Todd Spainhour
Assignment 4 – Development
Doggy Details
SWE 6623 – Summer 2023

GitHub Repo: https://github.com/ToddSpainhour/DoggyDetails
Brief Video Walkthrough: https://www.loom.com/share/ea6a8db1dc97482dae041fe75d4b249b?sid=9e9b59b4-b6b7-4f34-ad02-a588109b3978
Frontend Port Number: 5500   Backend Port Number: 7260

Project Overview

Doggy Details is composed of three main parts: a frontend, backend, and a database.

Progress Report

There is still a lot of work to be done on this project. I'd say I'm only 20% complete. The work done so far has all revolved around the Owner class since a Pet can't exist without an OwnerID.

I thought the best place to start the project would be to make it so a user can create an account. This process begins by letting them select an email or username and check its availability. The value the user enters is sent to a C# endpoint where it's turned into a parameter for the query. The query checks the AccountEmail column to see if this value already exists. If the value already exists in the AccountEmail column, the query returns a bit value (basically a Boolean). If the query returns 0 (false) this means that the AccountEmail is already taken and unavailable. If the query returns 1 (true), the selected AccountEmail is available.

If the AccountEmail is already in use and unavailable, JavaScript displays a message letting the user know to try again. If the AccountEmail they picked is available, JavaScript dynamically displays the other input fields for them to continue the account creation process.

After the user fills out the password, first name, and last name fields the values they entered are put into a JavaScript object and sent to the createNewOwner endpoint. Then we take those values and use them as parameters for our insert statement. After an insert, the database returns the newly created, auto-incremented OwnerID using Scope_Identity() in sql. We pass that value all the way back to the frontend where we store in the browser's cookies. This quick (and unsecure) approach will serve as the way to save a user's authenticated state.

I think the most interesting part of the work I've done so far is in the OwnersController file.
https://github.com/ToddSpainhour/DoggyDetails/blob/main/DoggyDetails/DoggyDetails/Controllers/OwnersController.cs

GetAllOwners - I created this one for testing purposes. It's a way to confirm everything was hooked up properly. Users of the site would have no reason for this functionality, but you'll see results logged to the console.

UsernameAvailability - This is where the user checks if the username/AccountEmail is available (mentioned above). The argument provided gets passed into the query as a parameter. If it's available, more sign-up input fields dynamically display.

CreateNewOwner - This endpoint is an HttpPost since we're inserting a JavaScript object into the database. The database returns the new auto incremented OwnerID which I save to cookies.

Login – Allows users who already have an account to login and eventually see data specific to them.

GetAllPetsForThisOwner - This returns all the pets in the database for a particular OwnerID that's passed in as an argument.