Assessment: React 2

Download code <../react-2.zip>

> **Warning: Assessments**
>
> Remember, assessments are meant to be completed **by you**, not as a shared exercise with friends or other members of your cohort.
>
> All code submitted should be **written by you**. If you incorporate code from elsewhere, it must be clearly specified.
>
> **Use a private GitHub repo to submit this assignment. Add your mentor as a collaborator. Please do not** put your assessment on public GitHub repo.

Part 1: Conceptual

Answer the following questions inside the ***conceptual.md*** file.

Part 2: Unroll

Write a function called ***unroll***, which takes in a square array of arrays (i.e. a grid with *n* rows and *n* columns). An input could look like this:

```
const square = [
  [1, 2, 3, 4],
  [5, 6, 7, 8],
  [9, 10, 11, 12],
  [13, 14, 15, 16]
];
```

unroll should take in such a square array and return a single array containing the values in the square. You should obtain the values by traversing the square in a spiral: from the top-left corner, move all the way to the right, then all the way down, then all the way to the left, then all the way up, and repeat.

For the above example, ***unroll*** should return `[1, 2, 3, 4, 8, 12, 16, 15, 14, 13, 9, 5, 6, 7, 11, 10]`.

Here's another example:

```
const smallerSquare = [
  ["a", "b", "c"],
  ["d", "e", "f"],
  ["g", "h", "i"]
];
```

```
unroll(smallerSquare); // ["a", "b", "c", "f", "i", "h", "g", "d", "e"]
```

Write tests for these cases and make sure your code passes these. Feel free to add any other tests you deem necessary.

Part 3: Build an App!

Until a week ago, you were the lead developer at Silicon Valley's hottest coding cafe, "The Closure Cafe". You had a nice React app that listed your food items drawing those from a static list of items hard-coded into your app.

Now, though, the cafe has gotten its liquor license and relaunched itself as "Snack or Booze", and your app needs to change!

- The homepage should show the # of food items ("snacks") and drink choices ("drinks")
- The navbar should add a new link, "Drinks", leading to a page listing the drinks — just like the "Snacks" link leads to a page listing food items.
- The drink listing page should show a list of drink choices, with each being a link to the details about the drink, just like for food items.
- **However**, you shouldn't solve this by just cloning the ***FoodMenu*** and ***FoodItem*** components — you'd have so much duplicate code! Instead, turn these into generic components that can work with either food or drink lists/items.
- Now that you have more things on your menu, you should add a page that lets site users add either a drink or a snack.
- **Design your app well.** We don't want to see poor names or see AJAX calls buried in your components. Use good design! Test whatever you can!
- Make sure you **document your code appropriately. Srsly.**
- Make sure you handle not-found pages — if a visitor tries to go to a link that doesn't work, it should give a friendly 404-style message. If someone tries to go to a drink or food item that doesn't exist, it should redirect to the drinks or food items listing page.

The app does not need any kind of login or authentication; everyone can see everything and can add new items.

Starting Up

Run `npm i` to install the dependencies and `npm start` to start the server.

Important! Although the React frontend doesn't yet use it, you do have your backend set up so that `npm start` starts both your backend *and* your frontend. The backend is available at ***http://localhost:5000/***. You have a RESTful JSON-oriented backend with two resources, ***drinks*** and ***snacks***. (The backend supports all sorts of methods, but you will only need ***GET*** and ***POST*** for the exercise).

Have fun and good luck!

> **Note: Our Backend Server**
>
> You don't need to understand how our backend works in order to work on this assessment — but you might find it helpful for your tech toolbox.
>
> We included a node library, ***json-server***, which creates a straightforward fully-featured JSON REST API from a JSON file, including the ability to update that JSON file when changes are made via POST/PATCH/DELETE. When you add drinks/snacks, you're changing the file ***db.json*** in the application directory.
>
> ***json-server*** is terrific for working on front-end code challenges that would benefit from a simple backend API, or small-scale personal projects where a custom backend server would be overkill.
>
> Also: we edited the ***package.json*** so that `npm start` doesn't just start up the React server on 3000, but also ***json-server*** on 5000. This is a nice convenience—to work on the app, you don't need to separately start/stop both. We did this with another add-on library, ***concurrently***.

Solution

View our Solution <solution/index.html>