



Course Project Report

CSI4999: Senior Design

Pet Symptom Tracker

Team:

Valeria Hernandez (Product Owner/Team Lead)

Ruth Wager (Scrum Master)

William Eng (Developer)

Mike Ward (Developer)

Silair Sileewa (Developer)

Project Goal

When loved pets grow sick, it becomes a constant worry of whether the pet is getting better. With the pet behavior tracker, different symptoms and behaviors can be logged for vets' usage and diagnosis. This project aims to enhance the quality of pet care, create stronger owner-pet relationships, and contribute to the overall welfare of animals.

Target Environment

Our target users will be any pet owner. Any pet owner will be able to access this tool and use it to track their pet's health and track health. The incentive for using this app will be for pet owners to easily document their pets' behavior, as it can help tremendously in determining detrimental behavior expressed in their pets.

Objectives

The primary goal of our pet behavior tracking system is to help pet owners and pet care professionals to better understand, monitor, and improve the well-being of pets by providing a user-friendly platform for logging, analyzing, and visualizing pet behaviors over time. This project aims to enhance the quality of pet care, create stronger owner-pet relationships, and contribute to the overall welfare of animals.

User Stories:

Identifier	User Story
UST-1:	As a user, I want to create an account with a username and password.
PRIORITY	1
Estimate: 25 Points	(1 points) designing the login page (5 points) designing database to keep track of the new users (5 points) adding an authentication method (We will be using Auth0) (8 points) adding the backend to communicate with the database (5 points) testing and debugged
UST-2:	As a user, I want the option to log in using my email or social media accounts.
PRIORITY	1
ESTIMATE: 21 points	(8 points) adding auth0 authentication tool (8 points) adding the backend to communicate with the database (5 points) testing and debugged
UST-3:	As a user, I want to reset my password if I forget it.
Priority	2
ESTIMATE points	(5 points) adding an authentication method (8 points) adding the backend to communicate with the database (5 points) testing and debugged

UST-4:	As a user I want to have all my pet's personal information available to me (name, sex, color, breed, species, weight, birthday, microchip number, insurance etc.)
PRIORITY	1
ESTIMATE:	<p>(3 points) Design Database that will ensure all this data can be stored properly</p> <p>(5 points) Create SQL script to create db</p> <p>(5 points) Create UI design for the website that will show all the pets</p> <p>(8 points) Research and implement a form for the user to input all of this information</p> <p>(5 points) Test the form and ensure information correctly stored in DB</p>
UST-5:	As a user, I want to easily add and remove animals from my account
Priority	1
Estimate:	<p>(5 points) Ensure the ERD accounts for adding and removing pets</p> <p>(5 points) Ensure UI design includes add and delete buttons for the users</p> <p>(5 points) add backend so the buttons now function</p> <p>(8 Points) ensure proper communication to the database</p> <p>(8 points) Test add and delete features and check database after done</p>
UST-6:	As a user, I want to be able to safely store my pet's medical data (encryption)
Priority	1
ESTIMATE	(8 points) With the correct AES key, the application can decrypt requested

	<p>data</p> <p>(8 points) Without the AES key, data is unreadable directly from database</p> <p>(3 points) Database will support AES</p> <p>(5 points) Server and Database will share new AES key every 24 hours</p>
UST-7	As a user, I want to record specific symptoms or health concerns for each pet, such as coughing, sneezing, limping, or vomiting.
Priority	1
Estimate:	<p>(5 points) Ensure UI design includes log symptom button for the users</p> <p>(5 points) add backend so the buttons now function</p> <p>(8 Points) ensure proper communication to the database</p> <p>(8 points) Test add and delete features and check database after done</p>
UST-9	As a user, I want to indicate the severity and duration of each symptom.
Priority	1
Estimate	<p>(5 points) Ensure the form has a section dedicated to that</p> <p>(8 Points) ensure proper communication to the database</p> <p>(5 points) Test the ability to input</p>
UST-10	As a user, I want to be able to easily view my pet's previous medications
Priority	1
Estimate	(13 points) query to DB that can select all the inputted info associated with logged-in user

	(8 points) Display all the user's pet's information in a chart for ease of user (8 Points) ensure proper communication to the database (8 Points) Test UI with different logged-in users
UST-12	As a user, I want to be able to have an easy-to-use UI
Priority	1
Estimate	(13 Points) Design each UI using Figma (1 point) Decide on the color scheme with team (8 points) Implement the UIs (8 Points) Test UI all together
UST-13	As a user, I want to see a cute UI
Priority	1
Estimate	(8 points) design every page for the website using Figma (1 point) decide on color scheme for the website (13 points) using REACT, code the frontend for the website (5 points) ensure that the pages are all connected, and the user can click around the website (5) Testing UI
UST-14	As a user, I want access to a FAQ section or user guides to help me navigate and use the platform effectively.
Priority	2
	(3 points) design the UI that will hold FAQ section

	(3 points) implements the UI using react (3 points) test
UST-16	As a user, I want to see nearby vets in my area
Priority	2
Estimate:	(3 points) design the UI that will hold a button for user to click to view vets in their area (3 points) implements the UI using react (3 points) test

Architecture

See [Architecture Requirements](#)

Process model & schedule

As a team, we have decided to utilize the agile methodology, specifically scrum. We are using the scrum methodology because scrum allows for us to be small and self-organizing. A member's webpage can be broken down into concrete deliverables to build our project consistently. We have our backlog to continuously see what changes we have performed and what we do on a week-by-week basis. We will be setting up our tasks to be delivered in two-week intervals. All deadlines for tasks will be in the backlog and have updates next to them so team members can easily refer to the progress of other team members.

Sprint Schedule

Sprint 0	<ul style="list-style-type: none"> ★ Formally meet the team and begin brainstorming projects. ★ Research needed steps to start 	Estimated sprint completion: 09/11/2023-9/25/2023
----------	--	--

	<p>a list of technologies that could be used</p> <ul style="list-style-type: none"> ★ Begin figuring out our timeline for the deadlines ★ Plan and list out the needed UML diagrams for project 	
Sprint 1	<ul style="list-style-type: none"> ★ Begin design and research of how the user registration and login will look like ★ Develop the full list of user stories and divide up by backend, frontend, database, etc ★ Research more into auth0 as it is the way the group plans on implementing login and assist in keeping track of users ★ Add priority to the user stories ★ Setup GitHub and possible deployment service (Vercel) ★ Continue making behavioral diagrams to ensure functionality of the application ★ ERD and SQL script creation 	<p>Estimated sprint weeks:</p> <p>9/25/2023 – 10/9/2023</p>
Sprint 2	<ul style="list-style-type: none"> ★ Define the architectural components ★ Design and begin implementation of the forms that the users will utilize 	<p>Estimated Project sprint weeks:</p> <p>10/9/2023-10/23/2023</p>

	<ul style="list-style-type: none"> ★ Begin implementation of some of the landing pages ★ Connect the code and debug deployment in Vercel ★ Set the functional and nonfunctional requirements for the project. ★ Develop the architecture of the project ★ Research and figure out the best way to deploy the database to connect the best with the Vercel website ★ Try to connect database to begin CRUD implementation <ul style="list-style-type: none"> ○ UST-1 ○ UST2 	
Sprint 3	<ul style="list-style-type: none"> ★ Begin making list of test case to use while testing ★ Test the user stories implemented by this point ★ Go back to original UI design and decide if the design needs to be updated in Figma ★ Add in the updated forms from Ruth into the UI for more detailed information to be inputted and tested ★ Begin implementation/research for adding in the AES 	Estimated Project sprint weeks: 10/23/2023 – 11/6/2023

	<p>encryption that will be used in the database</p> <ul style="list-style-type: none"> ○ UST-4 ○ UST-5 ○ UST-8 	
Sprint 4	<ul style="list-style-type: none"> ★ Continuously test branches that are in development ★ Ensure roles of the Vet vs. owner are correctly functioning ★ Clean up the UI ★ Implement the extra features such as finding a vet and food recommendations if the rest of the UIs have been successfully implemented <ul style="list-style-type: none"> ○ UST-9 ○ UST-11 ○ UST-12 	<p>Estimated Project sprint weeks: 11/6/2023 – 11/20/2023</p>
Sprint 5	<ul style="list-style-type: none"> ★ If we are on track and get the 1st priority features completed, we can focus on testing ★ Once the deep testing is complete, we will begin implementation of 2nd level priority <ul style="list-style-type: none"> ○ UST-15 ○ UST-14 ○ UST-10 	<p>Estimated Project sprint weeks: 11/27/2023 – 12/4/2023</p>

	<ul style="list-style-type: none"> ○ UST-3 ○ UST-16 	
Sprint 6	<ul style="list-style-type: none"> ★ FINAL STAGES OF TESTING ★ Working on final report and final presentation ★ Last minute clean-ups of UI/implemented user stories 	Estimated Project sprint weeks: 12/4/2023 – 12/13/2023

Challenges:

Something the group struggled with was deciding the best technologies to use in the project. Lots of the technology that is being used such as Auth0 and Vercel has learning curves. Two team members dedicated their time to researching and attempting to implement these tools into the project and test its functionality. It took some trial and error, but we consistently helped each other in debugging and determining if the given technology would be helpful to our project. Another challenge was the desired goals for each sprint. There are things the team would want to accomplish in these two weeks, but sometimes the tasks at hand are more daunting than they seem. The team tries their best to help each other and ensure that the end task moves along smoothly.

Top 5 end user level acceptance test scenarios:

1. The user will be able to login to our website using any of their existing social media accounts or create a new login with their desired email and password and be able to begin the process of documenting their pet's conditions.
2. Users have the ability to add a new pet to their profile with detailed background information on that specific pet
3. Test users ability to edit, delete, and view the existing pets they've inputted
4. Users should be able to add new logs to update the conditions of their pets and save easily onto the website

5. Users should be able to easily share their pet profiles with vets

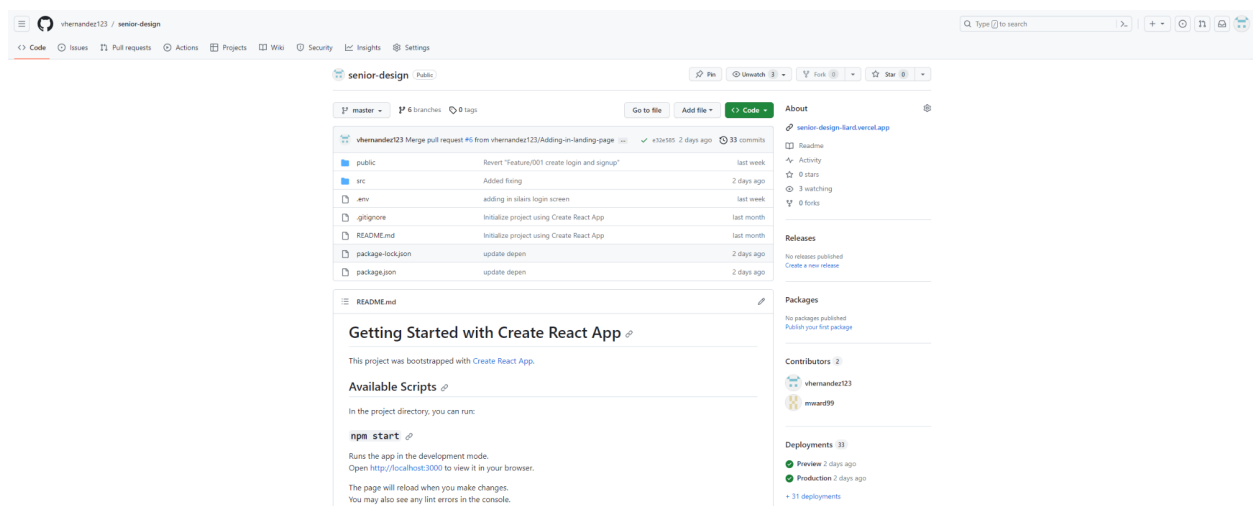
Nonfunctional Requirements:

- Data such as passwords and other personal information must be encrypted
- User authentication must be easy and quick for the user
- This application should be able to run on Chrome, Firefox, Edge, and any other major browser.
- User interface should be easy to use
- Possible figure out a way to regularly back up the data to prevent any loss

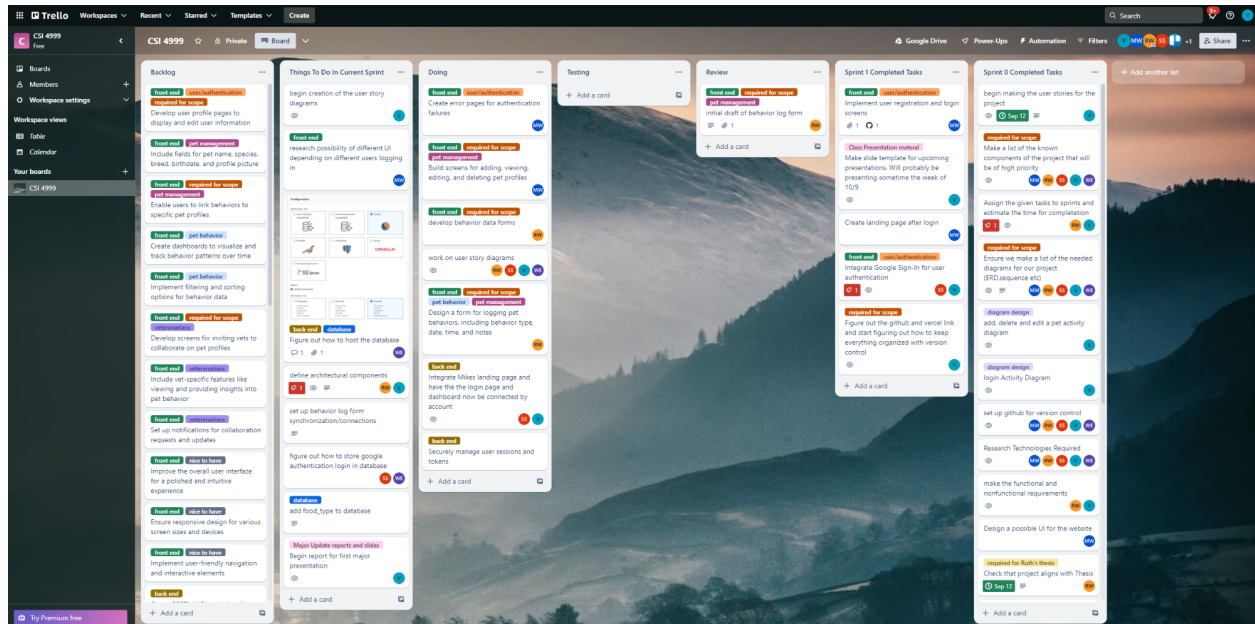
Functional Requirements:

- The user should be able to easily login to the website using their Google account
- The users will be able to add, view, edit, and delete pet profiles
- The users should be able to easily add their pets' behaviors to log and track
- The behaviors logged by the user should be easy to link to a pet in their pet list
- Users could maybe invite their vet to view or collaborate on specific pet profiles

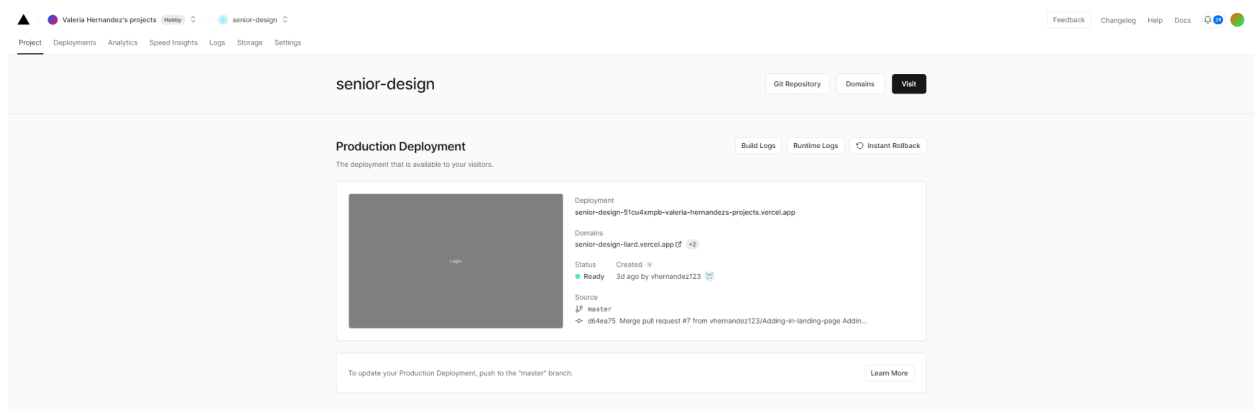
Provide readable images (and explanations) from tools your team are using for your project, which means:



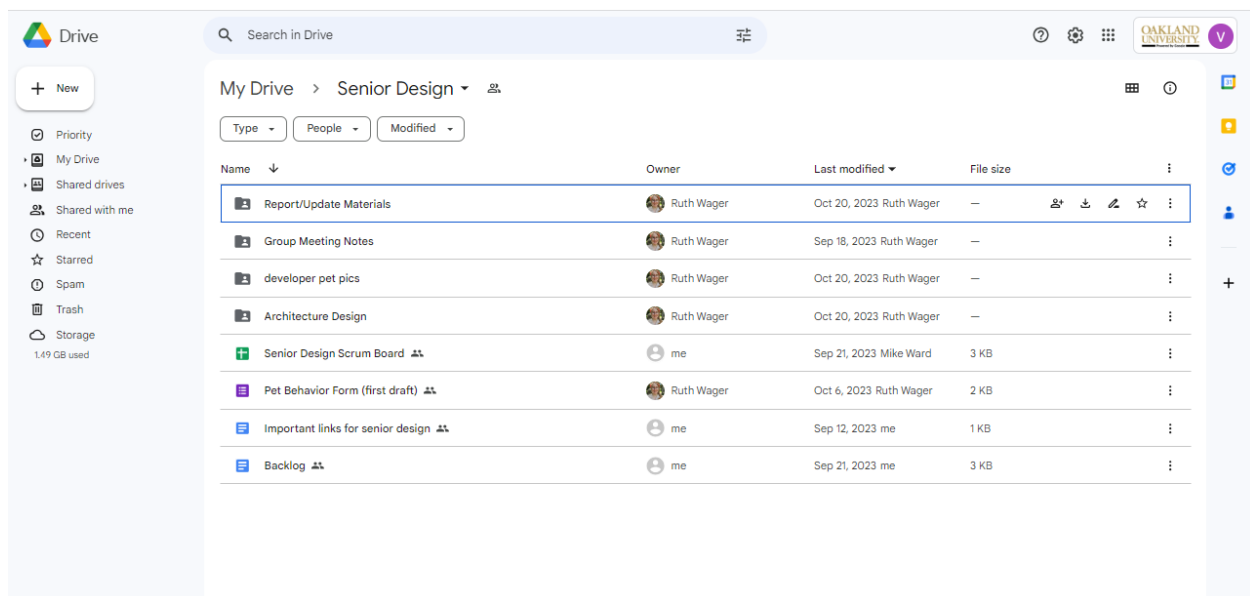
Picture 1: In the above photo, you can see the version control tool we are using. In GitHub we are making individual branches to implement different user stories and ensure they will properly deploy in Vercel



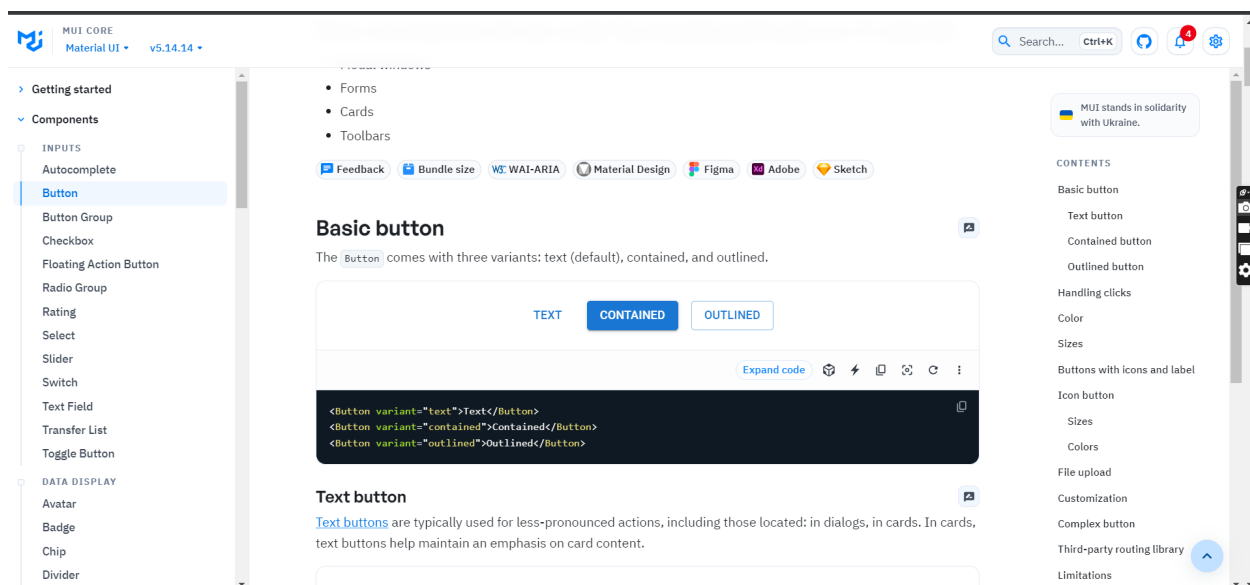
Picture 2: The above photo showcases how we track our progress, Trello allows us to add cards with the tasks we are working on and add labels to that, so we can better organize the tasks we are working on.



Picture 3: The photo above is showcasing the deployment tool we are using to host our website. Vercel is a very important tool as it hosts our website on a domain, and it is directly linked to our GitHub. All changes we make are built and deployed using this platform.



Picture 4: This photo is showcasing the way we are keeping all our documents. Google Drive allows us to easily create and add documents to share among the group.



Picture 5: This photo is showcasing the library that we use (MUI) for components like buttons, tables, alerts, text fields, etc.

CloudWatch (2/4)

Search by metric

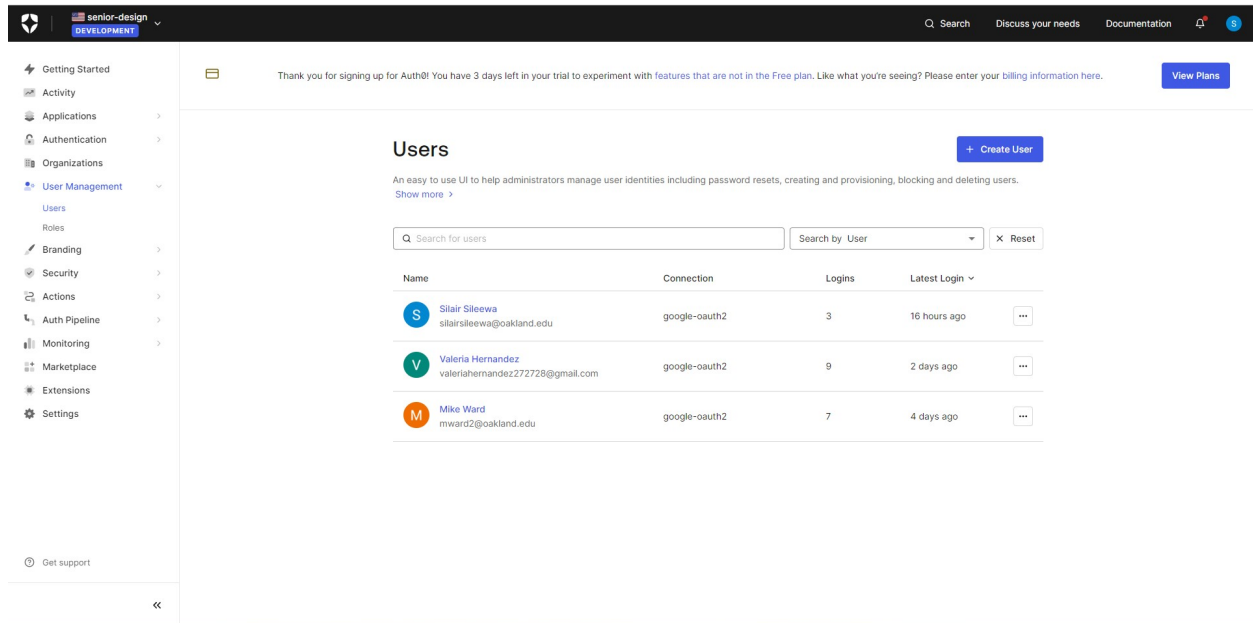
Period: 5 minutes | Add instance to compare | Monitoring

1h 3h 12h 1d 3d 1w Custom UTC timestamp

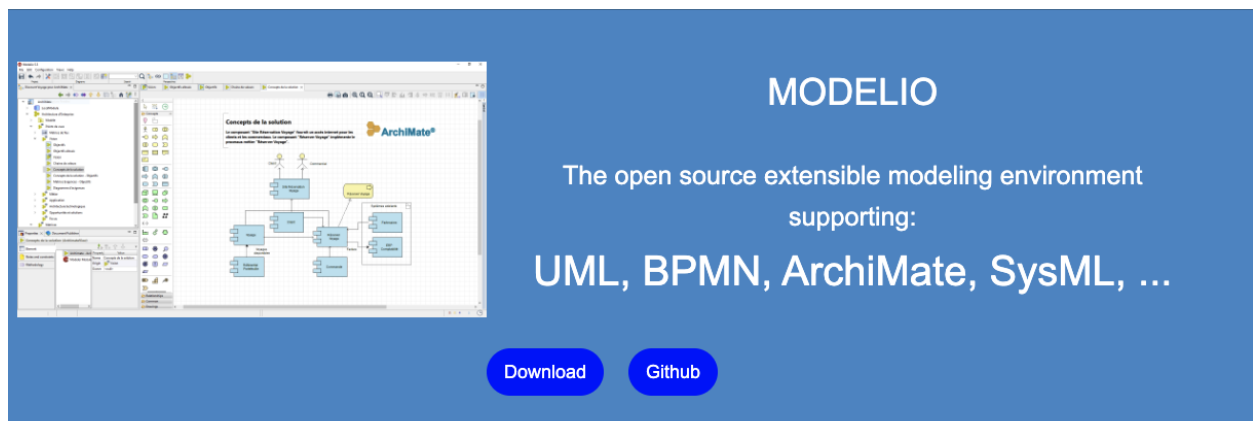
The screenshot displays the AWS CloudWatch console for a specific EC2 instance, 'perl-app-4102'. The interface shows a grid of nine metrics over a 15-minute period. The metrics are as follows:

- BinLogDiskUsage:** A line graph showing disk usage in bytes, fluctuating between approximately 1,000 and 1,500.
- BurstBalance:** A line graph showing the burst balance in percent, which is consistently at 100%.
- CPUCreditUsage:** A line graph showing the count of CPU credits, fluctuating between 0 and 0.5.
- CPUSurplusCreditBalance:** A line graph showing the surplus credit balance, which is consistently at 0.
- CPUUtilization:** A line graph showing the percentage of CPU utilization, fluctuating between 0% and 2%.
- DatabaseConnections:** A line graph showing the number of database connections, which is mostly at 0 but has a sharp spike to approximately 0.8 at the end of the period.
- EBSByteBalance:** A line graph showing the byte balance, which is consistently at 0.
- EBSIOBalance:** A line graph showing the I/O balance, which is consistently at 0.
- FreeableMemory:** A line graph showing the amount of freeable memory in bytes, fluctuating between approximately 40,000 and 60,000.

Picture 7: this photo showcases our AWS cloud RDS console where you can see our performance metrics for our aws server.

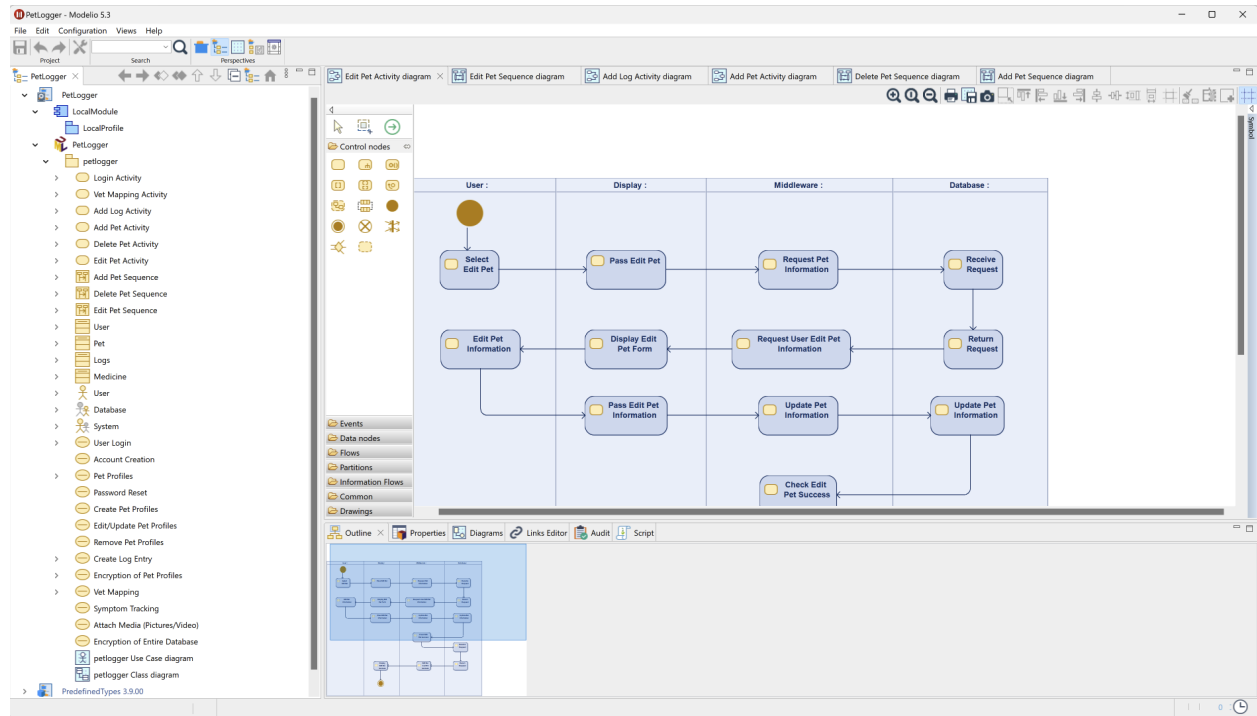


Picture 8: This picture showcases our auth0 dashboard, where you can see we have users registered. Auth0 is our main method for user authentication, it enables us to get users registered as well as handle session tokens while keeping user data safe.

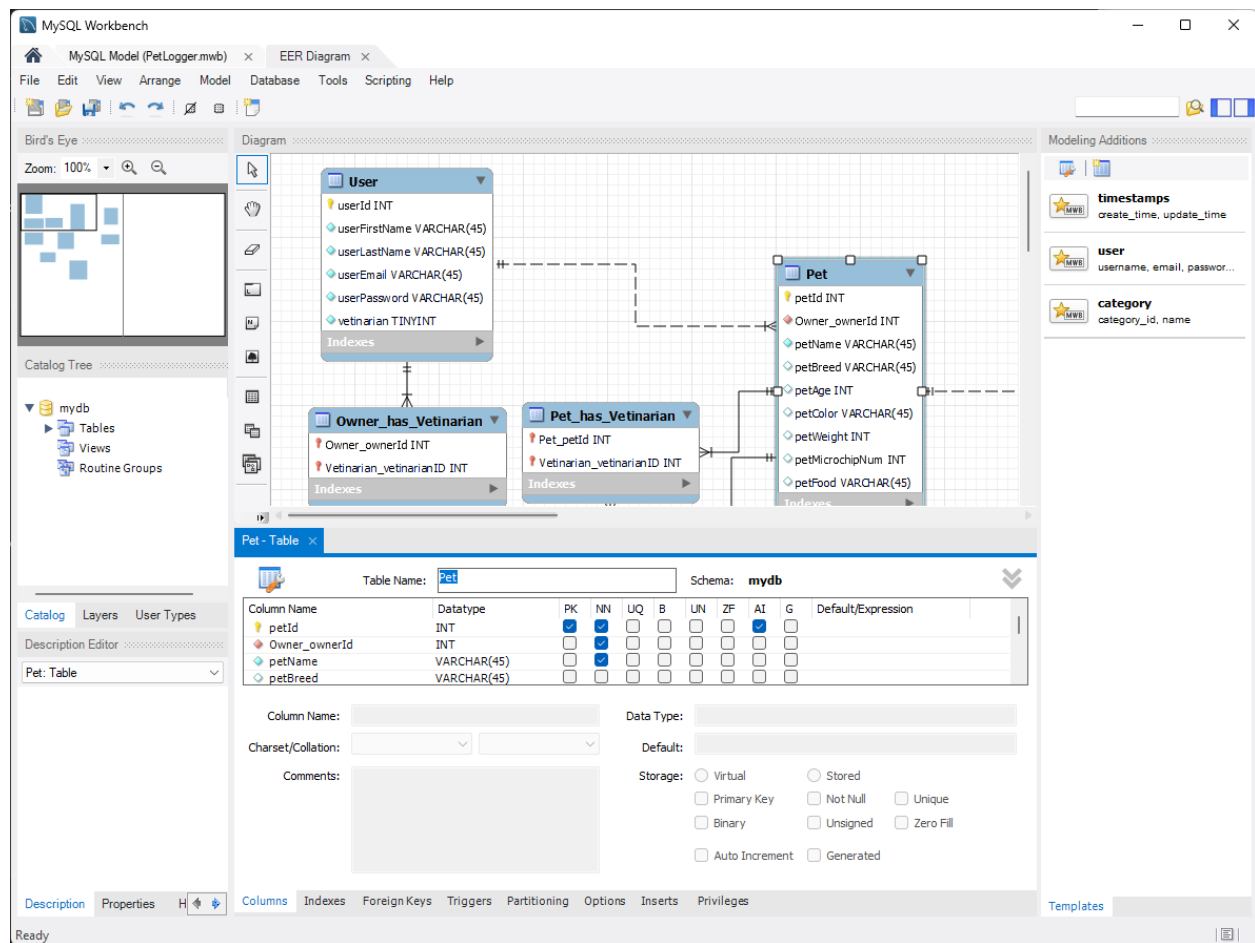


Modelio 5.3.1 Release

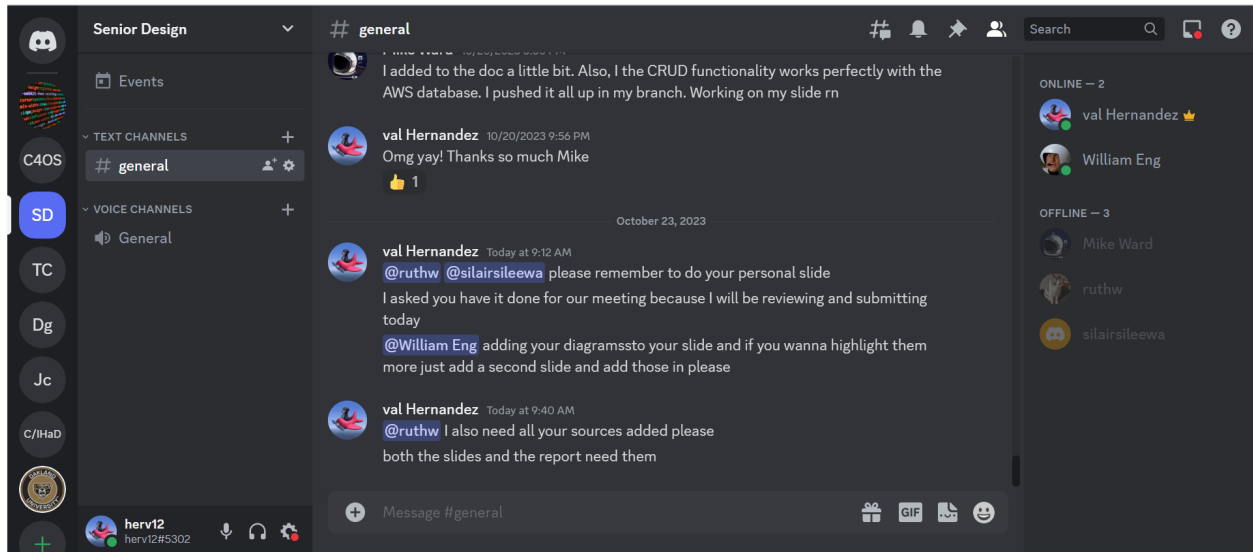
Picture 9: We are using a program called Modelio. It is an open source UML tool. With this tool, we can create all of our UML use case, class, activity, and sequence diagrams.



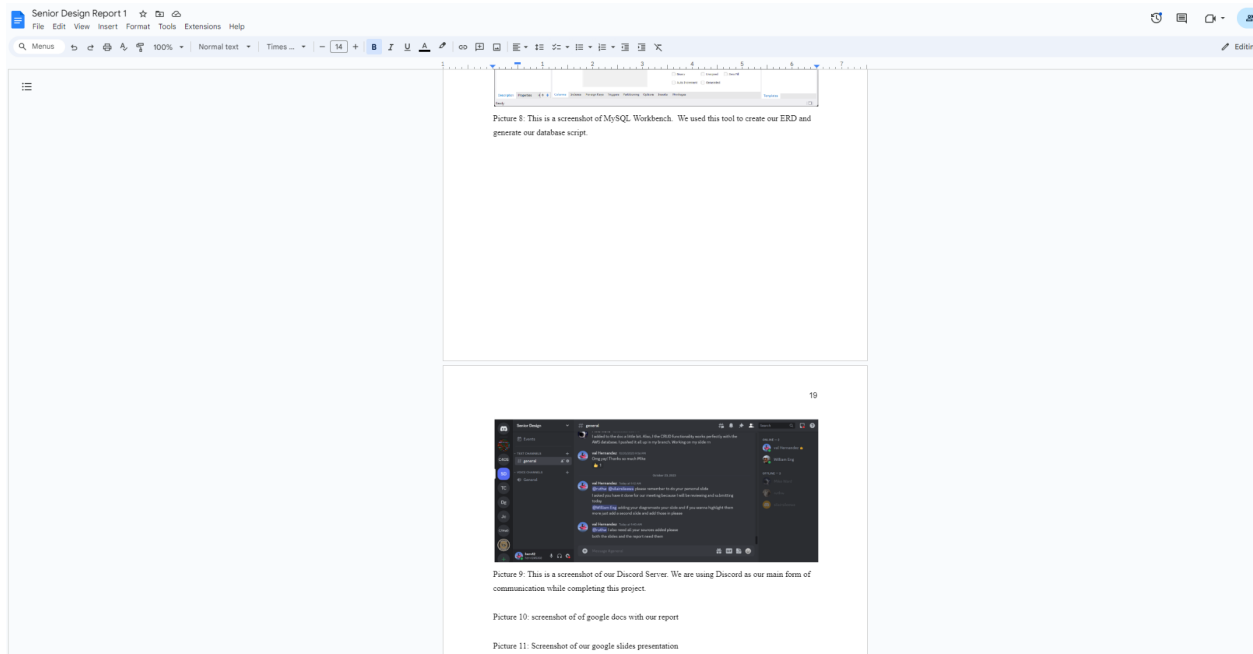
Picture 10: This is a screenshot of our project file in Modelio.



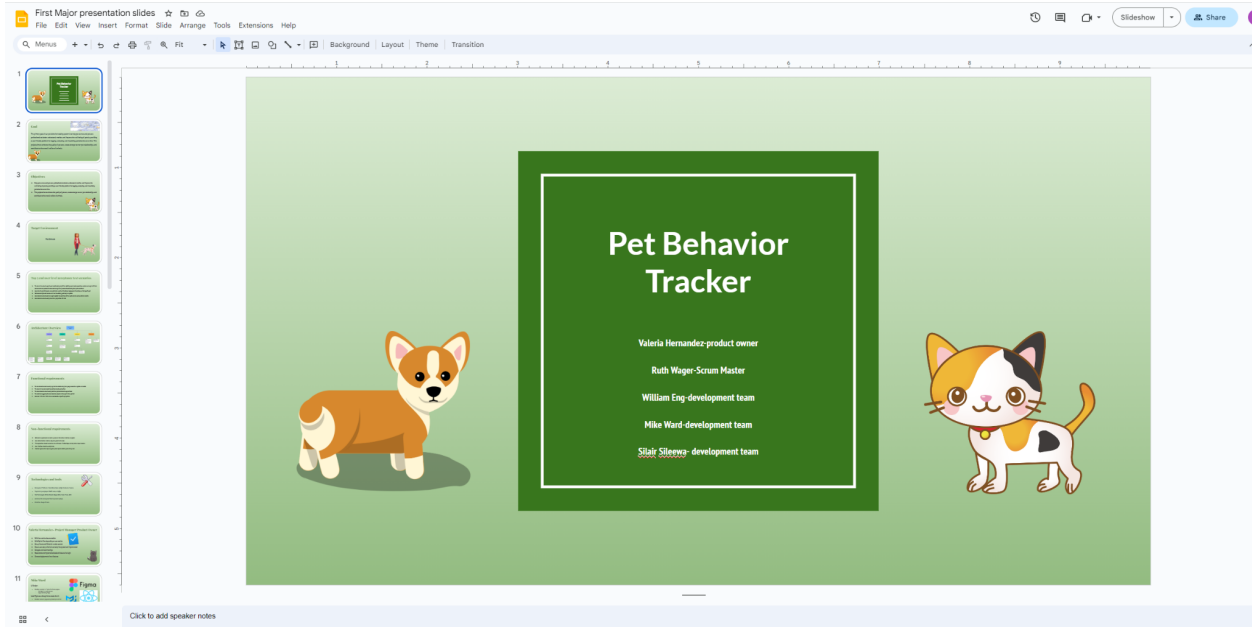
Picture 11: This is a screenshot of MySQL Workbench. We used this tool to create our ERD and generate our database script.



Picture 12: This is a screenshot of our Discord Server. We are using Discord as our main form of communication while completing this project.



Picture 13: The team utilizes Google Docs to collaboratively work on the reports for this class.



Picture 14: The team utilizes Google sheets to collaboratively work on the presentations for this class.

References

- An1, K., & Tilevich, E. (2022). Adaptive Redistribution and Replication to Improve the Responsiveness of Mobile Web Apps. *Journal of Web Engineering*, 21(6), 1-30. doi: 10.13052/jwe1540-9589.2168
- Beginners CRUD Tutorial - ReactJS, MySQL, NodeJS*. (2019, March 9). Beginners CRUD Tutorial - ReactJS, MySQL, NodeJS - YouTube. Retrieved October 24, 2023, from <https://www.youtube.com/watch?v=re3OIOOr9dJI&t=3935s>
- Dent, R. (2022, May 24). *Create Free Database on AWS*. YouTube. Retrieved October 24, 2023, from <https://www.youtube.com/watch?v=UllIPQTpjZ0>
- Google Form output to remote MySQL database*. (2015, February 22). Stack Overflow. Retrieved October 24, 2023, from <https://stackoverflow.com/questions/28655570/google-form-output-to-remote-mysql-database>
- Google OAuth 2.0 Login for React in 5 minutes*. (2022, February 14). YouTube. Retrieved October 24, 2023, from <https://www.youtube.com/watch?v=HtJKUQXmtok>
- `<input>` – *React*. (n.d.). React. Retrieved October 24, 2023, from <https://react.dev/reference/react-dom/components/input>
- kaushal, k. (2021, June 1). *Simple Data Fetching in React with the Fetch API and Axios with Hooks useEffect and useState....* Medium. Retrieved October 24, 2023, from <https://medium.com/geekculture/simple-data-fetching-in-react-with-the-fetch-api-and-axios-with-hooks-useeffect-and-usestate-85d6bd7357c2>
- Malavolta, I., et al. (2023, July). JavaScript Dead Code Identification, Elimination, and Empirical Assessment. *IEEE Transactions on Software Engineering*, 49(7), 3692-3714.

doi: 10.1109/TSE.2023.3267848

Pavić, F., & Brkić, L. (2021). Methods of Improving and Optimizing React Web-applications.

2021 44th International Convention on Information, Communication and Electronic

Technology (MIPRO), 1753-1758. IEEE. doi: 10.23919/MIPRO52101.2021.9596762

React Forms. (n.d.). W3Schools. Retrieved October 24, 2023, from

https://www.w3schools.com/react/react_forms.asp

React.js User Login and Registration with Auth0. (2019, March 9). React.js User Login and

Registration with Auth0 - YouTube. Retrieved October 24, 2023, from

<https://www.youtube.com/watch?v=pAzqscDx580&t=727s>

React + Node.js + Express + MySQL example: Build a CRUD App. (2023, October 16).

BezKoder. Retrieved October 24, 2023, from

<https://www.bezkoder.com/react-node-express-mysql/>

Sigler, C. (2019, November 5). *A Simple Approach to Securing Web Forms | by Chad Sigler |*

Virtru Technology Blog. Medium. Retrieved October 24, 2023, from

<https://medium.com/virtru/a-simple-approach-to-securing-web-forms-801cd5af405e>

Wilson, K. (2023). HTML Forms. In *The Absolute Beginner's Guide to HTML and CSS: A*

Step-by-Step Guide with Examples and Lab Exercises. Apress.

https://doi.org/10.1007/978-1-4842-9250-1_7

<https://www.emailjs.com/docs/examples/reactjs/>

<https://www.youtube.com/watch?v=NgWGIlOjkbs>

