

Frontend Assessment - Student Profiles

The goal of this assessment is to replicate the frontend application below (as close as possible). You are allowed to use any frontend framework (React.js, Vue.js, etc.) or use plain Javascript, HTML, and CSS. **We recommend using a frontend framework as the later parts of the assessment will be very difficult without it.**

If you notice something is not working (like the API, or any of the links in this document), please contact hello@hatchways.io.

This assessment will be evaluated based on the following criteria:

- Correctness: Is your solution complete and does it pass different test cases?
- Code Organization, Readability, & Maintainability: Is your code easy to read and well organized?
- Code Performance: Is your code efficient? Did you use appropriate data structures?
- Best Practices: Did you utilize good programming practices (write unit tests, avoid anti-patterns)? Did you show a good grasp of your language/framework of choice?
- Completion speed: A fast completion time comparable to the completeness of your solution. This is the least important criteria.

We use the [following rubric](#) to evaluate your submission.

The assessment is broken into 5 parts to make it easier to complete.

You just need to submit one version with as much functionality as you can.

Public Repositories

Do not post your solution to a public repository. We understand that you may want to share projects you have worked on, but many hours go into developing our tools so we can provide a fair skills evaluation. Posting this assessment online impacts our ability to judge candidates fairly and equitably.

Part 1

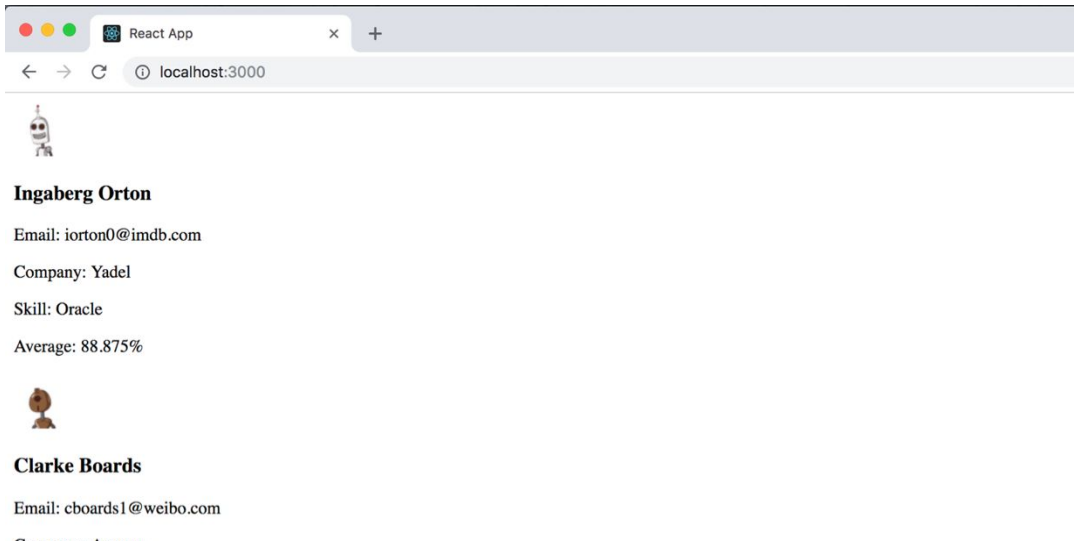
The first step of the assignment is to fetch data from this public JSON API, and present the information on the screen. The route to fetch the data is:

method: **GET**

url: <https://api.hatchways.io/assessment/students>

You do not need any credentials to access the URL above. It should also have CORS set up, so you should be able to access the data from the browser.

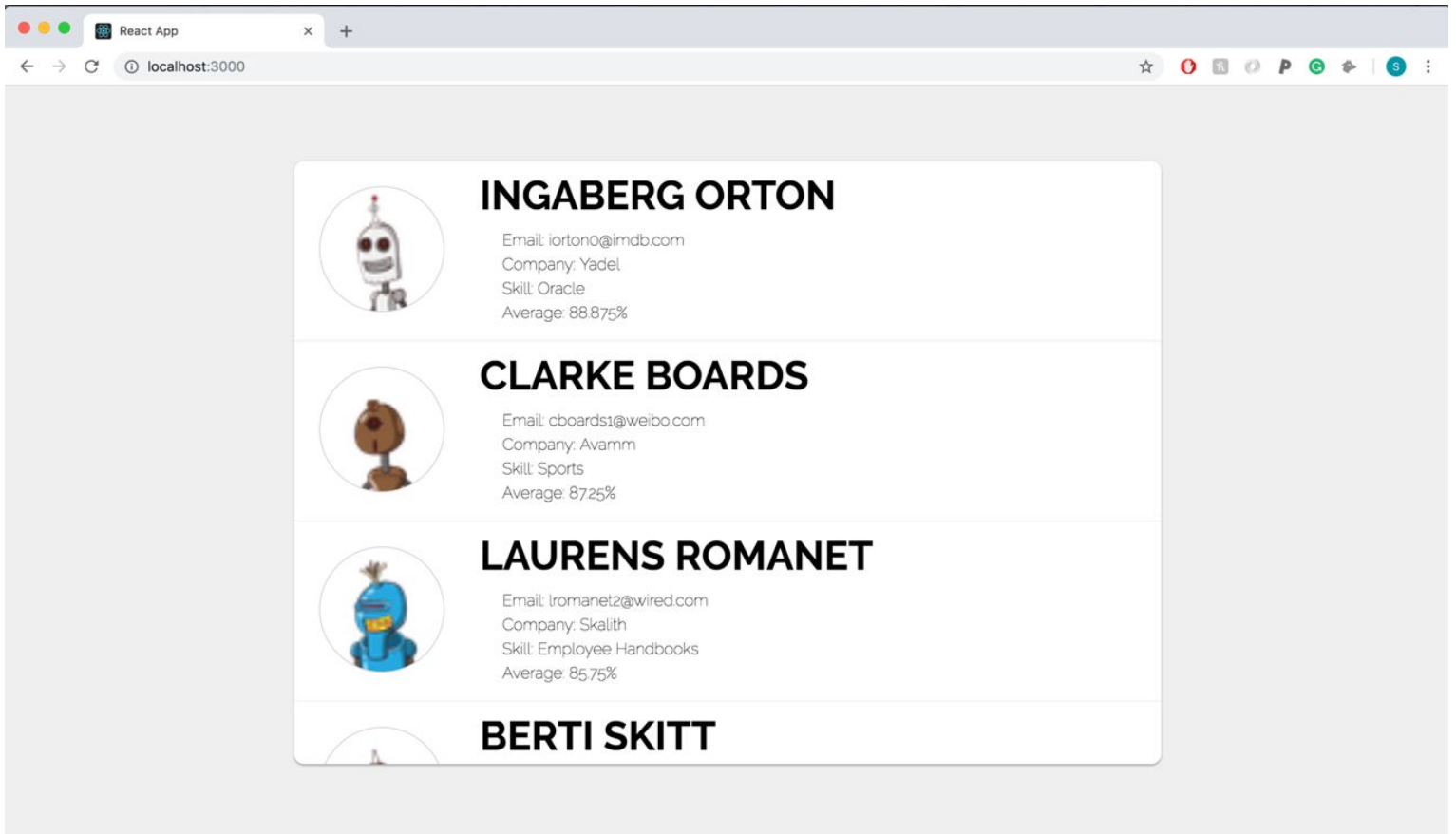
The goal of this part is to present the data on the screen. Do not worry about styling as you will be doing that in a future step.



Part 2


The second step of the assignment is to style the web page. Replicate the style of the image and gif below as best as you can. The font used in the image is [Raleway](#).

Below is a picture of what it should like, and [here is a link to a video](#) to show you how the scrolling should work.




Part 3


In this part, you are going to add a text input, where a user can filter the list of students by their name (both first and last name). Below are a few images of what this should look like. [Here is a video of the filtering in action.](#)




INGABERG ORTON
Email: iorton0@imdb.com
Company: Yadel
Skill: Oracle
Average: 88.875%



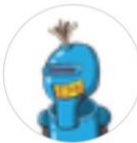
CLARKE BOARDS
Email: cboards1@weibo.com
Company: Avamm
Skill: Sports
Average: 87.25%




LAURENS ROMANET
Email: lromanet2@wired.com
Company: Skalith
Skill: Employee Handbooks
Average: 85.75%




REDT SKITT




LAURENS ROMANET
Email: lromanet2@wired.com
Company: Skalith
Skill: Employee Handbooks
Average: 85.75%



ROBBYN CORYNDON
Email: rcoryndon5@cargocollective.com
Company: Twinte
Skill: Cinema 4D
Average: 89.875%



RORY IBAN
Email: riban8@hubpages.com
Company: Fadeo
Skill: EE4
Average: 87.5%

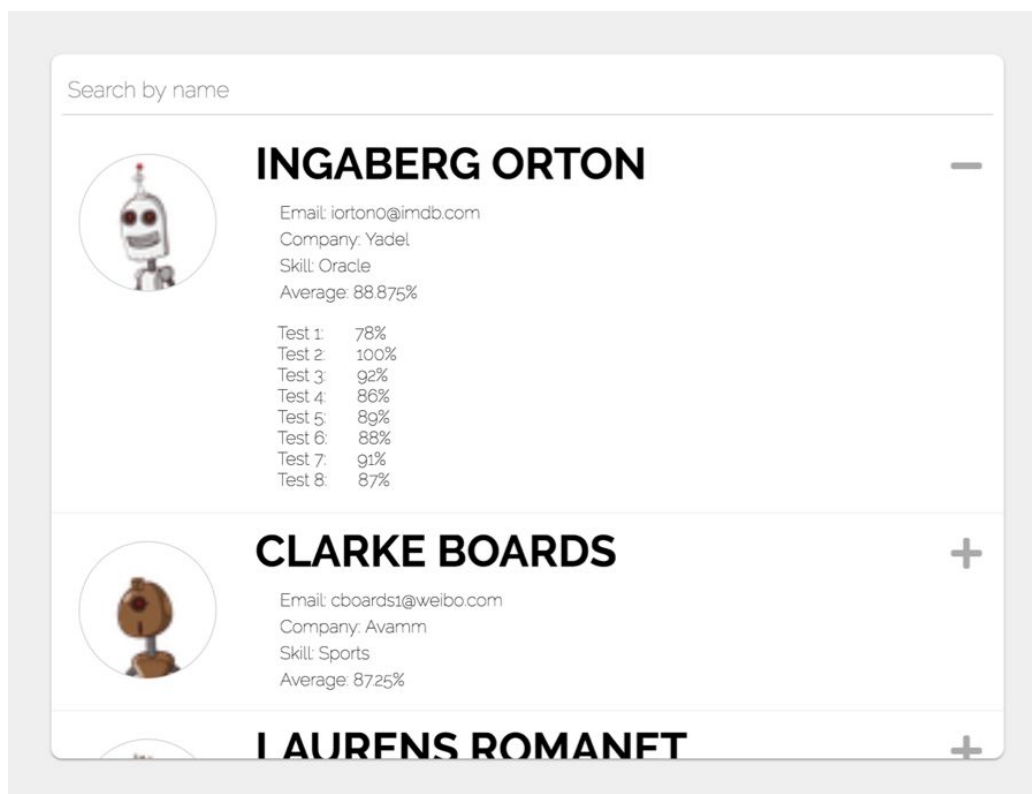
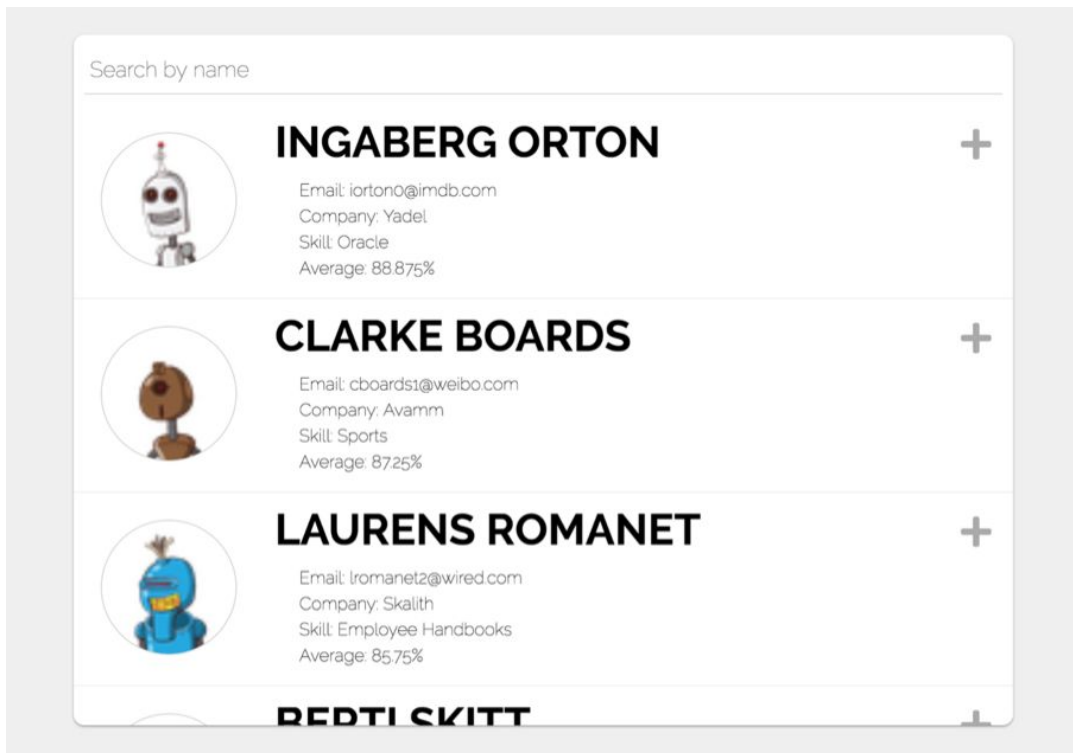


LENNA BOYBY

Part 4

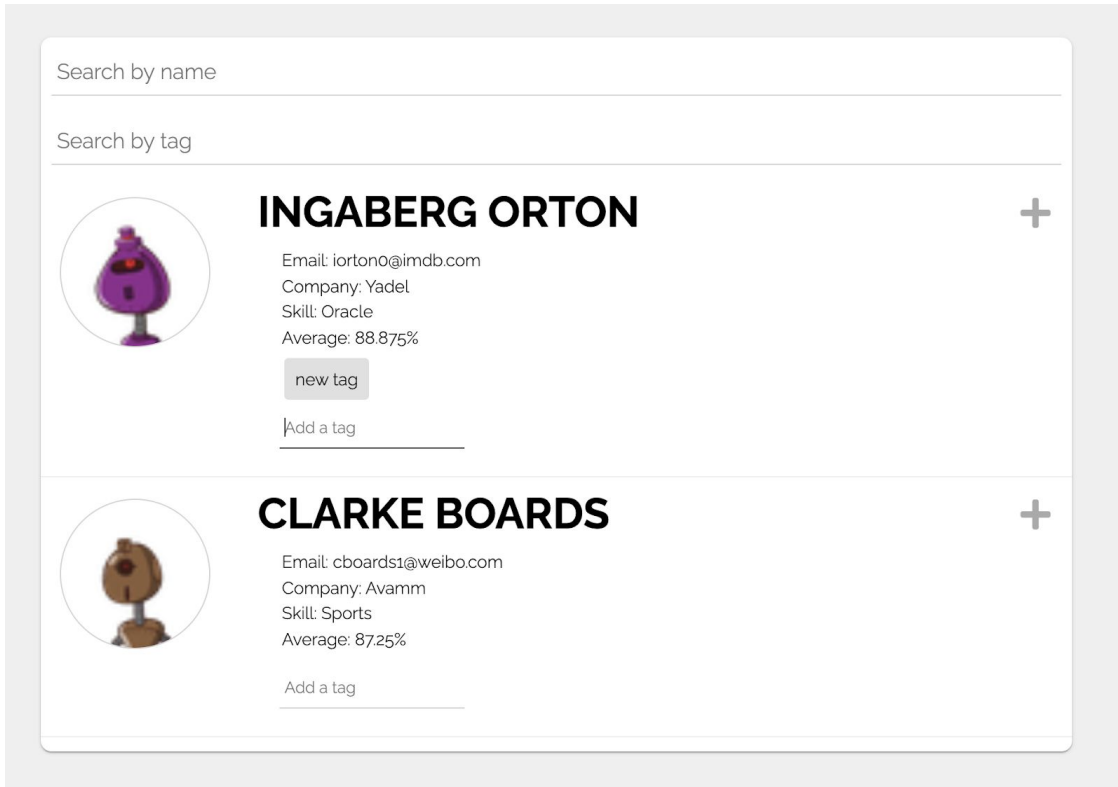
In this part, you are going to make each student have an expandable list view, so that all of their test scores may be viewed. See the images below for what the expanded view should look like. See the images below for what the view should look like when expanded. [Here is a video as well of what it should look like.](#)

Required: the plus should be an HTML button, and if the button is clicked on, the expansion will toggle



Part 5

Finally, you will add a text input field to add tags for a specific student. Here is what it looks like (adding a tag called “new tag”):



The screenshot displays a user interface for managing student profiles. At the top, there are two search bars: "Search by name" and "Search by tag". Below these, two student profiles are listed. Each profile consists of a circular avatar on the left, a plus sign on the right, and a block of text in the center containing the student's name, email, company, skill, and average score. Below the text is a "new tag" button and an "Add a tag" input field.

Student Name	Email	Company	Skill	Average
INGABERG ORTON	lorton0@imdb.com	Yadel	Oracle	88.875%
CLARKE BOARDS	cboards1@weibo.com	Avamm	Sports	87.25%

You will then add another search bar to search for students based on tags. A strong submission will have this functionality:

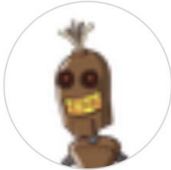
- If a name is entered in the search by name input, and a tag is entered in the search by tag input, results that include both the name and tag should be shown.

[Here is a video of the tag searching in action.](#)

The next page contains an example of filtering by tag and name at the same time as well as submission instructions.

ro

ta




LENNA ROXBY

+

Email: lroxby9@cam.ac.uk
Company: Yakidoo
Skill: LPS
Average: 82.875%

tag10

Add a tag



ROSALYND FITZALAN

+

Email: rfitzalana@parallels.com
Company: Photolist
Skill: Geography
Average: 90.25%

tag20

Add a tag

Submission Details

Please submit your code in a compressed folder on the [Hatchways platform](#). The max submission size is 5MB.

Upon clicking the submission button, you will see a form as pictured below. We need this information to be able to test your application.

1. Choose which language and technologies you used to develop your solution
2. Provide us with the **install command**, the **run command**, and the **port** that you used to run your application.
3. If you cannot find your commands in our suggestions, simply type your own and select "Use command".

More information on the next page ->

Submit Your Assessment Solution ✕

Upload a compressed folder (.zip, .sitx, .7z, .rar, and .gz) containing your code here:

Submitted file - test_assess.zip (0.000404 MB) ✕

Tell us how to run your program:

List the technologies/frameworks used (Required)

JavaScript ✕ React ✕ ✕ ⌵

Install command:

npm install ✕ ⌵

Run command:

This command runs your program ⌵

npm start

yarn start

Do not submit any built folders, since the compressed folder will be too large. **Do not submit your external dependencies (like the node_modules folder), since the compressed folder will be too large. We will be installing your dependencies before we run your code.**

If your submission is too big and you can't figure out how to compress, you are welcome to email your solution to hello@hatchways.io. Please include your name, and use the email you signed up with on the Hatchways platform. Use the subject line "Front-end Assessment Submission".