Title:
PetLife (Team 011-01)

Who:
Charlie Gau
Collin Hayes
Ella Herniak
Amber Kou
Logan Proffitt
Eric Wilmes

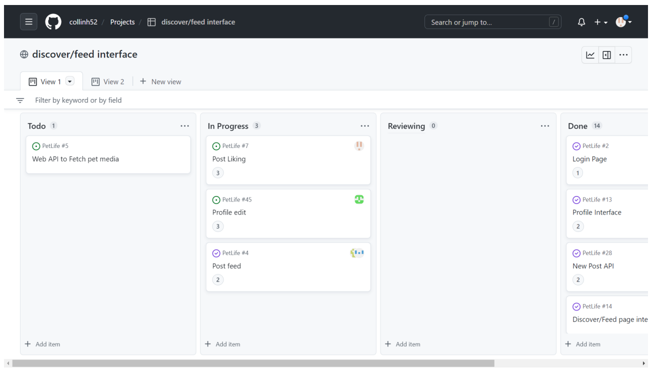
Project Description:
PetLife is a social media site aimed solely at providing a space for pet owners to share photos of their beloved companions. Users, after creating an account, can scroll through a feed of pet photo posts. They can then like those posts, view their profile, and create posts of their own. Within their profile, they can also set a profile picture, edit their information, and share which types of pets they own.

In terms of the backend, PetLife operates by storing posts in a PostGreSQL database, with links to image storage via the Cloudinary API. The website is built using HTML and CSS, primarily utilizing Bootstrap 5 stylesheets, and constructed with partials through EJS. Functionality is executed through NodeJS, and the site itself is deployed and hosted locally through the CU Boulder wireless service.

PetLife was ideated and developed over a period of four weeks by a team of six, communicating both in-person and over services such as Google Meet and Discord. The primary IDEs used were Visual Studio Code and IntelliJ IDEA. Progress was tracked through the GitHub Project Board, and code collaboration was also made possible through Git and the GitHub interface.

Project Tracker - GitHub project board:

- Link to project board: https://github.com/users/collinh52/projects/1/views/1
- Screenshot:


collinh52 / Projects / discover/feed interface
Search or jump to...
View 1
View 2
New view
Filter by keyword or by field
Add item

discover/feed interface

Todo 1
- PetLife #5 — Web API to Fetch pet media

In Progress 3
- PetLife #7 — Post Liking — 3
- PetLife #45 — Profile edit — 3
- PetLife #4 — Post feed — 2

Reviewing 0

Done 14
- PetLife #2 — Login Page — 1
- PetLife #13 — Profile Interface — 2
- PetLife #28 — New Post API — 2
- PetLife #14 — Discover/Feed page inte

Video:
https://drive.google.com/file/d/1zuqJswytvsjM4m9uBIst1BjDrDaWUFf2/view?usp=sharing

VCS: https://github.com/collinh52/PetLife

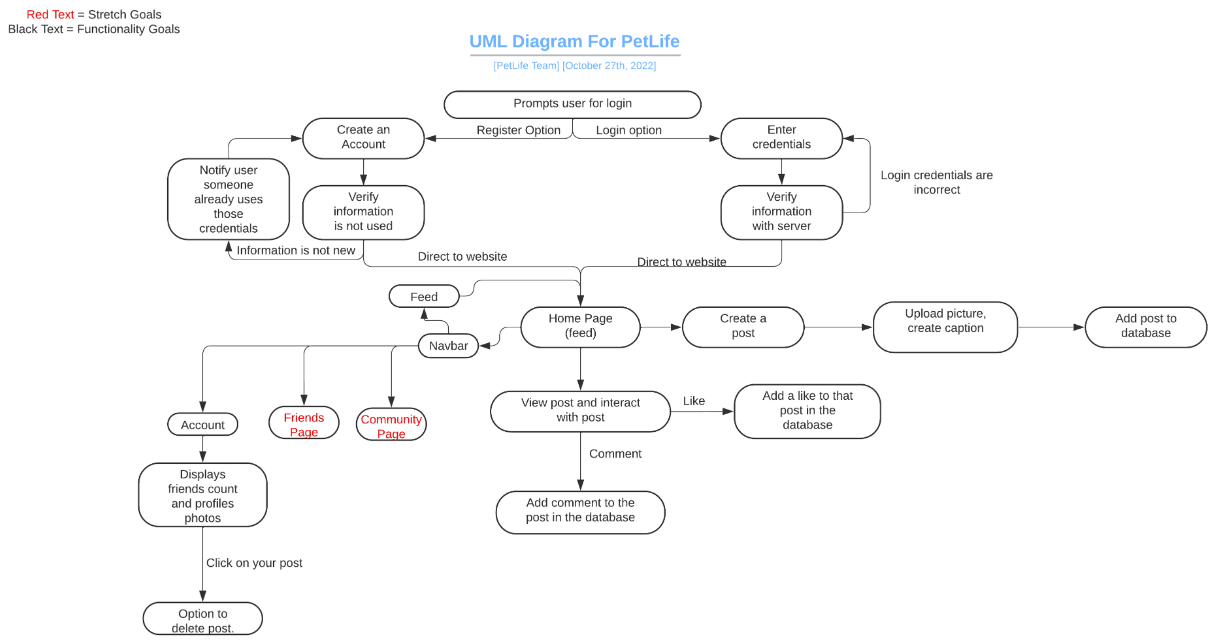
Contributions:

- **Charlie:** Charlie's first contribution was the login page. He did the back and front for that, cleaned it up and put a nice banner above the credentials. He then worked on creating the profile page. This took him some time to completely grasp using session data to present someone's information from the database, but got it eventually. He then worked with logan to get post feed working, which also took a second, but it was successful. He finished off his contributions to the team by working with Logan again to get the communities displaying on the post. This involved more than 1 query, which was newer territory so it took some time.
- **Collin:** Collin first worked on the design and creation of the database. Then he created the API call to create a new post and later added in the feature to include pictures in those posts. Then he added to the post liking API that Amber created to count the number of likes on a post and deleted likes when a post is already liked. He also created the edit profile API and added the feature to view other user's profiles. He used NodeJS for the backend APIs, PostgreSQL for the database, and worked on the frontend using EJS and HTML.
- **Ella:** We decided on using Cloudinary to store images, since we couldn't store them in postgres, and I primarily worked on setting up the cloud and connecting it to our node server. After I connected it successfully, I used the multer library so that a user could upload any image file from their computer to cloudinary. Next, I worked on an api to insert the information returned by cloudinary into the SQL database so that my other team members could finish the website's posting functionality.
- **Amber:** I primarily worked on the page partials (footer, header, navbar) as well as the like button, implementing a NodeJS call that increased the number of likes when the button was clicked. I was also responsible for most of the non-code aspects of the project, such as taking all notes during meetings, keeping the README updated, and setting up templates and document structures for the presentation and project report.
- **Logan:** The first thing I worked on was the registration page. I created the ejs page and node js api to handle registration and hash the password. Next I created the ejs page for creating new posts. This page included image, caption, and location fields. Then, Charlie and I worked on the post feed,

creating the api and html to display user posts. We also added a display for the user’s communities on the website’s profile page.

- **Eric:** The very first thing I did was set up the docker-compose for the project so we could start development right away. Next, I worked on the home page so that we could start putting our dog posts on the landing page of our website. Finally, I worked on the communities page of the website, making a call to the database to render the communities, then connecting the join button to the user’s communities in the database. I also worked on getting our deployments up for lab 12. The technologies I worked with were NodeJS, HTML, CSS, EJS, Docker, and SQL.

Use Case Diagram:



Red Text = Stretch Goals
Black Text = Functionality Goals
UML Diagram For PetLife
[PetLife Team] [October 27th, 2022]
Prompts user for login
Register Option
Login option
Create an Account
Enter credentials
Notify user someone already uses those credentials
Verify information is not used
Verify information with server
Login credentials are incorrect
Information is not new
Direct to website
Direct to website
Feed
Home Page (feed)
Navbar
Create a post
Upload picture, create caption
Add post to database
Account
Friends Page
Community Page
View post and interact with post
Like
Add a like to that post in the database
Comment
Add comment to the post in the database
Displays friends count and profiles photos
Click on your post
Option to delete post.

Test results:

https://docs.google.com/document/d/1YdMtUUgVP3ZXwBPKmfMQyt4p0zOlDMDqiGGnYOEpP40/edit?usp=sharing

Deployment:

We deployed to the computer science department’s servers. Sadly, this deployment does not stay up, so to access the app, you need to follow these steps.

1. Clone the git repository to your local machine
2. Navigate to Petlife/ProjectCode/src
3. Run the command `docker-compose up`
4. Navigate to localhost:3000
5. You're there!