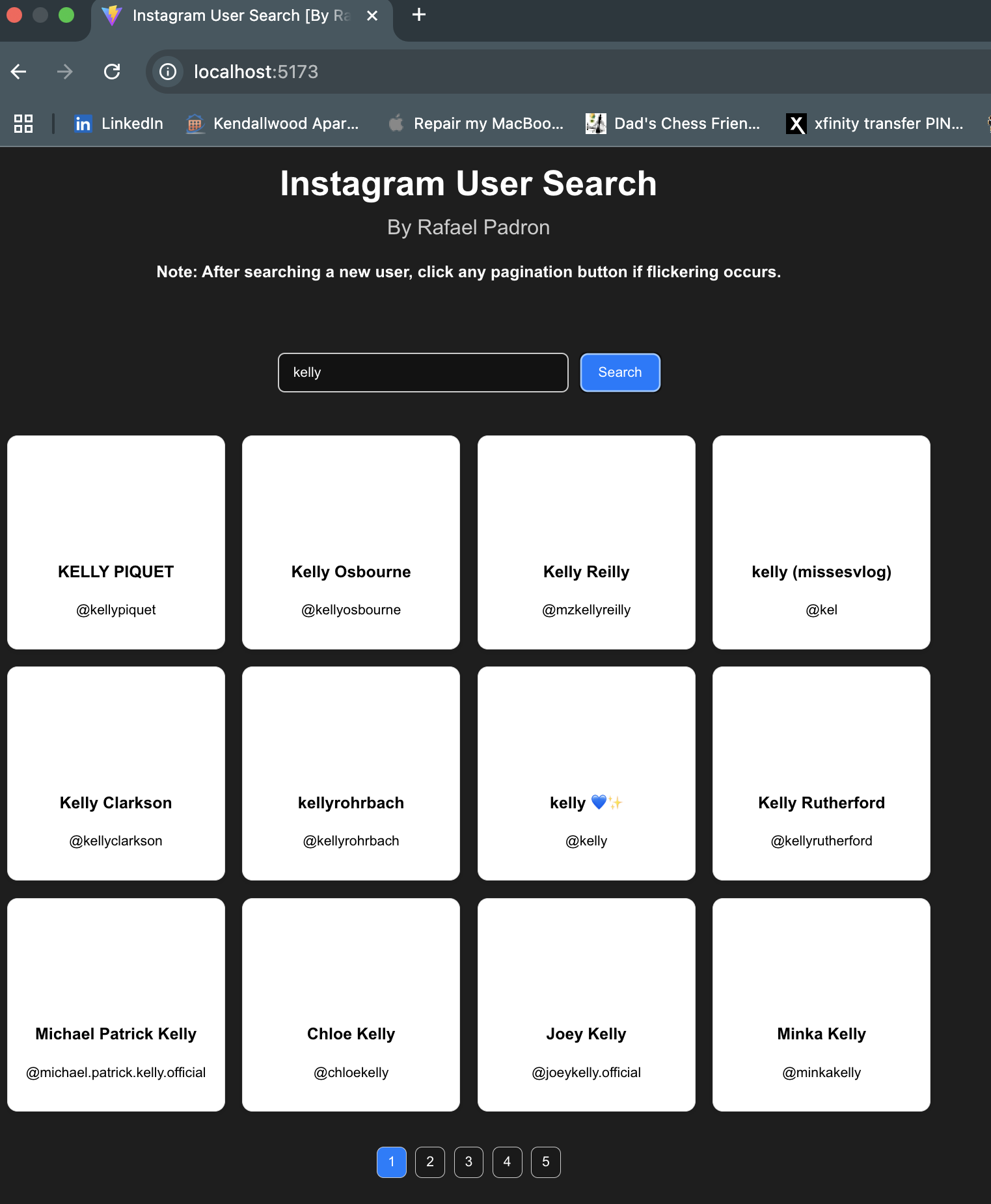
1. Type “kelly” in the TextField:



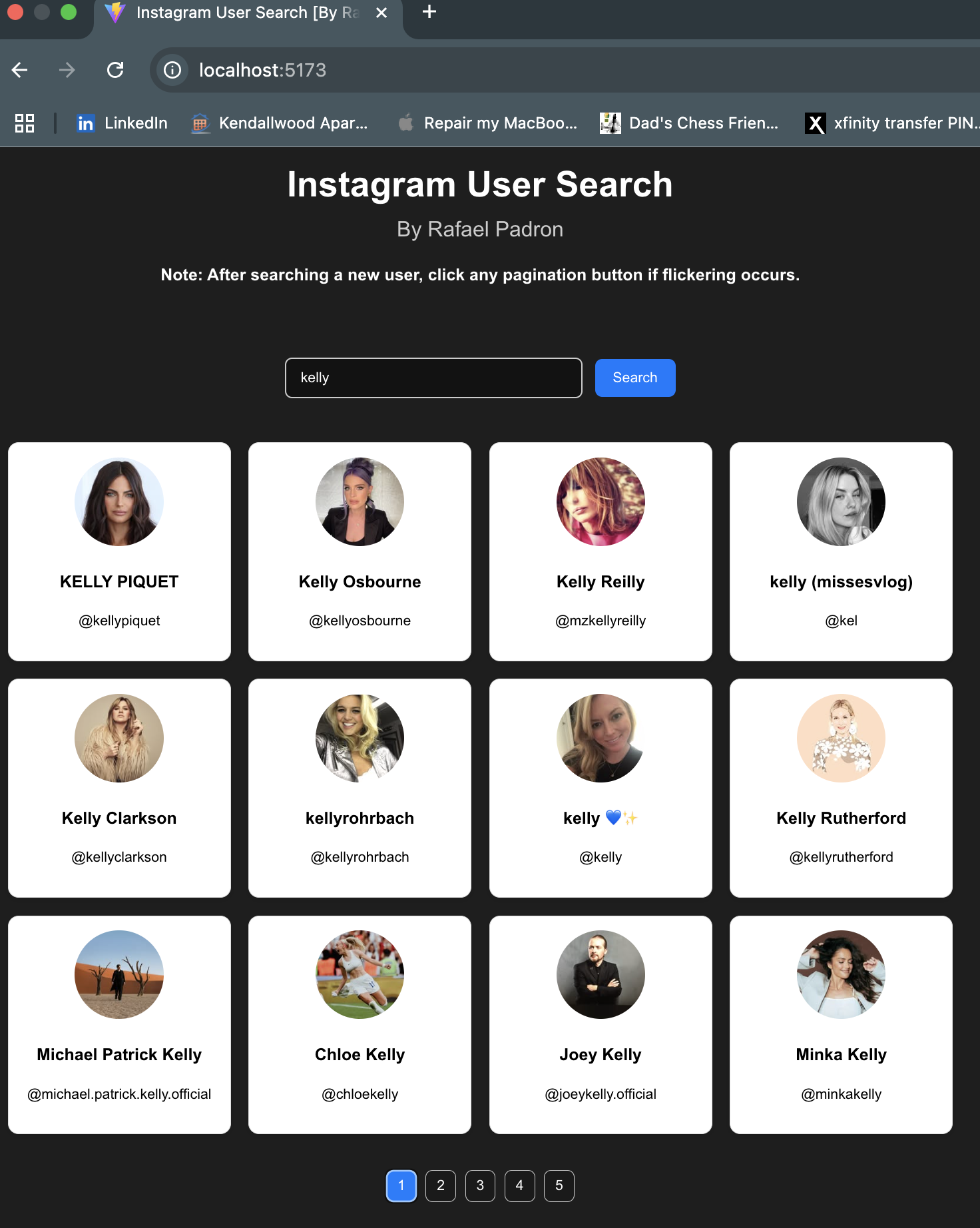
1. Click on the “Search” button:

**NOTE:**

**If flicking occurs, please click any of the pagination buttons (at the bottom).**



Either once clicked on the bottom pagination buttons or reentering the same “Instagram username again” thus we have:



**Sample response**

