**Project #6**: PERN Full Stack Application

(PostgreSQL, Express, React, Node)

**Start:** 4/12/2018

**Mid-Check:** 4/20/2018

**Version 1.0 Due:** 4/25/18 - Blue Badge Presentation

**Final Due:** End of Track

Submission Guidelines:

Client & Server should be deployed to Heroku with a cloud instance of Postgres.

Pre-Reqs

* React basics (prop, state, components, etc.).
* Go through guided React lessons.
* Built an application with a SQL database.
* Built an application with Node.
* Understanding between a server - client relationship.
* Created API endpoints, and connect to them with a client.

Objective:

The objective for this project is to build a full stack application using PostgreSQL, Express, React and Node.

Requirements:

* The server must be written using **Node**.
* Have AT LEAST 3 tables created for your database. One of those tables must be a Users table that stores user information.
* The server must support session validation with **JWT**.
* The server must encrypt sensitive material such as passwords.
* There needs to be AT LEAST 3 API endpoints. Of which one of them must implement full CRUD (Create, Read, Update, Delete) functionality.
* The client must be written in **React**.
* Add custom styling to all views in the Client.
* Add **validation** to user sign up, ie an email address should look like this ’[test@test.com](mailto:test@test.com)’, passwords should have 5 or more characters.
* The client side must implement react-router.

Demo**:**

You will demo this app to a small group in a fun and informal way. This presentation will not be a fully dressed presentation to the entire class. Then, the small groups will choose applications to be shown off to the class as a whole.

Check-ins:

You will have daily stand ups regarding your active development on this application. You must be prepared at all times to give updates both seen and verbal about your application.

Escalation:

Please follow the escalation guide that is in place, [here](https://docs.google.com/document/d/1vfG6Ltltajnglqg3E8e3gks2S8K5C80LU9yIl0wacr0/edit).

How to Proceed:

It’s really easy to have big ideas of grandeur, in fact we encourage that. We do have a caveat of figuring out what is version 1.0 of the application.

Start with thinking through the application. Without doing this you will lose time in the long run, trust us. So let’s have an example of a hypothetical idea, an application for a Farmers Market.

Now that we have our idea, we now think through some questions:

**Q**: **Who will this application be for? (Who will be our users?)**

**A**: *This application will be specifically for the Farmers Market vendors. Not for people who go to farmers markets, just the vendors. (****Notice that it’s very specific****)*

**Q**: **What is the purpose of this application?**

**A**: *The application is for Farmers Market vendors to submit their inventory to the organization that runs the market.*

**Q**: **What will this application give users the ability to do?**

**A**: *Vendors can log on. They can add their inventory. This app will also allow the organization to manage the vendors at the Farmers Market - creating, updating, and deleting them.*

Now that we have our idea rounded out a bit, let’s start thinking about how we will store the data:

***Thought process***:

I would like to have 3 tables, Users, Vendors, VendorsInventory.

Users table needs:

* firstName
  + Cannot be null
  + Needs at least 1 character
  + Needs to be a string
* lastName
  + Cannot be null
  + Needs at least 1 character
  + Needs to be a string
* email
  + Cannot be null
  + Needs to be a email
* password
  + Cannot be null
  + Needs at least 6 characters
  + Needs to be a string
  + Needs to be hashed

Continue for the rest of the tables.

This gives you an idea of how to contain your development. If you ever go off track you can ask yourself “*is this a part of my original goal?”*  If not, then take a step back and get back on track.

Have Fun

Don’t lose sight of the 50,000 ft. view of why you are doing what you’re doing. You’re learning, it’s going to be messy, but that’s all a part of the fun.