Properly.AI – AI Real Estate Agent

Views Implemented:

* Home Page \*
  + Displays hero image, and section for brief description of the platform. Properties list also populates with property cards and modal.
* Sell Page \*
  + Displays sell form to create property and add it to the database.
* Update Page \*
  + Displays sell form with pre-populated form fields taken from database and updates any fields on /update/{id} route.
* Buy Page \*
  + Displays properties with the modal feature and see more button, routing to property details page.
* Property Detail Page \*
  + Displays property and details, as well as ‘buy’ button that updates listing status field to sold.
* Admin Page \*
  + Displays all properties and users with the ability to update and delete each with admin role attached to user. (Not yet implemented in user model.)
* Support Page
  + Will display contact information and post to backend. Will also display FAQs, as well as support documentation received from backend.
* Thank You Page
  + Will display a thank you message for various functions, such as uploading a property, completing a transaction, etc.
* Blog Page
  + Will display a list of blog posts retrieved from backend and, when clicked on, will take you to blog detail on the /blog/{id} route.
* Sign Up Page
  + Will display a form to sign up and send data to backend for user authentication.
* Login Page
  + Will display form to log in and verify/authenticate with backend.
* Logout Page
  + Allows user to log out and redirects to home page.
* About Page
  + Displays about information from the company, potentially retrieving company employee profiles and bio’s from backend.
* Title Verification Page
  + Page that verifies ownership of property by hitting a parcel database and matching tax ID to property and owner that was submitted.
* User Profile Page
  + Allows a user to update their existing user data. Also allows admin (if role is assigned to user) to update existing user data.

Problems Encountered / Questions:

* Problem - I really wanted to figure out a way to let users’ login/signup and then verify that using my backend. I looked into a few packages and also using if statements to validate username and passwords but figured I would implement everything first and then worry about conditionally rendering pages based on user logged in or out status later.
* Question - When having a full stack app, should I have 3 dependency files? One for frontend, backend, and the app as a whole? I was a little confused on where to install packages - specific to sides of the application or in the root directory.
* Problem - VSCode kept flagging errors when I changed the names of files, even if the code (the route importing) was correct. Have you ran into this type of bug before? I had to rename my file to something completely random, save, and then rename it to what I wanted to change it to.
* Question – When should I use .jsx files for .js files in React? Does it matter?
* Problem – I kept getting errors defining the associations between my models so I moved them to a separate and it seems to work well. What is the best practice here?
* Question – How far should I nest files? I want to be thorough but not cumbersome.
* Question – How should I go about setting up ports? What is best practice and what isn’t?