Design and Development Department

Recruitment test

Development

Hello!

Welcome to the Design & Development Department’s recruitment process. To solve this challenge, you will have to make use of your React.js skills. These exercises are not intended to be used outside of this test.

React.js Assessment

Welcome to Condor's React.js Assessment challenge!

To begin the challenge, please enter our Github Recruitment Condor Agency and fork the RecruitmentCondorAgency/Challenge-React.js repository. Once finished, issue a pull request to our repo so we can review your solution.

This challenge requires:

- NodeJS 16.13.2
- NPM 8.3.2 or Yarn
- A cup of ☕ or 🍵

The repo is pre-configured, so just by running npm i or yarn, you should be able to install these basic packages:

- React Router + Axios: To fetch data and navigate through the site.
- Parcel + React.js: To serve and build a simple front-end.
- Json Server: To serve and expose a basic json db.

In the main package.json you'll have access to the following scripts:

- npm run app: To run the react app.
- npm run server: To start the local db.

You may use any other package of your choice to enrich your development experience.

The challenge

Tasks are not ordered or weighted! Before coding, read all the items carefully and prioritize as you consider.

During the SOW meeting, our customer voiced these requests:

- We need a simple website for our students.
- They should be able to select the university they are interested in and add it to a list to later compare their country's data.
- Nobody likes to study on a rainy day.

In response to our client's request, we made a simple Figma mockup that we need you to recreate. The images can be found in the ./graphics folder with the name of the required screen and width, like search. Additionally, you may use the basic prototype for reference.

Since this is the only wireframe, feel free to use any ui lib or style you like.

Do note that responsiveness has to perform correctly. We won't be evaluating design quality, but responsiveness-related-code quality.

Finally, if any design element does not comply with best practices, we expect you to take care of fixing it.

The task

Using react and the APIs referenced in the resources section, create a simple web app consisting of four screens:

1. Login: A login/register screen
2. Search: A universities search screen, with results
3. University country details

Aside from the screens, the app has to comply with the following specs:

- Search values have to be autocompleted with the correct (or close-to-correct) university name.
- A menu to go to the search page, the user profile, and logout.
- User data must include at least an email, password, and a universities list.
  Use Json-server for this. Do not worry about hashing or security as this is only a basic local db. We just want to see how you handle the data.

Personal user data (like the email) may or may not be shown depending on your preference, but a way for the user to identify that he/she is logged in would be appreciated.

What we'll be looking at

- Clean, formatted, readable, KISS, DRY code
- Commits structures and information
- Componentization and reusability
- State management
- API implementation
- UI-Logic detachment
- Error handling
- Design responsiveness
- Coding following industry best practices
- UX patterns

Submission Guidelines

- Please fork the repo to start.
- Feel free to add or replace any package. The only one that is mandatory, is React.js.
- Once finished, issue a pull request to our repo so we can review your solution.

Resources

- React
- React Router
- Axios: Fetch client
- Parcel
- JSON Server
- Hipolabs Universities API: An API and JSON list with domains, names, and countries of most of the universities of the world.
- RestCountries: Rest API with countries' data like latitude and longitude.
- 7Timer: Weather data.