

## Title: Furr-Ever Home

**Team:** Zunairah Abbasi, Saanika Fadia, Declan Griner, Samantha Larsen, Matthew Quintanilla, Judah Taylor, Leah Tiktin

### Project Description:

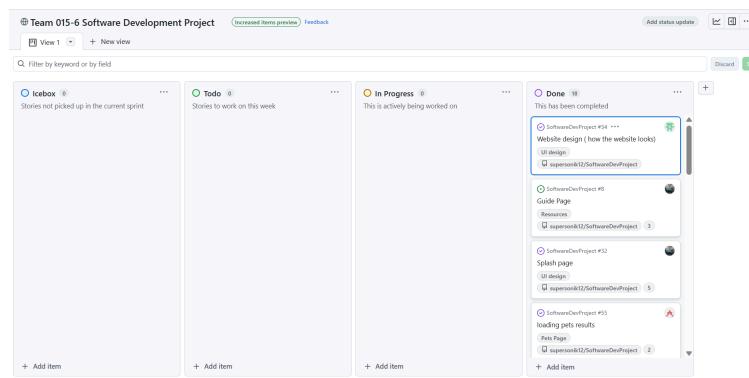
Our project, *Furr-Ever Home*, is a pet adoption website that helps users find adoptable animals in their area. The site displays pets using a clean card layout, showing each animal's name, age, breed, and a short description, along with buttons to learn more or express interest in adoption.

We built the site using HTML, CSS, and JavaScript, and integrated the Petfinder API to pull real adoption data. This allowed us to display up-to-date listings and make the experience more realistic and meaningful. Each team member contributed to different pages and respective functionality of the site, such as a home, splash, guides, shop, quiz, login/logout page and consistent UI across all. We used GitHub for collaboration, managing branches and pull requests to stay organized.

The main goal was to create a working adoption platform that was functional, easy to navigate, and connected to real-world data. We kept the structure simple so users could focus on the pets themselves without distractions.

Overall, *Furr-Ever Home* gave us practical experience with API integration, frontend development, and teamwork. It also showed how we can build something that makes an impact by making the pet adoption process easier and more accessible for users.

**Project Tracker - GitHub project board:** <https://github.com/users/supersonik12/projects/1>



### Site Demo:

<https://drive.google.com/file/d/1Vr9HHYoACx335UlxNOe6RjXlIvbwf3c0/view?usp=sharing>

**Git Repository:** <https://github.com/supersonik12/SoftwareDevProject.git>

## **Contributions:**

**Zunairah Abbasi:** I created wireframe mock-ups for the login, sign-up, guides, and home/splash pages. I also designed the central UI layout and visual style, which was then adapted across the rest of the site to maintain consistency. I focused on making the site look clean, user-friendly, and welcoming. I collaborated closely with the front-end team to make sure the implemented design matched our original vision. In addition to design, I was responsible for writing and organizing the project report.

**Saanika Fadia:** I implemented the register and login functionalities of our website. This involved working with the user database, writing API routes, and basic UI. I contributed to the account management functionality and authentication. I also worked on testing and bug fixing. I wrote unit tests, did use-case testing, and helped other members of the group debug various issues.

**Declan Griner:** I primarily worked on the quiz functionality, including both the frontend implementation. I also did some work on the database, primarily to support the quiz, though these contributions supported other features as well. After the quiz was fully functional and known bugs were fixed, I worked on site-wide error handling, in order to improve stability and user perceived stability.

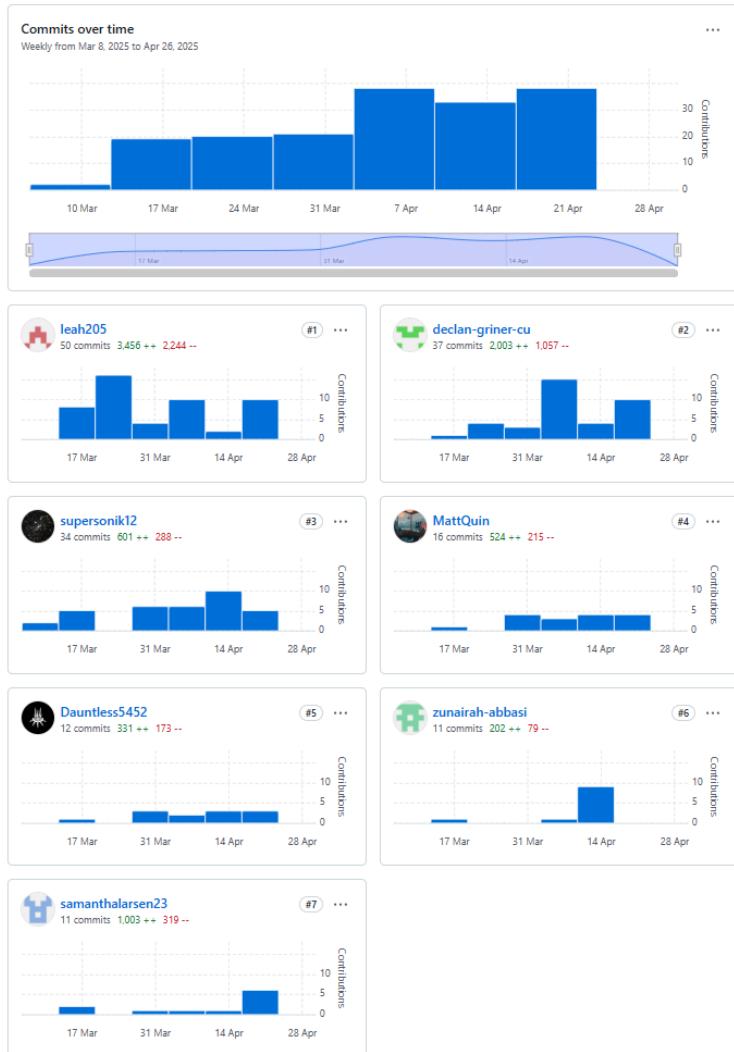
**Samantha Larsen:** I created the nav bar, account page, and favorites functionality. I also modified most of the pages to make sure we had a consistent UI. This involved a combination of front end and back end coding. I worked with our database, wrote API routes, and contributed to our css page. I also used handlebars and node.js to create dynamic pages for the account page/favorites functionality.

**Matthew Quintanilla:** I worked on the guide page and the initial splash page. For the guide page, I made sections for each type of pet and their age, (puppies, dogs, kittens, cats, etc. ) Each section has a short description on the basic needs of the animal. At the bottom of each description there is a link to a vet website that specializes in that certain pet and has more details on how to take care of it. For the splash page I just coded the layout, a basic square with an image and text buttons.

**Judah Taylor:** A big thing I picked up at the beginning of the project was the store page. I started by just designing the cards that the items would be displayed on. The most difficult section was deciding how to do the api routes for the page. I spent a significant amount of time trying to figure out if it would be feasible to use the amazon api to pull directly from the site dynamically. I ended up just mocking the data for the presentation. After that I made the page look nice and ready for the presentation.

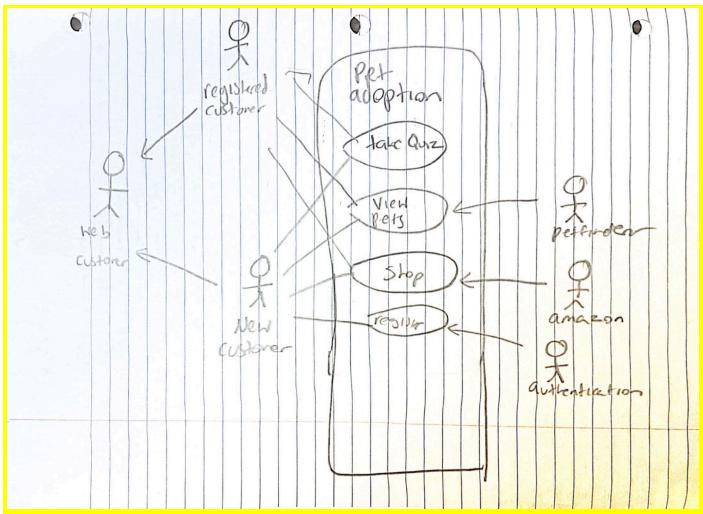
**Leah Tiktin:** I initially worked on getting everything set up on the website. I mainly worked on the homepage of the website. I worked with the petfinder API to display pets available for adoption, with buttons to get more information. I segregated pets into multiple pages using frontend js. I also created various filters that called the api to narrow down the pets and a

search functionality to get a specific breed. I used css to style the home page and used the database to create a button to get compatible pets for a specific user.



### Use Case Diagram:

[https://lucid.app/lucidchart/4dbd19dd-9011-4e0c-8e13-67aac2ca3cbe/edit?invitationId=inv\\_e1ff8246-76b5-4e3a-8caa-1d3c594e3847](https://lucid.app/lucidchart/4dbd19dd-9011-4e0c-8e13-67aac2ca3cbe/edit?invitationId=inv_e1ff8246-76b5-4e3a-8caa-1d3c594e3847)

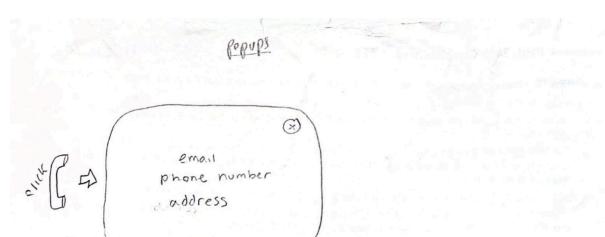


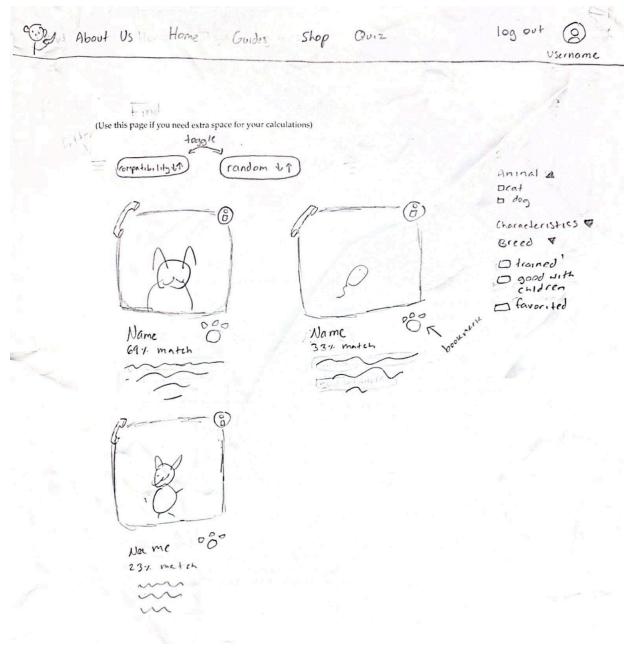
### Wireframes:

- Login page
- Sign Up Page

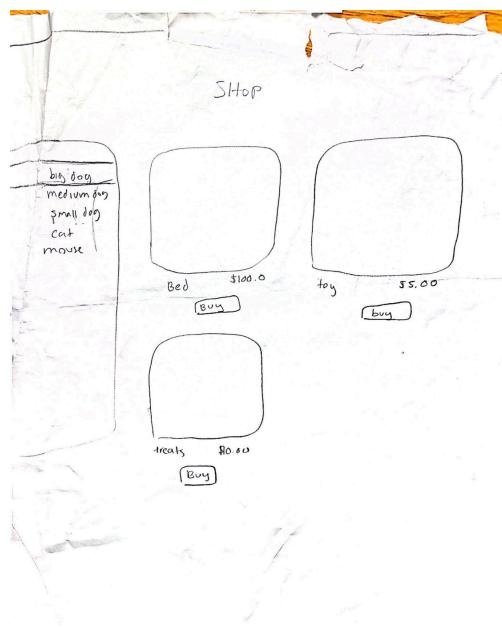


- List of pets

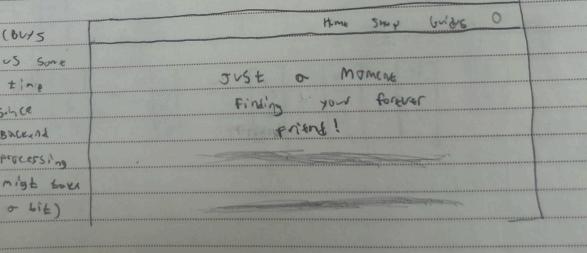
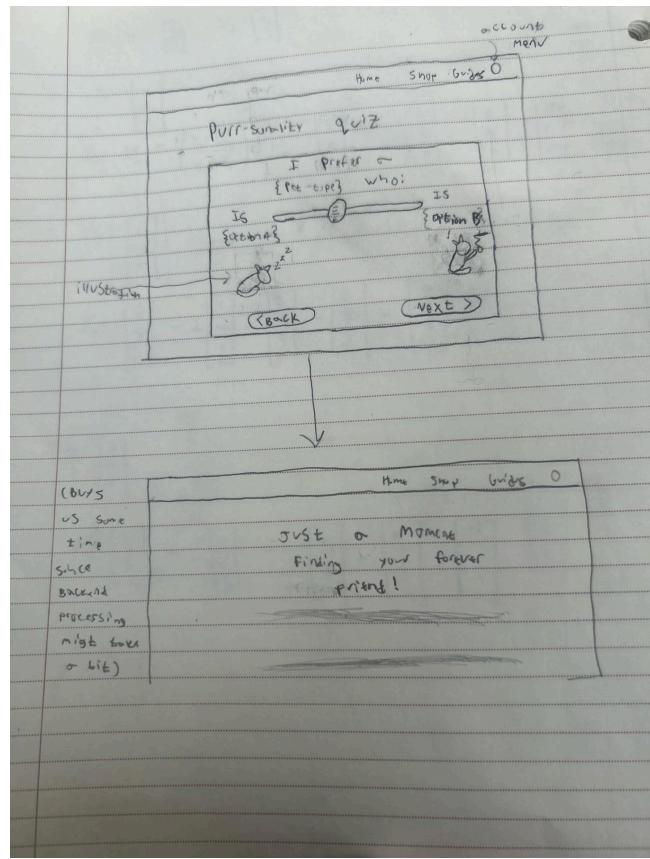




- When you click on a pet it shows you pet details
- d. Pet Store



- e. Purr-sonality Quiz



#### f. Checkout

#### g. Guide

##### i. Pet-care Guide



##### ii. Veterinarian Contacts

#### h. Home Page/Splash Page



-Pet search should be different page

## 1. Account creation and introduction

The user should be able to create an account and take the intro quiz.

- Test data: name, email, password, user-provided quiz responses
- Test environment: localhost, dev environment
- Test results: The Account page should show the user's name, email address and the user's quiz responses as a part of the user profile. If they logout and log back in, the profile should remain the same.
- Testers: Us (developers)

Results: We observed that it was slightly confusing to new users that they had to navigate to the "Register" page (instead of directly to the quiz.) We fixed this by including a link to the quiz on the nav bar, and creating a welcoming splash page with register and login links. The rest of this use case works as planned.

## 2. Pet discovery page

The user should be able to see the list of available pets based on their profile.

- Test data: user profile, type, description, health
- Test environment: localhost dev environment
- Test results: The Pet page should display a gallery of pets with their photos. When a user clicks on a photo, the pet's description should display their info. They can exit out of the info and scroll through other pet portfolios. Somewhere on the page, there should be
- Testers: Us (developers)

Results: Users are able to click into the pet discovery page once they have taken the quiz. The filters are intuitive and helpful. Also, the addition of the "Get Compatible Pets" button makes it more clear when the home page is showing specific or all pets.

## 3. Editing account information

The user should be able to visit the account page and change their account name or password.

- Test data: name, email, password

- Test environment: localhost dev environment
- Test results: The form on the account page should first request the user's password before they are allowed to change any information. Then they can fill out a form, and their potential new name will show on the account page, and their information in the database will be updated.
- Testers: developers

Results: Users are able to change their password or name after creating an account. It is clear what needs to be changed, but it could be more clear to users that they don't need to change both name and password at the same time, since those are separate functionalities.

#### 4. Logout Functionality

The user should be able to log out and be redirected to the login page.

- Test data: An active user session
- Test environment: localhost, dev environment
- Test results: After clicking logout, the user should be redirected to the login screen.  
Attempting to access a protected route like /account or /profile after logging out should return a 401 error or redirect back to the login page.
- Testers: Developers

Results: Successful login and logout functionality. Users are able to login and logout of their accounts, which are stored in the database.

#### Risks and Edge Cases

Potential issues that may arise during user interaction with the system.

- Account creation: User submits incomplete or invalid form fields, or tries to register with an email that already exists in the database.
- Quiz responses: User skips questions or inputs unexpected answer formats.
- Pet Discovery: No pets match the user's profile criteria, or the pet data fails to load due to a backend/API error.
- Editing account information: User provides an incorrect current password when attempting to make changes, or submits invalid new input.
- Logout: User session is not properly cleared, allowing access to protected routes even after logging out.

Deployment Link: <https://softwaredevproject-1.onrender.com>