Senior Enrichment Project

Make a thing!

Getting started

1. Fork and clone this repo
2. *Set the name of your project in `package.json`*. The skeleton intentionally ships with an invalid name.
3. `npm install`
4. Check out the mock-view in the `wireframes` folder
5. Start the build process with: `npm run build-watch`
6. In another terminal, start your app with `npm start`
7. If you navigate to the URL you should see some UI already :) [We already have some connection code to get you started]

Requirements

The Premise

You are the CTO of the Margaret Hamilton Interplanetary Academy of JavaScript. Create a RESTful web platform that allows you to manage your students and campuses.

The tools

Use at least sequelize, express, react, and redux when creating this app. You can incorporate any additional libraries or tools you wish.

DB Design

- Students
  * have profile info (e.g. name and email)
  * must be assigned to a campus

- Campuses
  * have info such as a name and image
  * can have many students assigned (may have none)

Views and Functionality
- See Wireframes folder for visual

- Navigation: as a user I...
  * will land on **Home** by default
  * can navigate to **Campuses** from **Home**
  * can navigate to **Students** from **Home**
  * can navigate to view a **Single Campus** from **Campuses**
  * can navigate to view a **Single Student** from **Students**
  * can navigate to view a **Single Student** from **Single Campus** (for any student at that campus)
  * can navigate to view that student's **Single Campus** from **Single Student**

- Views: as a user I...
  * see a list of all campuses on the **Campuses** view

  * see a list of all students on the **Students** view
  * see details about a campus on the **Single Campus** view, including that campus's students
  * see details about a student on the **Single Student** view, including that student's campus

- Actions: as a user I...
  * can create a campus
  * can edit a campus's info, including adding/removing a student to/from that campus
  * can delete a campus
  * can create a student
  * can edit a student's info, including the campus that student is assigned to
  * can delete a student

Routes

```
GET
- all campuses
- a campus by id
- all students
- a student by id
```

```
POST
- new campus
- new student
```

```
PUT
- updated student info for one student
- updated campus info for one campus
```

```
DELETE
- a campus
- a student
```

How to test functionality without a frontend
- GET: use your browser
- POST / PUT / DELETE :
 - CLI (command line interface) with `curl`
   - e.g. `curl -H "Content-Type: application/json" -X POST -d '{"username":"kate","password":"1234"}' http://localhost:3000/api/login`
   - `-H`: headers. `-X`: verb. `-d`: data (must be of the type specified in headers). http://[address]:[port]/[route_path]
 - [Postman](https://www.getpostman.com/)
   ![](https://www.dropbox.com/s/4fk3b90cd0i1a5y/postman_post.png?raw=true)
- Databases: use Sequelize in your routes and see if you are receiving what you expect

Video Walkthrough

Please submit a 3-minute screencast of a walk-through of the functionality *and code* for each user story in your app. E.g. for "As a user, I can create a campus", please show us that you can successfully create a campus in your app, and also the actual code that is involved in doing that (from the front-end components to the backend routes and models). We recommend using Quicktime to record the screencast (instructions on how to do that [here](https://support.apple.com/kb/PH5882?locale=en_US&viewlocale=en_US)).

Once you've recorded your screencast, please *upload it to YouTube as an unlisted video*, and send us the link. This will aid us in evaluating your submission.

Evaluation

- Code modularity/readability (25%)
- Models (25%)
- Routes (25%)
- Frontend logic and functionality (25%)
- Design + Bonus features (up to 10 Extra Credit points)