

Software Engineering Fullstack Candidate Assignment

Before you begin, please reply to this email confirming you have received these instructions. Thank you!

We're excited for you to continue the interviewing process for the Software Engineering position here at BrightEdge!

You have **48 hours from the time this email is sent to complete the challenge.**

DEADLINE: <Deadline>

Deliverables: documentations, configurations, code, and a demo screenshot/video of the development result.

Let's begin!

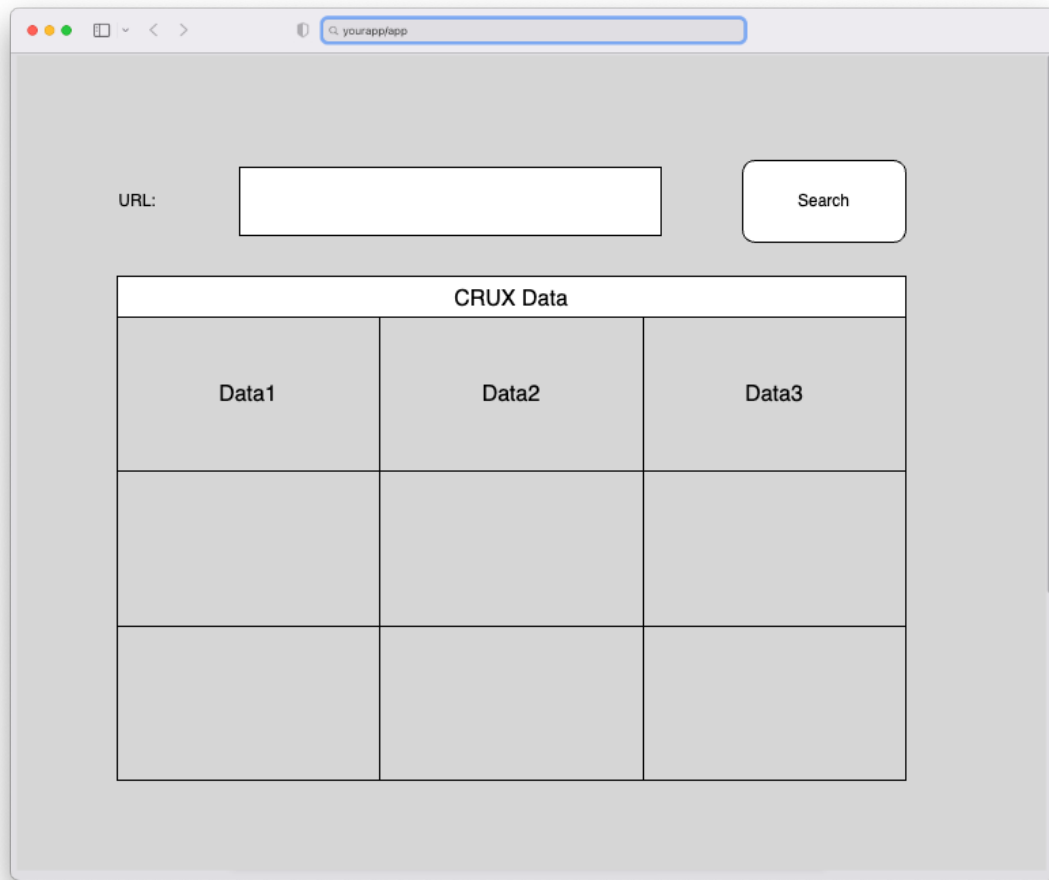
The speed to serve a URL in the browser is critical for the success for a website to be populate. BrightEdge assists our customer for find out any slowness in their site, and provide insight and recommendations to improve page speeds.

One of the common way to get performance data is using Chrome UX Report. This is a public API that we can get the data for a given url, and generate insights for our customers.

Assignment:

Part 1:

Create an app to get Chrome UX report for a given URL. Here's a wire frame of the application



Follow the steps from google to setup an account to access the data:

1. Setup to use the CrUX API to make POST/GET request for your application
 - a. [CrUX on BigQuery - Chrome for Developers](#)
 - b. [CrUX API - Chrome for Developers](#)
2. Create a React/Angular App, using Materials UIs (or equivalent) and python django or node.js backend
 - a. Have a Text input to allow input of URL
 - b. Have a Search Button
 - c. Have a data table to display the result of the data
3. User will fill the URL values, such as,
 - a. <https://developer.intuit.com>
 - b. hit search button
 - c. Display the result in the table

Part 2:

- Create a way to allow user to filter the values
- Support Sort
- Example, filter to only display certain data
- Example, filter to only display certain data above certain threshold

Part 3:

- Support multiple URLs. For example, allow input multiple URLs,
 - <https://developer.intuit.com>
 - <https://quickbooks.intuit.com>
 - <https://turbotax.intuit.com>
- In the data table, display a summary of the data gathered from the API, such as
 - average
 - sum

Output:

- zip up your code and upload to a downloadable location for us to download
- instruction how to configure and run on evaluator's machine
- demo video of how the app works, what are the features, any known issues
- documentations: design, issues, next step.

Bonus:

- Insight and recommendation of the data consumed
- deployment of the application

What we are evaluating from the assignment:

- Evaluate core functionality and exception handling (completeness of code)
- Good object-oriented development. Clean code.
- Ability to articulate how you built your solution (design document)
-

What's allowed: All open source libraries are allowed

What's not allowed: consult with jobs.questions@brightedge.com

FAQs

Q: Can I use external libraries for any of the features?

A: Yes, you can. A brief description why the choice of the libraries would be helpful to understand your decision making.

Q: May I have more time to complete the assignment?

A: This is a 48 hours challenge, please send in all the work that can be done within 48 hours or contact jobs.questions@brightedge.com

Q: May I only work on the assignment for 4 hours?

A: Yes, you can return all the work can be done and return within 48 hours.

Questions about the assignment: jobs.questions@brightedge.com