Full-stack project

In Phase 4 of my software engineering course, we learned how to use Ruby on Rails, or Rails as it’s otherwise known, to create an API server. The idea is that by building an API to serve JSON data in Rails, it will provide control over the backend functionality: from how requests are handled and how responses are generated, to how models interact with the database using Active Record. By the end of the phase, we were shown all the elements to create a full stack application using a React frontend and a Rails backed. For our phase 4 project we were asked to build a full-stack project with a React frontend and a Rails backend with the following deliverables:

* Have at least three models on the backend, that include:

at least one one-to-many relationship

at least one many-to-many relationship

full create, read, update, delete (CRUD) actions for at least one resource

* Have at least three different client-side routes using React Router. Be sure to include a nav bar or other UI element that allows users to navigate between routes.
* Implement authentication/authorization, including password protection. A user should be able to log in to the site with a secure password and stay logged in via user ID in the session hash.

*Coming up with an idea*

Probably my least favourite part of the project process, is coming up with an idea of what application to build that will meet the project deliverables. I spent a while jotting ideas down and how the models would work and after a few days of playing around with different concepts, I decided to create the beginnings of an e-commerce application. The basic idea is that it’s an online store for bird house enthusiasts. Users can log in or sign up and once in can peruse the list of bird houses and click on them which will allow users to read, update or create reviews. I haven’t added a shopping cart feature in this phase but it’s something I’m looking to do for the phase 5 project of the course. Phase 5 project is also a full stack application so kind of similar to phase 4 but I imagine they are expecting the application to have some more advanced features. I have no idea how to do a shopping cart feature, so I figure that’s more advanced lol.

*Creating the models*

Now that I had an idea of what I was going to build, I started by creating the resources I would need for each model. So, using the rails resource generator I created tables with the columns I would need for the data to be populated together with the file for creating the API routes, a file for controlling the routing logic, a file for selecting the data that I wanted to display and of course a file for the model where the relationships between the models could be established. I did this for process for all three of my models, BirdHouse, User and Review. BirdHouse has many reviews and has many users through reviews. User has many reviews and has many bird houses through reviews. Review belongs to both user and bird house.

*Tackling application features*

Now that I had all the models and relationships set up, I began to build out each feature of my application one at a time. I started by creating the React components needed to be able to fetch the data from the API and display the list of bird houses. I then worked on the routing logic needed to enable the fetch to work. I followed the same structure for clicking on an individual bird house to view more details and then worked on the login/signup feature. I had a little trouble figuring out how to get the reviews to display on an individual bird house but after talking through it with my mentor I realised I just needed to add some code to one of the serializers to get this working. Finally, I worked on implementing full CRUD capability on the review resource. This was probably the most challenging aspect of the whole build and I had a mini meltdown trying to figure this out but eventually got it working after finding some helpful code in the course work that I could adapt for my project.

*Styling it up*

Once I was happy that the application was working as intended, I used CSS to style the application to make it as visually appealing as I could. I tried to use some of the CSS frameworks out there but I haven’t yet found one that I understand well enough to implement. They all seem a bit complicated for me and it might be something I will try tackle for Phase 5 but for this phase I just stuck with writing my own CSS, importing a font from Google fonts and adding a background image to my login/signup page.

*The final product*

With all this done, I had completed my application and I used Render to deploy this to the internet. The deployment didn’t go as well as I’d hoped and the app doesn’t work exactly the same as it does on my local environment but the jist of it is there and for now that’s good enough for me. You can see the final product at <https://bird-house-project.onrender.com/> and I’ve added some screen shots of it below. Phase 4 done!



