* The challenge is solved using HTML, CSS, JS, ES5/ES6, jQuery and Bootstrap4.
* It is responsive and follows the design specifications
* It is deployed and hosted in a publicly accessible site [https://user-searchbyemail.netlify.app](https://user-searchbyemail.netlify.app/)
* As no 'Access-Control-Allow-Origin' header is present on the requested resource “<https://ltv-data-api.herokuapp.com/api/v1/records.json?email=janesmith@example.com>, I prepended <https://cors-anywhere.herokuapp.com/> url to the above API endpoint

My Approach:

Designed two responsive views. One is the email search view with validation and the second view is the result screen, where the results are displayed.

In the first view, the email is validated and once the “GO” button is clicked, the view is directed to the result view bypassing the email value in the query parameter string.

Used header and footer templates in both screens. Also used ES6 templates with backticks for populating the “0 Results” view, “1 Result found” view.

Also made Ajax call on page load with async and await. “Loading image” appears on the result view, while waiting for the response. Once the response is received, the result view is populated accordingly. While in the result view, the search view can be accessed by clicking the “search icon” on the right side of the navigation menu.

I feel this is the temporary approach to overcome the CORS issue. I prepended cors-anywhere URL to the existing URL, but the requests are limited. I read that if the app is accessed more than the limited requests, then we have to set up a proxy server using node.js. I tried to deploy my code in the Heroku, but faced a problem. Only node.js apps can be deployed on Heroku.

I don’t have knowledge of Node.js, but I tried to implement my existing application to node js. But I am facing problems with jQuery code, the routes, etc. Its a long way to learn. I followed stack overflow,medium.com etc to make my code work in nodejs application, but it is not working.