Springboard Software Engineering Capstone Proposal

App Name: ProPulse

Tech Stack: React, Node, Express, PostgreSQL

Focus: I will be creating a full-stack application, writing my own backend and frontend.

Type: My application will be a website, but my goal is to make it mobile-friendly

Goal:

The goal of this application is to provide a centralized communications hub with construction projects, specifically, being the inspiration. However, the app may have potential for other industries with similar personnel chains/structures as well.

I want to create a platform that will provide a structured, yet **accessible** means to communicate deadlines, design conflicts, safety concerns, progress updates, and more across contractor teams and other construction personnel.

Accessibility is a huge part of this application. In my experience, many construction workers are not very tech savvy (*you wouldn't believe how many times I've seen peers "reply all" when they **really** shouldn't*), but most can use Facebook or X. There exist applications for jobsite communication, but those I have seen are far more complex than they need to be, leading to slow, confusing, and laborious interaction that prohibits the tech-layman from benefiting from their use.

I envision an app where a single admin, most likely a project manager, can create a thread which is isolated to a specific project. From there, they can invite relevant participants to the thread, such as the other project managers, the foremen for each trade, their crew personnel, and whoever else they would like. Now, if there is a design issue, safety concern, or any important bulletin, a participant can post a message in the project thread. They can tag relevant people if they like, set a priority for the post, even set a deadline for a response. The app will send notifications to any or all participants, depending on who was tagged, as well as send reminders automatically as deadlines approach. All posts will be tracked with a status, and remain in the thread history until project completion so they can be referenced at any point, even when marked complete. These are just ***most*** of the features I plan to implement.

In summary, I picture something like Twitter/X, but private threads for controlled groups of participants, with project-oriented functionality.

Demographic: Anyone who manages/works in project-oriented industries.

Data:

I plan to create my own API for this application. It will handle user information, post information, and project/thread information. The data tables will resemble what one would expect to see behind any social media application.

Approach:

I plan to create the backend API first. The database schema should be close to the following:

USERS
email PK
first_name
last_name
phone
role
password
posts
tagged_in_posts
projects

POSTS
id PK
date_posted
posted_by
project_id
users_tagged
deadline
status
priority
content
images

JOBS
id PK
admin
participants
posts
street
city
state

Once I have finished the backend, and tested it thoroughly, I will proceed to build the frontend accordingly.

Potential Issues:

The potential issues/problems I could run into with my API will come mostly from building the database schema. Trying to foresee the necessary columns for each table will be tricky. It could turn out that, mid-development, I realize that I need more/less columns, different primary or foreign keys, or maybe different data formats.

Handling Data:

The only sensitive data that definitely needs to be secured in my app would be passwords. Other than that, the app only requires fairly common data such as email addresses or phone numbers. In the context of the app, this data would be shared to other users as it enables better communication. Also, the app will require this data in order to send email/text alerts to users.

Functionality:

I have listed the key features of the app above, in the **Goal** section. I believe these features will make for a fairly robust example of what I have learned throughout the bootcamp.

User Flow:

The user flow will look something like the following:

- Admin creates a user account
- Admin then creates a “project”
- Admin invites participants via email/text
- Participants create user account then accept invite to the project
- Users see their dashboard upon login, which displays the projects they are participating in, their alerts for posts they are tagged in, updates on their posts, etc.
- User can click on a project, which will bring them to the thread
- Here they can see all posts, or filter posts by different criteria
- They can click on a post to expand the reply history

More Than CRUD?

I think the one feature that makes this app more than a CRUD app will be the notification feature. Notifications are a linchpin feature here. They will assist with the overall goal of the app, which is communication. The data will be the driving force for these notifications, making the data more than just static information.

Stretch Goals:

Oddly enough, one of my stretch goals here is the notification system. Throughout the curriculum there hasn’t been an assignment that utilized such a feature. As such, I will have to learn how to achieve this without knowing where to even begin. However, I have no reservations and will strive to implement this feature to the best of my ability.

My only other stretch goal, if you can call it that, is to implement all of my planned features as thoroughly as possible, leaving no stone unturned. I want every feature to be as complete as it can be. I also want the interface to be appealing, yet logical and accessible. In short, I want this app to feel professionally created, and not like a “project” quality product.