11 - React UI for Search

In this, we develop a React frontend to enable search functionality for job posts, leveraging the searchByKeyword method from the backend.

Backend API Endpoint: The search functionality is exposed through a REST API endpoint defined in Java as:

```
@GetMapping("jobPosts/keyword/{keyword}")
public List<JobPost> searchByKeyword(@PathVariable("keyword") String keyword) {
   return service.search(keyword);
}
```

React Frontend Workflow: A text field in React triggers the search when a user enters a keyword. The component fetches job posts matching the keyword from the backend and displays them.

- **State and Initial Data Fetching:**
    - **useEffect** is used to load all job posts initially when the component mounts.
    - axios.get is used to make a GET request to http://localhost:8080/jobPosts.

```
useEffect(() => {
   const fetchInitialPosts = async () => {
      const response = await axios.get("http://localhost:8080/jobPosts");
      setPost(response.data);
   };
   fetchInitialPosts();
}, []);
```

- **Search Functionality:**
    - A text input captures the keyword and calls the backend searchByKeyword endpoint dynamically.
    - When the keyword is updated, another axios.get request is sent to the URL http://localhost:8080/jobPosts/keyword/{keyword} to retrieve matching posts.

Code Link :
https://github.com/navinreddy20/spring6-course/tree/c6690e4f2c70d8f530d70623f13d14ff0ffd7e7d/React%20UI