

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?

We use the [following rubric](#) to evaluate your submission. **Please note that if your submission does not attempt to complete all of the requirements, we will be unable to provide feedback on it.**

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

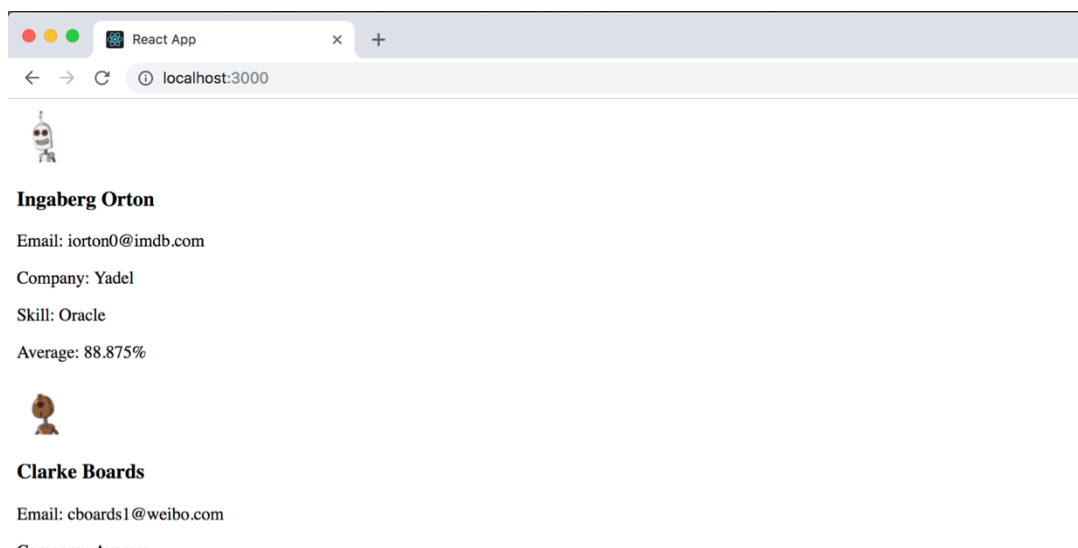
The first step of the assignment is to fetch data from this public JSON API, and present the information on the screen. The route to fetch the data is:

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.

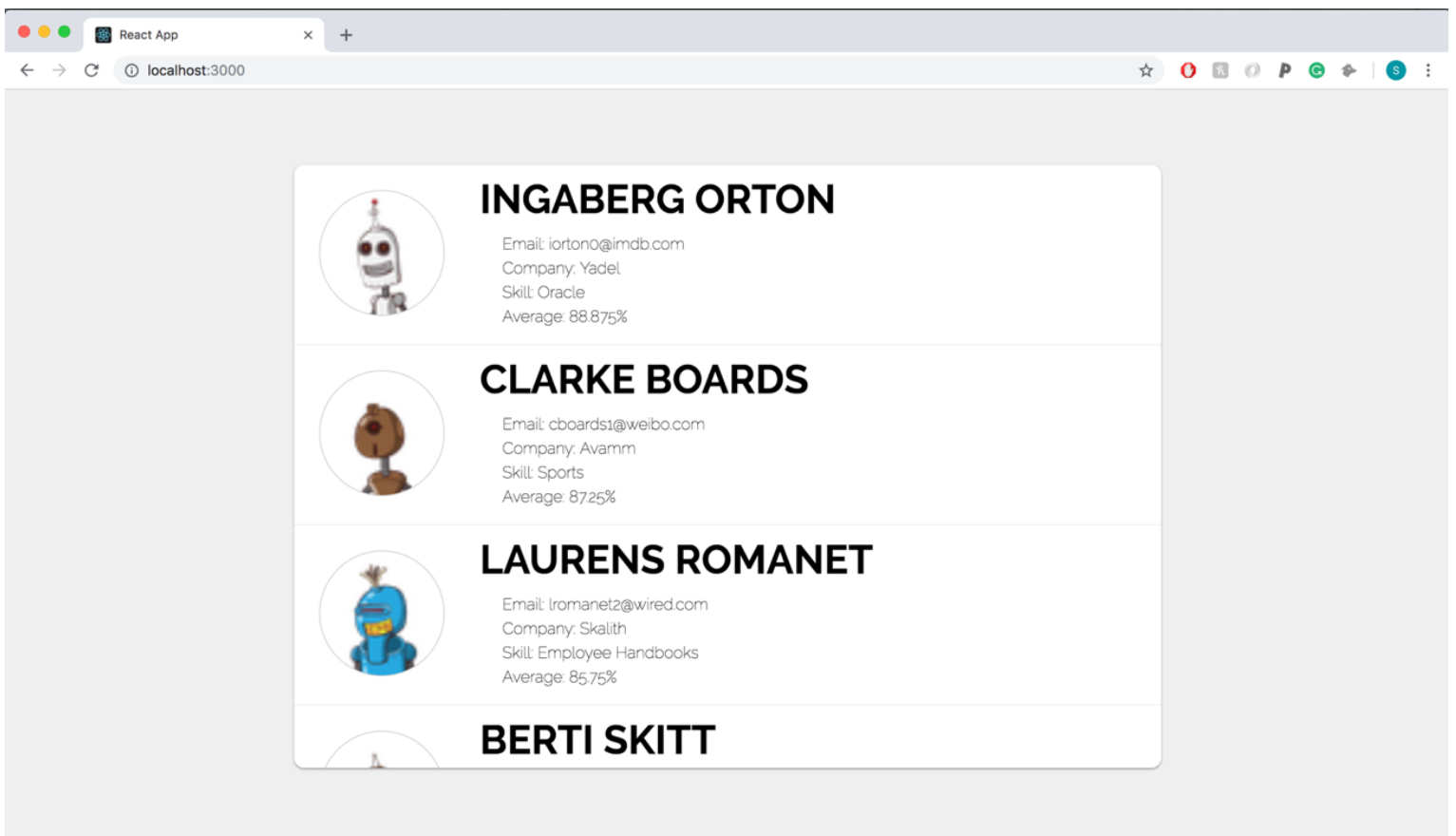
The goal of this part is to present the data on the screen. Do not worry about styling as you will be doing that in a future step.



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](#). Please also make the page mobile responsive, so that it works on screens of all sizes.

Below is a picture of what it should look 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 (including full 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%

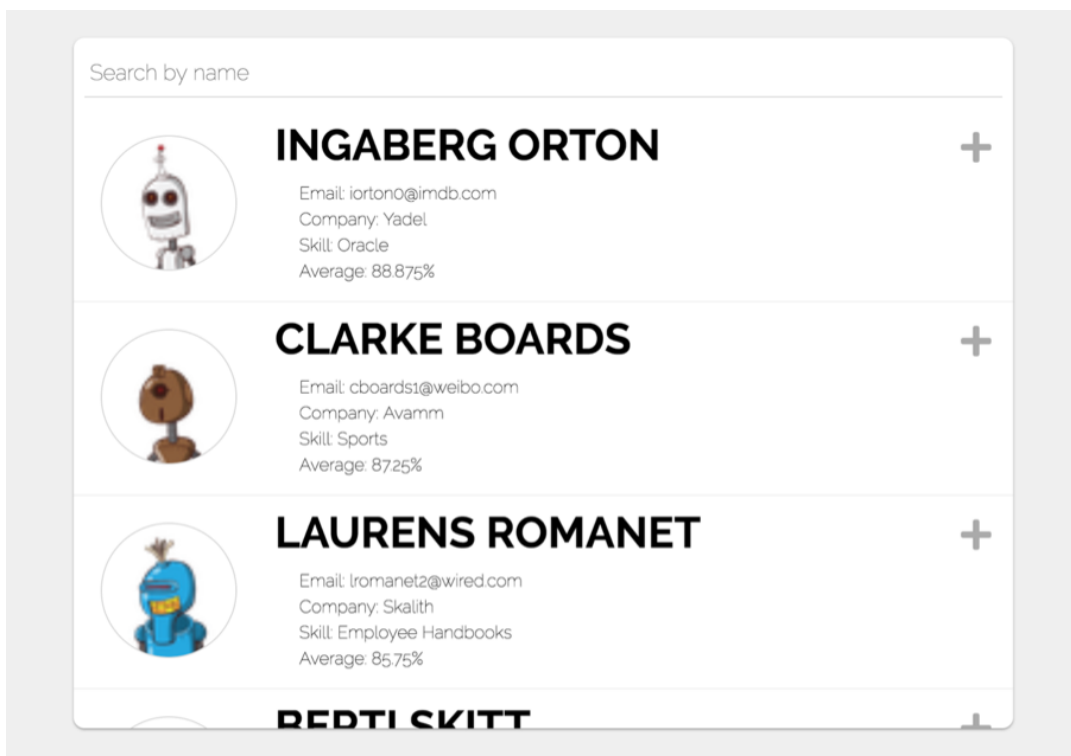


LENNA BOYBY


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



Search by name



INGABERG ORTON

Email: iorton0@imdb.com

Company: Yadel

Skill: Oracle

Average: 88.875%

Test 1: 78%

Test 2: 100%

Test 3: 92%

Test 4: 86%

Test 5: 89%

Test 6: 88%

Test 7: 91%

Test 8: 87%

CLARKE BOARDS

Email: cboards1@weibo.com

Company: Avamm

Skill: Sports

Average: 87.25%


L AURENS ROMANET

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”):

Search by name

Search by tag



INGABERG ORTON

Email: iorton0@imdb.com

Company: Yadel

Skill: Oracle

Average: 88.875%

new tag

Add a tag

CLARKE BOARDS

Email: cboards1@weibo.com

Company: Avamm

Skill: Sports

Average: 87.25%

Add a tag

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.

The screenshot shows a search interface with two input fields at the top: "ro" and "ta". Below these are two student profile cards. Each card features a circular profile picture of a robot head, the student's name in bold, their email, company, skill, and average score. A tag is displayed in a grey box, and there is an "Add a tag" input field with a plus icon to its right.

Name	Email	Company	Skill	Average	Tag
LENNA ROXBY	lroxby9@cam.ac.uk	Yakidoo	LPS	82.875%	tag10
ROSALYND FITZALAN	rfitzalana@parallels.com	Photolist	Geography	90.25%	tag20

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 additional technologies you used to develop your solution. Be sure to select the appropriate version for the language you have used.
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".

Please note that these commands will be used to run automated tests, so filling in every relevant field and providing accurate commands will allow you to receive feedback more quickly on your submission. If you have any notes to provide about your submission, please put them in a README, not in the submission form. Additionally, note that the install and run commands will be run from the root level of your submission, so please organize your files accordingly.

×

Submit Your Assessment Solution

* Indicates a required field

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

Submitted file - test_assessment.zip (0.223226 MB) ×

TELL US HOW TO RUN YOUR PROGRAM:

Language*

JavaScript (Node.js) ▾

Version*

12 (default) ▾

List any additional technologies/frameworks used (select or enter your own)

React × ▾

Install command

npm install × ▾

Run command

This command runs your program ▾

npm start

yarn start

npm run 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".