

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.

This assessment will be evaluated based on:

- Completion: Did you complete all the steps? Are there any bugs?
- Code Quality: Is your solution organized well? Does it follow the best practices of the framework you are using?
- Speed: We are looking for solutions that are completed within an appropriate time frame. If you can complete the whole assessment in a fast time, you will receive bonus points.

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

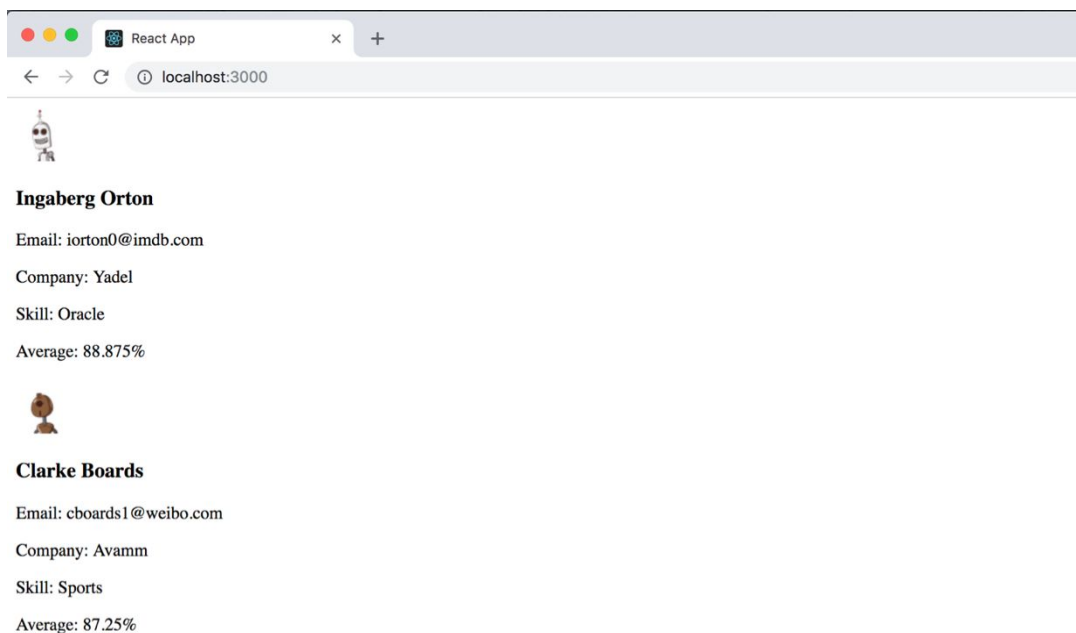
Part 1

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://www.hatchways.io/api/assessment/students>

You do not need any credentials to access the URL above. The goal of this part is to present the data as a list on the screen like below.

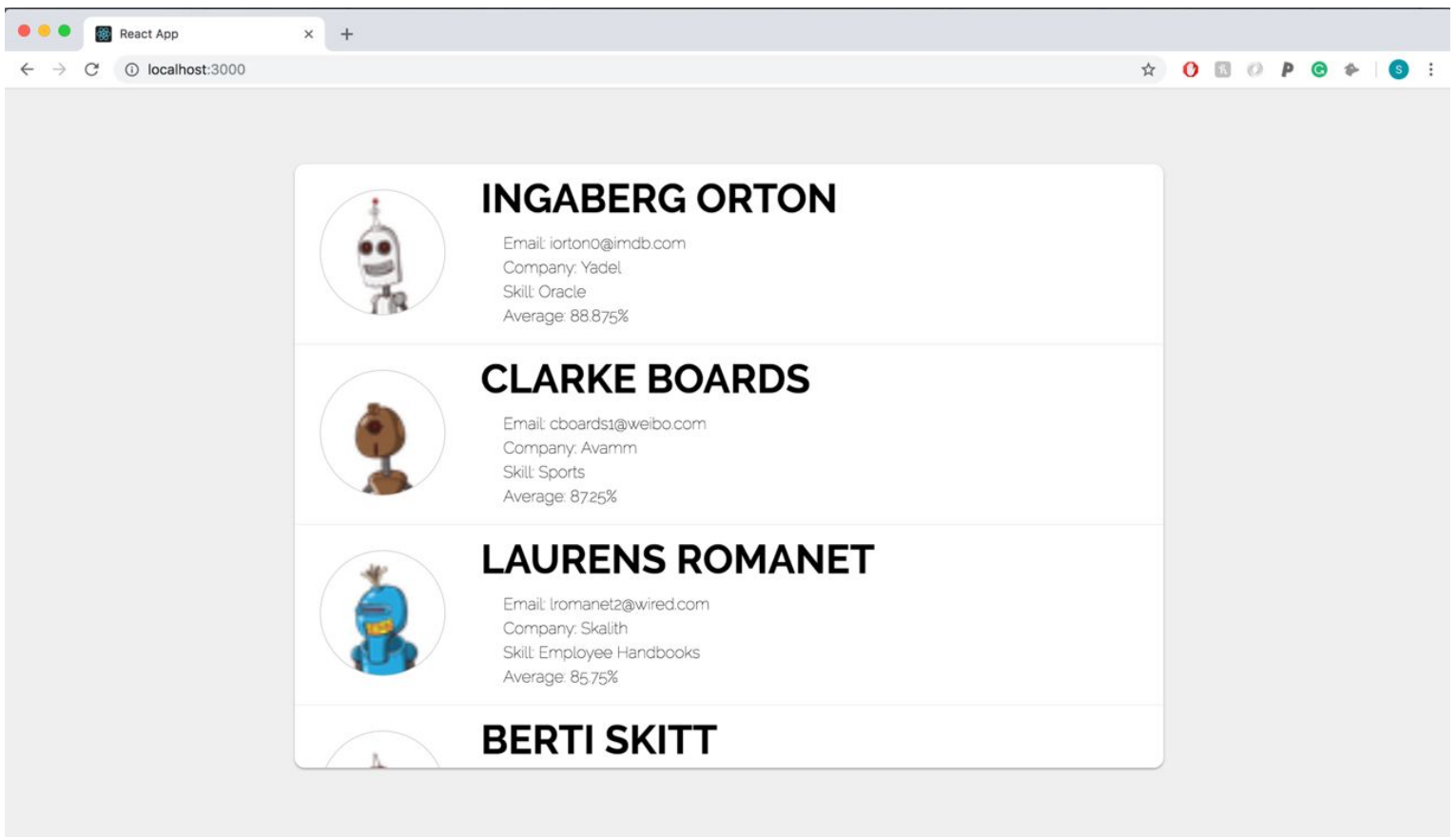


Part 2

The second step of the assignment is to style the student data in the previous part using CSS. Replicate the style of the image and gif below as best as you can.

If you are having difficulties replicating this part exactly and it is taking too long, move onto the next part. The future steps do not measure your ability to write CSS and HTML.

Below is a picture of what it should like and a [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). On the next page there are a few images of what it should look like. [Here is a video of the filtering in action.](#)

Search by name



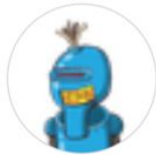
INGABERG ORTON

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



CLARKE BOARDS

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

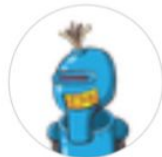


LAURENS ROMANET

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

REDTI SKITT

ro



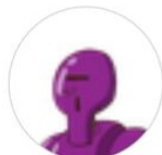
LAURENS ROMANET

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%



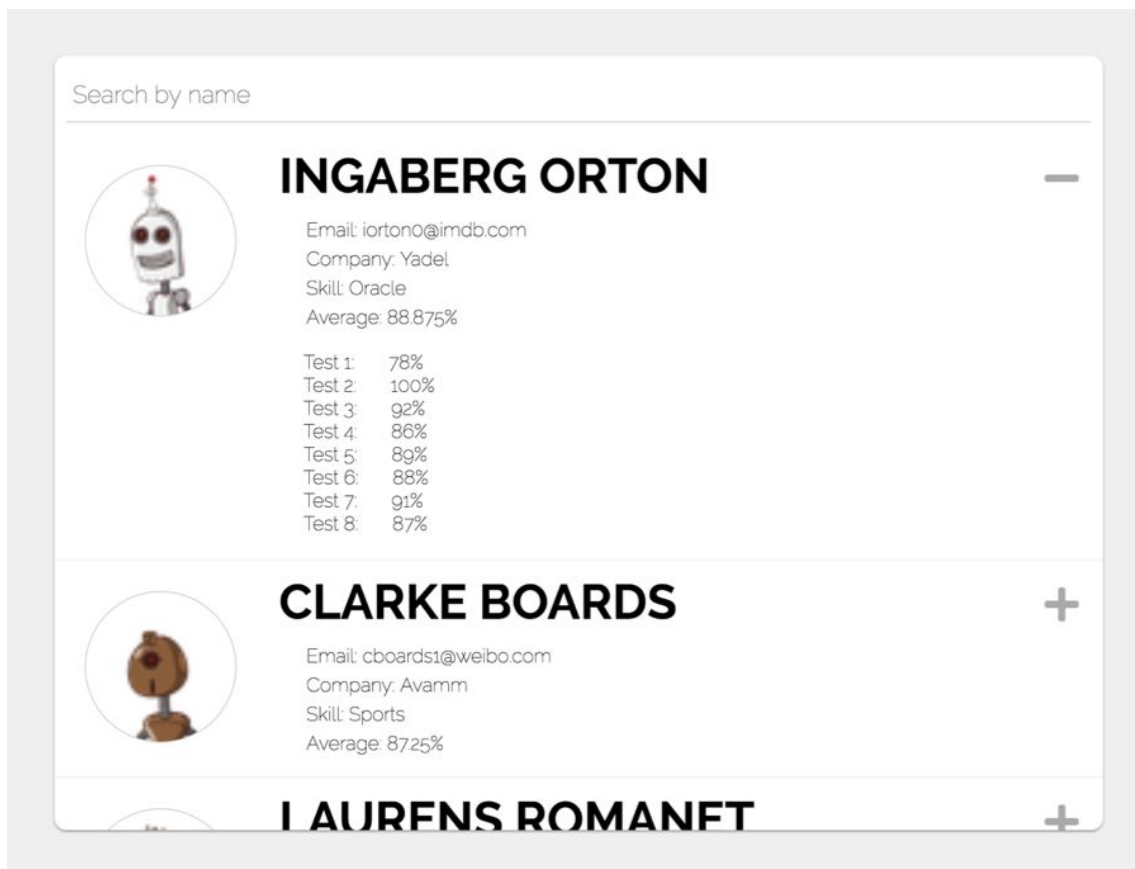
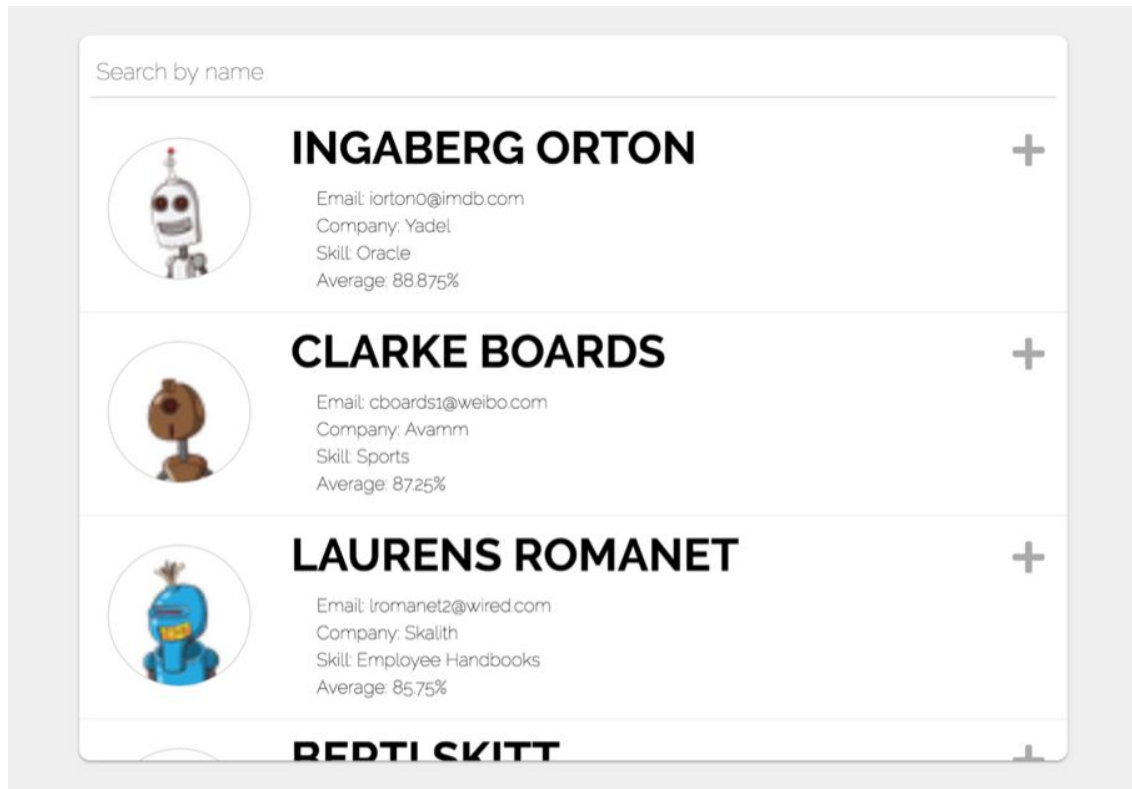
RORY IBAN

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

LENNA DOYRV

Part 4

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




Part 5

In this last part, you will add a text field in the expanded view of each student to add tags about the student. Here is what it looks like (added a tag called “new tag”):

Search by name

Search by tags



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%

new tag

Add a tag


CLARKE BOARDS

Secondly, you will add another search bar to search for students based on tags. A stronger submission will be able to filter both names and tags at the same time. [Here is a video of the tag searching in action.](#)

See the image on the next page for an example of filtering by tag:

Search by name

an|



LAURENS ROMANET

Email: lromanet2@wired.com

Company: Skalith

Skill: Employee Handbooks

Average: 85.75%

Test 1: 88%

Test 2: 90%

Test 3: 79%

Test 4: 82%

Test 5: 81%

Test 6: 99%

Test 7: 94%

Test 8: 73%

another tag

Add a tag

Submission Details

Please submit your code in a compressed folder (.zip, .sitx, .7z, .rar, and .gz) on the [Hatchways platform](#). The max submission size is 5MB.

Please include a Readme.md that describes how to run the application. **Do not submit your external dependencies (like the node_modules folder), since the compressed folder will be too large.** If you are using plain Javascript, make the index.html file the entry point to your application. You are free to use any scaffolding tools to help you get started.

If your submission is too big (although this should not be the case if you follow the steps above), 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 the Hatchways platform. Use the subject line "Frontend Assessment Submission".

Public Repositories

Please avoid posting 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.

If you would like to keep a similar version of the assessment on a public repository to showcase your skills, then please do the following:

- Remove all references to Hatchways in the assessment
- Do not use any Hatchways APIs (replace them with an API of your own)