Project Report Team 14-01

Title: Fur-Ever Friend Finder

Team Members:

- Lindy Zhang
- Lucy Profeta
- Vince Frazzini
- Andrew Wang
- Yash Singh
- Matthew Simpson

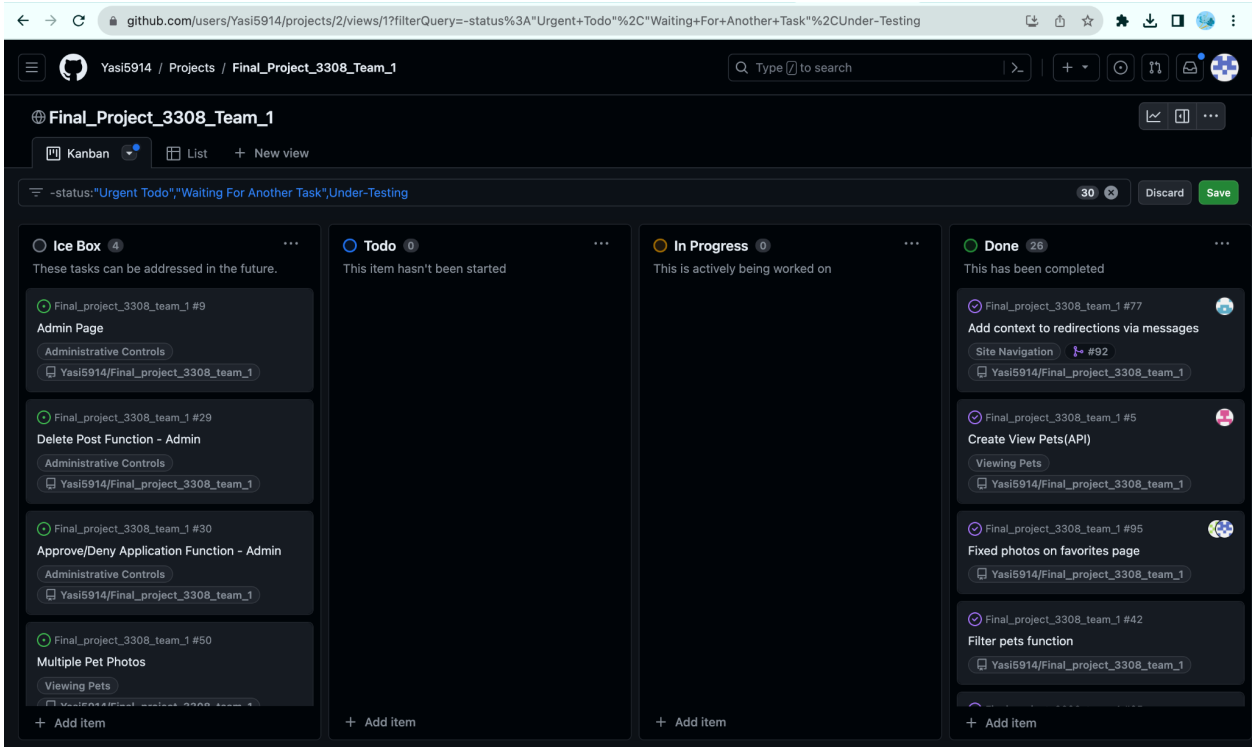
Project Description:

- Fur-Ever Friend Finder simplifies the search for furry companions, offering a seamless experience both locally and nationwide. The user-friendly explore pages are equipped with dynamic filters, including species, breed, color, and more, empowering users to refine their search and discover their ideal pets effortlessly. Users can also favorite potential pets, curating a dedicated favorites page for easy access.
  Not only can users explore, but they can also actively contribute by posting animals available for adoption. The platform's secure login function facilitates efficient communication between users, storing essential information for seamless connections. This functionality extends to the ability to begin an application after being redirected to external adoption platforms for pets beyond their local vicinity.
  With a user-centric approach, the application ensures individuals can manage their adoption journey efficiently. Users can review and update their posted content, fostering a sense of community engagement. Additionally, the platform provides a space to modify account information, granting users the flexibility to tailor their experience to their preferences.

Project Tracker:

- https://github.com/users/Yasi5914/projects/2

- Screenshot of project board:



Video:

- https://drive.google.com/file/d/1ktssfh-XtXmfjSWRnDHPaOVu2yec5yZA/view?usp=sharing

VCS:

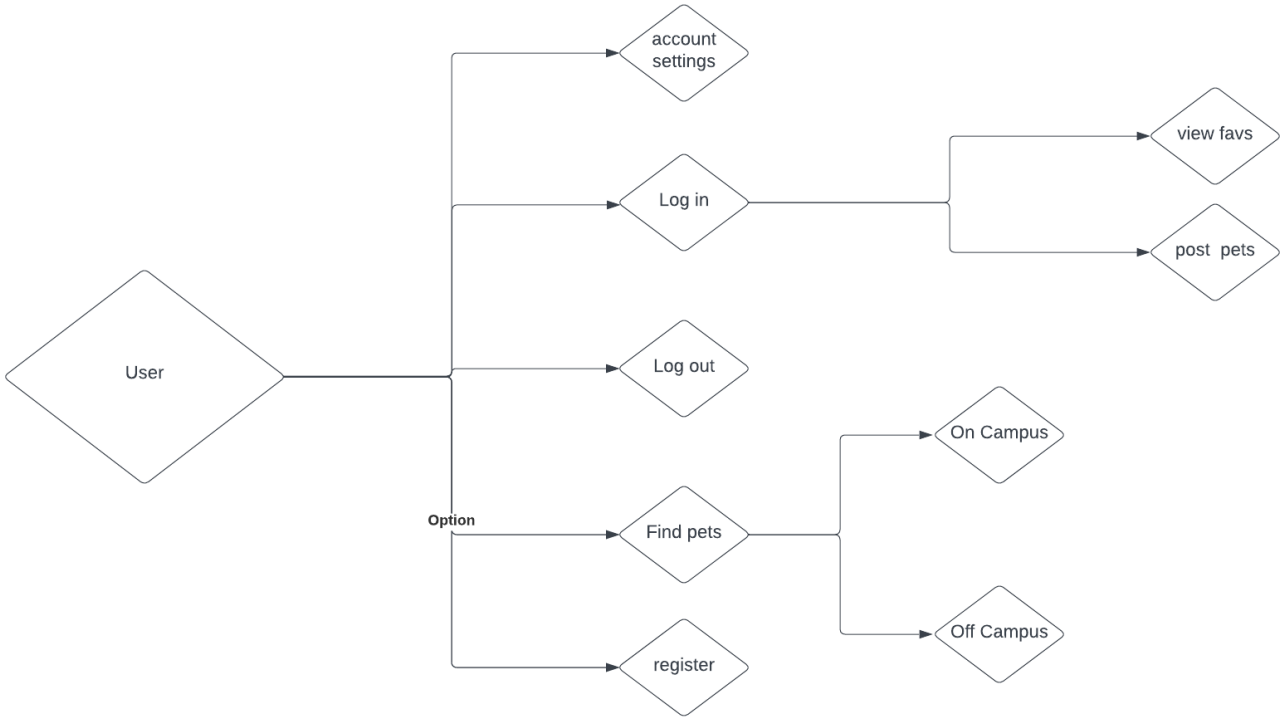
- https://github.com/Yasi5914/Final_project_3308_team_1

Contributions:

- Lindy: I handled login, register pages, and their testing. Contributed to navbar creation, including a separate one for login/register. Updated databases' structure as project requirements evolved. Created the favorites page, implementing it for explore boulder and explore anywhere. Collaborated with Lucy on displaying liked pet images on the favorites page from boulder or anywhere explore pages.
- Lucy: I created the initial database design and insert statements. I also created the Explore Boulder page and the Boulder pet cards which display information from the database. Collaborated with Lindy on displaying liked pet images on the favorites page from Boulder or anywhere explore pages. I created the application logo and formatted it onto the navbar. I added the application colors onto all buttons and pages and formatted the padding on all pages.

- Andrew: I created the partials in EJS for the head, the menu, the footer, and the message, and then formatted them with Bootstrap and CSS. The menu was created in Bootstrap. Additionally, I implemented password hashing via the bcrypt node.js library. I created the account page and implemented the ability to change all account details except for the username used to login. I improved UX by removing unnecessary redirects and using the message partial to explain the remaining ones.
- Yash: I made the pet pages for all the pets on the website. I had to make two different versions of the pet page, one for the explore anywhere page, and one for the boulder page. For the explore Anywhere page I used the Petfinder API to retrieve data about the pet. On the other hand, for the boulder pets I ran an SQL query on our database. I also did a lot of changes to the website's CSS for improved readability, and just made it look more professional. I also helped on the registration page and testing for its functionality.
- Vince: I made the explore anywhere page for the website that implemented the PetFinder API. For this I designed the pet cards that were then implemented into the explore Boulder page. I also implemented a filter system for the explore anywhere page that allows the user to add any number of filters from a dropdown menu and then display the current filters applied.These filters can then be easily removed individually after being selected. I helped with the implementation of the petfinder API onto other pages as well, specifically figuring out how to pass specific petIDs for the More Info page of a pet.
- Matthew: I was in charge of creating the My Posts and Create Posts pages. I made the two ejs files and wrote the routes in index.js for getting post information, posting pets, and deleting posts. I also added css styling to the My Posts page and Yash helped with the css for the Create Posts page. When a user creates a post, it is given a unique ID which is linked to the user and will display on the My Posts and Explore pages. The user can also delete a post which removes it from the database.

Use Case Diagram:



Test results:

- Unit testing: We wrote test cases for the register and login page. We had two types of tests for our unit testing, which were positive and negative. For the positive, test cases sent expected inputs and expected the register and login page to work. For the negative, test cases sent unexpected inputs and expected the program to fail and return an appropriate error message.
- User acceptance testing: We asked our friends who were not familiar with software development to go through our website and tell us if they could navigate through the website without a problem. We also asked them what they would like to see changed. Then we used that feedback on further implementation. We also asked them what we should do to be different from our competitors, and considered that feedback for the next sprint. For a smaller project where we did not have time outside of implementation, it was constructive to do User acceptance testing. The feedback that we received from other people showed us what the actual website's impression was on the user, and how we can improve that even more.
    - Login UAT: We had the users try to login without registering and then try to login after registering. When a user tries to login without already having registered an account, we want the user to be redirected to the register page. We then had the user register and after they were redirected to the login page, we had them login with the username and password they just registered. We tested successful and

unsuccessful logins in this use case. We asked the users to login with a correct username + password and an incorrect username + password for this case.
    - Observations: The users typically registered an account when they first opened the website. They then logged in with the correct username and the password in order to fulfill the correct username and password. When it came to the incorrect username and password, they would normally put in the wrong username or wrong password. After testing this, we added a message to alert the user when they put in a correct username with an incorrect password. The user would not be redirected to the register page for typing the password in incorrectly. However, if the username was incorrect, we did redirect the user to the register page. There was no deviation from our expected actions.
- Register UAT: We had the users try to register an account with a missing username in order to cause a failed registration and ensure that the website sent a message saying they were missing a required field (username or password). Once they had failed registration, we had them enter the required fields so that we could ensure they were redirected to the login page after. We asked users to successfully register an account and also unsuccessfully register an account for this case.
    - Observations: The users would navigate to the register page and then fill out all the fields for the successful registration. We did not see any deviations from the expected actions for this case. For the unsuccessful registration, we saw users try to register the same account again which prompted us to add in the functionality of not allowing users to register with the same username and to display a message saying that they could not register the account with that username. Other than this, there were no deviations from the expected actions. We had users try to register without putting in a username or password and this caused an unsuccessful registration.
- Explore page: We had the users make sure that the correct pets showed up depending on if the user wanted to explore pets in boulder or if they wanted to explore pets from around the nation. The users clicked on the Explore option in the navbar and chose Boulder first. We made sure that the page with only Boulder pets showed up along with the heading “Explore Pets in Boulder!”. The users then clicked on the Explore option in the navbar again and chose Anywhere. We made sure that the page with pets from petfinder loaded along with the heading “Explore Pets Anywhere!”. We asked users to explore pets in Boulder and explore pets from anywhere for this case.
    - Observations: The user would be taken directly to the explore pets in boulder page upon logging in, so we did not see any deviation from exploring pets in boulder. When the users explored pets from anywhere,

they would click on the dropdown arrow next to explore to see what was contained there, and then upon seeing the options Boulder and Anywhere, they would choose Anywhere to explore those pets. There were no deviations from our expected actions.

- Creating + Deleting a post made by a user: We had the users create a post for the explore boulder page and then delete that same post. The users were asked to navigate to the My Posts page to see any previous posts they had made. Then, we had them create a post by clicking on the Create A Post button on the My Posts page. We then had them populate all the fields and create the post. The next task was to make sure that the post had been successfully created by making sure the post the user just made showed up on their My Posts page. The user was redirected to the My Posts page after creating the post, so that change is easy to see. We had the user navigate to the explore boulder page to make sure the post showed up on this page as well. The posts are added to the bottom of the page. After making sure the post was added everywhere, we had the user delete the post by going back to the My Posts page and clicking on the Delete a Post button. The page asked the user if they were sure and then deleted the post. We made sure the post was deleted successfully by checking the My Posts page and also the explore boulder page. We asked users to create a new post and then delete that post for this case.
    - The user would typically take a moment to decide where they would create a post and everyone decided that it would likely be under the My Posts page. After navigating to that page, users would click on the Create a Post button to begin their task. Once they had filled out all of the fields and clicked on the button to create it, they were redirected to the My Posts page where they were able to verify that the pet they had created had showed up. Many people ended up returning to the explore page to see the pet they had created on the explore page. After they had done this and we had verified their pets had been created, we had them delete the post. Everyone returned to the My Posts page and clicked on the Delete button at the bottom of the pet card they had just created. The users then went back to the explore page and checked to see if their pet was still there. We did not see many deviations from the expected actions, but we did make the navbar options more obvious for the pages they would direct users to.
- Favoriting/un-favoriting a pet: We had the users scroll through the explore page and choose pets that they like and favorite them using the heart in the top right of the pet cards. After favoriting a pet, the users were asked to check the My Favorites page to ensure all the pets they favorited showed up there. Once we determined all the pets showed up, we had the users un-favorite all the pets to ensure that the pet cards were removed from the My Favorites page once they

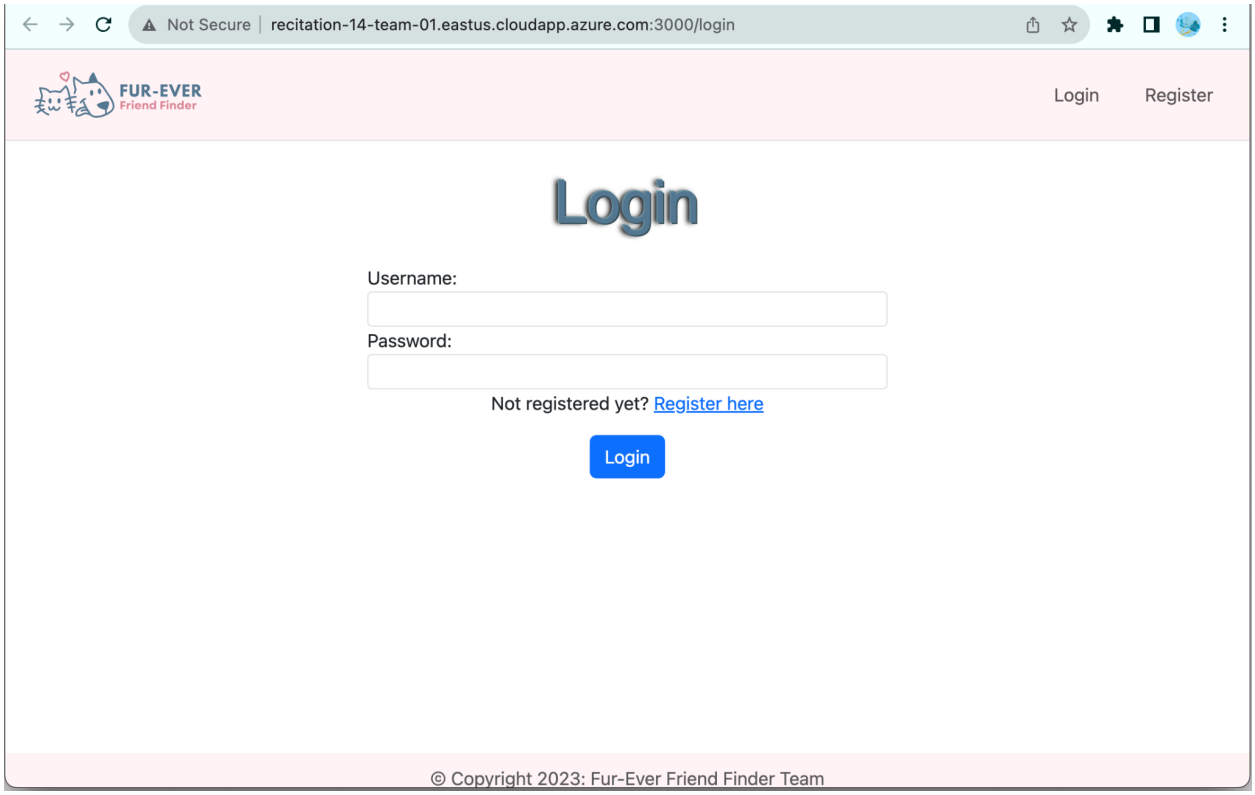
clicked the heart in the top right of the corner to un-favorite them. We asked users to favorite pets and then unfavorite them from the My Favorites page for this case.

- The users all checked the My Favorites page out first, and after seeing it was empty, they returned to the explore page where they could find their favorite pets. We had them choose a few pets from the Boulder and Anywhere pages. They all clicked on the heart button to favorite the pet and a few of them went to the pet page to get more information about the pet they liked. Everyone checked the My Favorites page after they had liked a few pets to make sure that the ones they had liked were showing up on the favorites page. Their reasoning was that they weren't always sure that the pet was showing up. Another feature we could add in the future would be displaying a message when a pet is added or changing the color of the heart on the explore page when the user likes it. Their behavior was consistent with the use case and we did not see big deviations from the expected actions. When the users were removing the pets from favorites, they did not deviate from the expected action and clicked on the pink heart in the top right corner of the pet card on the favorites page. The page reloaded to show the pet was removed so no users took any additional actions.

Deployment:

- http://recitation-14-team-01.eastus.cloudapp.azure.com:3000/login

- Screenshot:


Not Secure | recitation-14-team-01.eastus.cloudapp.azure.com:3000/login
FUR-EVER
Friend Finder
Login
Register
Login
Username:
Password:
Not registered yet? Register here
Login
© Copyright 2023: Fur-Ever Friend Finder Team