

Team Members: Cole Arduser, Luke Farmer, Sam Nicklaus, Sam Loecke  
ECE:4880 Senior Design  
Lab 3 Report  
Source Code: [HERE](#)  
Website Link: [HERE](#)

# Senior Design Lab 3

## Introduction

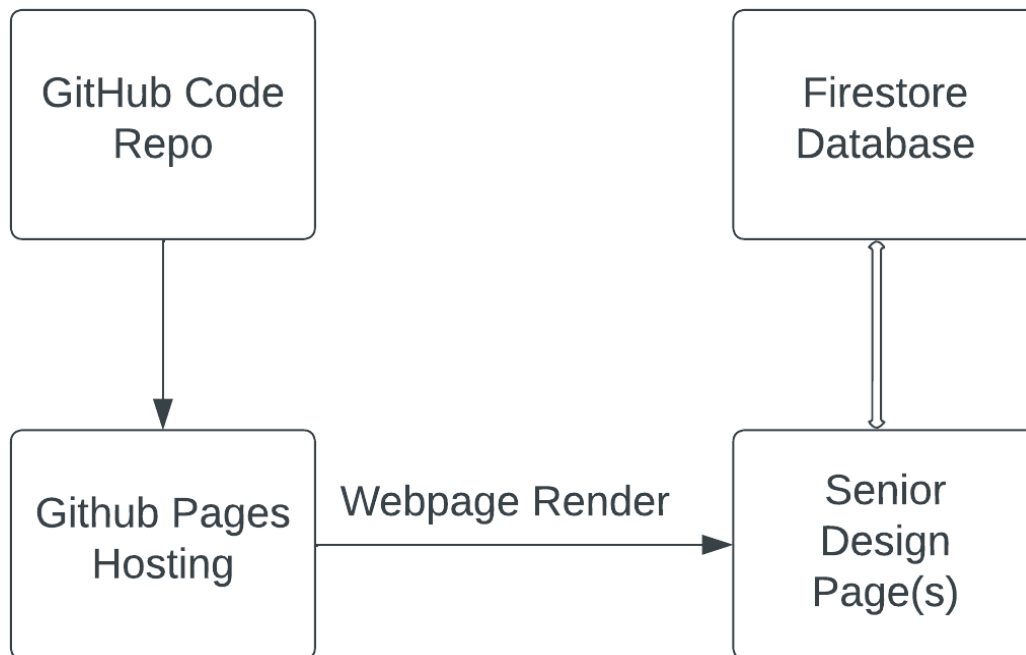
This lab's objective was to provide hands-on experience in developing a website, emphasizing the importance of user interface (UI) and user experience (UX) design. This lab's challenge was to design and implement a portfolio website for each team member to showcase their interests and projects outside of class. The website needed to be accessible publicly, with functionalities for user comments and login/logout.

## Design Documentation

The design that we used for our solution to our lab prompt uses both Github Pages along with Google Firebase for web hosting and database storage respectively. With these two services, we were able to produce a web application with login and comment functionality along with an easy to use user interface. Our initial design started with a team meeting discussing how we wanted to lay out the website and made initial Figma designs for all of our pages (Refer to Appendix & References for the initial designs).

### **Functionality Overview:**

For our front-end software, we used Vue.js 3 which is a popular front-end framework based on Javascript, HTML, and CSS. We chose Vue for the flexibility and variety of add-on libraries that can be used to enhance the user experience. For the database design, we opted to use Firestore as we implemented our own comment and login system.



**Figure 1: High-Level Technology Flowchart showing overall relationship**

(default)	User	Default
+ Start collection ColeComments LukeComments SamLComments SamNComments User >	+ Add document 2cGDXqnYyYB05V6rpUh8 7RD6V03HAVqn6zRjCAJq A00YwoH8yProq6QEXEUp CFcxvsDoY8GjRlWXpMbt Default > J4j2tPQ8JzBqq2z5EHs9 K1CyQI1Kqb05V5F360e7 L4UK1i2D4zdJMRM06KGo R1nIWZdfWQfKH50KIflf U9HAWk9zjcLfeIQy1XkL	+ Start collection + Add field pass: "2023FallLab3!" user: "ECE4880"

**Figure 2: User tables in Firebase for Login Functionality**

Figure 2 represents our user tables which handles the storage of usernames and passwords for the login functionality where both “user” and “pass” are both strings with a random hash as the primary key. We use this in the “Login.vue” file of our web application that first checks that the user entered a valid username and password length, then checks those details in the database to either log the user in or create an account. If no account is found

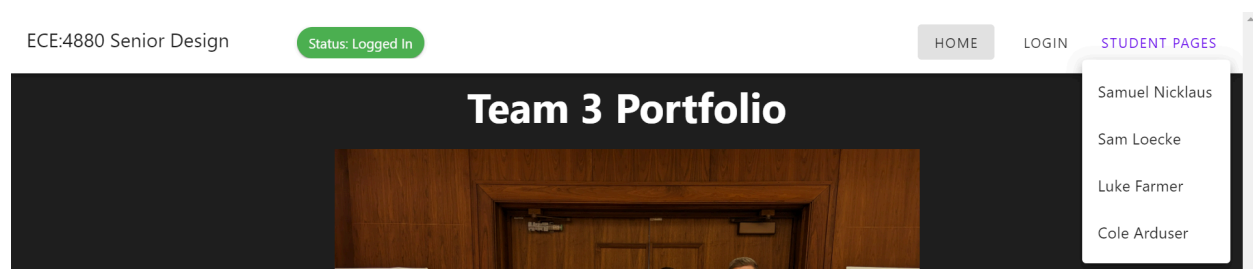
when trying to log in the user is prompted on screen accordingly. After a user is successfully logged in, a cookie is stored in the browser that allows the user to refresh the page and remain logged in for a specified period of time before the cookie expires. Refer to our Login.vue file [here](#) on line 27 for the start of our login functions.

(default)	SamLComments	Default
+ Start collection	+ Add document	+ Start collection
<ul style="list-style-type: none"> <li>ColeComments</li> <li>LukeComments</li> <li><b>SamLComments</b></li> <li>SamNComments</li> <li>User</li> </ul>	<ul style="list-style-type: none"> <li><b>Default</b></li> <li>Ny7jWnMmMs230cf1Zgm1</li> <li>XezyYLeCNtc5Xk7s0ReQ</li> <li>gU1FW2SaIH0C4c6ZFQtt</li> <li>yJtgy4hxePS3KcJvEvN3</li> <li>zNu1S79pt1z3YnfsTXh6</li> </ul>	<ul style="list-style-type: none"> <li>+ Add field</li> <li>comment : "Cool page Sam LI"</li> <li>user : "ECE4880"</li> </ul>

**Figure 3: Comments table for Sam Loecke's personal page**

Figure 3 represents one of our four comment tables which associates a comment with a given user with a random hash as the primary key. We use this information in all of our personal pages to display and add comments.

Our website design starts with our persistent navigation bar displayed at the top of every page which provides links to each section such as login, home, and the private pages. The links to the private pages are designed in a way where if the user is not logged in and attempts to access a page, they will be redirected to the home screen with a prompt to log in first.

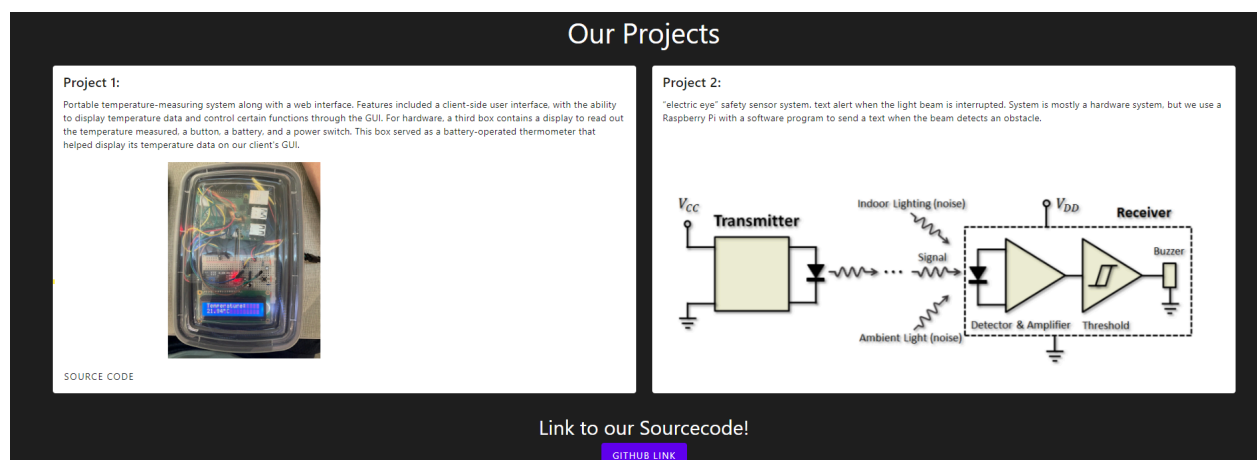


**Figure 4: Navigation Bar**

The main landing page for our website is the home page, which is designed to give viewers information about us and our projects we have completed over the course of the semester. Users are first greeted with an image of us along with our names, followed by our first two projects and links to learn more about them.

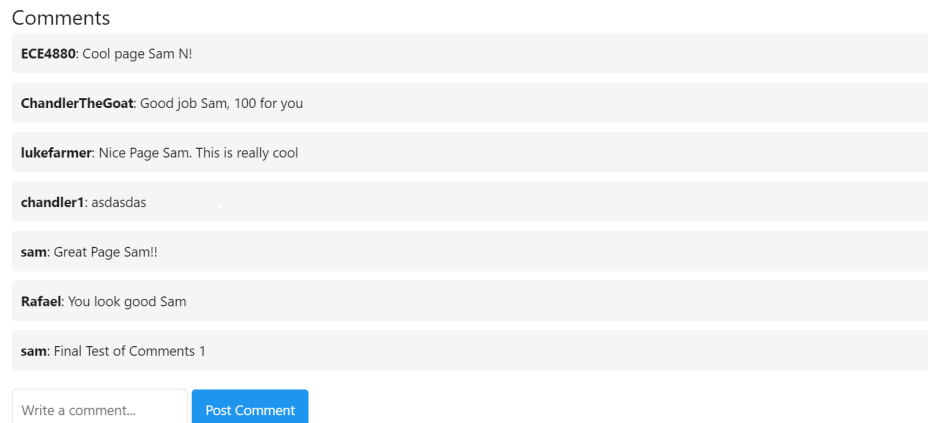


**Figure 5: Home page team photo and names**



**Figure 6: Home page projects section along with link to the website's source code**

Similar to the home page, each teammate has their own unique page designed to display information about the individual. At the bottom of every teammates page is a comments section where users can leave feedback for that page.



**Figure 7: Comments section at the bottom of every private team page**

### **Components Overview:**

To build our website, we used the Vuetify library on top of Vuejs to build out our personal pages. Vuetify components allow us to further customize our pages without making components from scratch. One example of this is the carousel component that scrolls through a set number of images.

```
<v-carousel style="max-width: 90vw;" cycle>
  <v-carousel-item src="./SamNPhoto2.JPEG" cover></v-carousel-item>
  <v-carousel-item src="./SamNPhoto1.jpg" cover></v-carousel-item>
  <v-carousel-item src="./SamNPhoto3.jpg" cover></v-carousel-item>
</v-carousel>
```

**Figure 8: Code Snippet of Carousel Component in Samn.vue**

The code in figure 8 highlights the simplicity that adding in a library adds as compared to making a rotating carousel of images by hand. Other examples of Vuetify components we used were buttons, dropdown tabs, image sheets, and more.

### **Styling and Accessibility:**

Vue.js is built with accessibility in mind, with CSS we can scale all of our components to fit a variety of screen sizes. Many of the components on our page utilize background colors, different text sizes, and unique animations to highlight when something is clickable or not. On top of that, almost all components utilize viewport width and viewport height which fits the component size to a percentage of the screen. We also make sure to use high contrast text on the desired background color for readability of text.

## Design Process and Experimentation

### Software Design:

The software design considerations were:

Database Storage Choice

Language/Front-end framework choice

Login Support

Comment Support

Cookies Support

#### Database Storage Choice

Solution	Implemented?	Reason
Google Firebase (Firestore)	Yes	Firestore is an all in one solution for both hosting and integration in Vuejs. Making it a good choice when accessing the database from the website directly.
PostgreSQL	No	Hosting a local PostgreSQL server was an option we considered but realized that would be overkill for the application we are creating.

#### Language/Front-end framework choice

Solution	Implemented?	Reason
Vue.js	Yes	Teammates were more familiar with Vue.js vs other

		frameworks. Vue allows a good mix of functionality and customizability.
React	No	Teammates were not familiar with react and documentation looked complicated to start.
Pure HTML, CSS, Javascript	No	HTML would be a simple solution, but we wanted more customizability with our website to make it look modern.

#### Login Support

Solution	Implemented?	Reason
Create from scratch	Yes	Firestore made it easy to implement and allowed for more customizability in putting username requirements in.
Use off the shelf solution	No	More complicated implementation, especially if using OAuth tokens and such.

#### Comment Support

Solution	Implemented?	Reason
Create from scratch	Yes	Firestore made it easy to implement as well and allowed for the username we use to login to post comments as opposed to using google or github to login.
Use off the shelf solution	No	Using off the shelf comments would mean that users would have to login twice to be able to post a comment.

#### Cookies Support

Solution	Implemented?	Reason
Use Cookies	Yes	Quality of life improvement so that the user can refresh the page after logging in and still be logged in.

Don't use cookies	No	Users would have to login every time they visit the website.
-------------------	----	--

## Test Report

Software Test	Result
1) Website opens and displays the home page (both locally and publicly on GitHub pages)	Pass
2) Website contains an accessible team introduction page	Pass
3) Website contains comment boxes that can input and display a comment	Pass
4) The website contains embedded figures inside all pages	Pass
5) A user can successfully create an account	Pass
6) A user can successfully log in after creating an account	Pass
7) A user is prompted with "Incorrect information" if an a user enters details in incorrectly	Pass
8) The website contains three personal pages for each member with an embedded figure and brief introduction	Pass
9) All external links in the website are link to the correct sources	Pass
10) Website contains headers on each page that can help the user access other pages	Pass
11) The codebase is stored inside a Github Repository	Pass
12) All pages have proper routing to other pages	Pass



13) All pages have a clean UI/UX design	Pass
---	------

## Project Retrospective

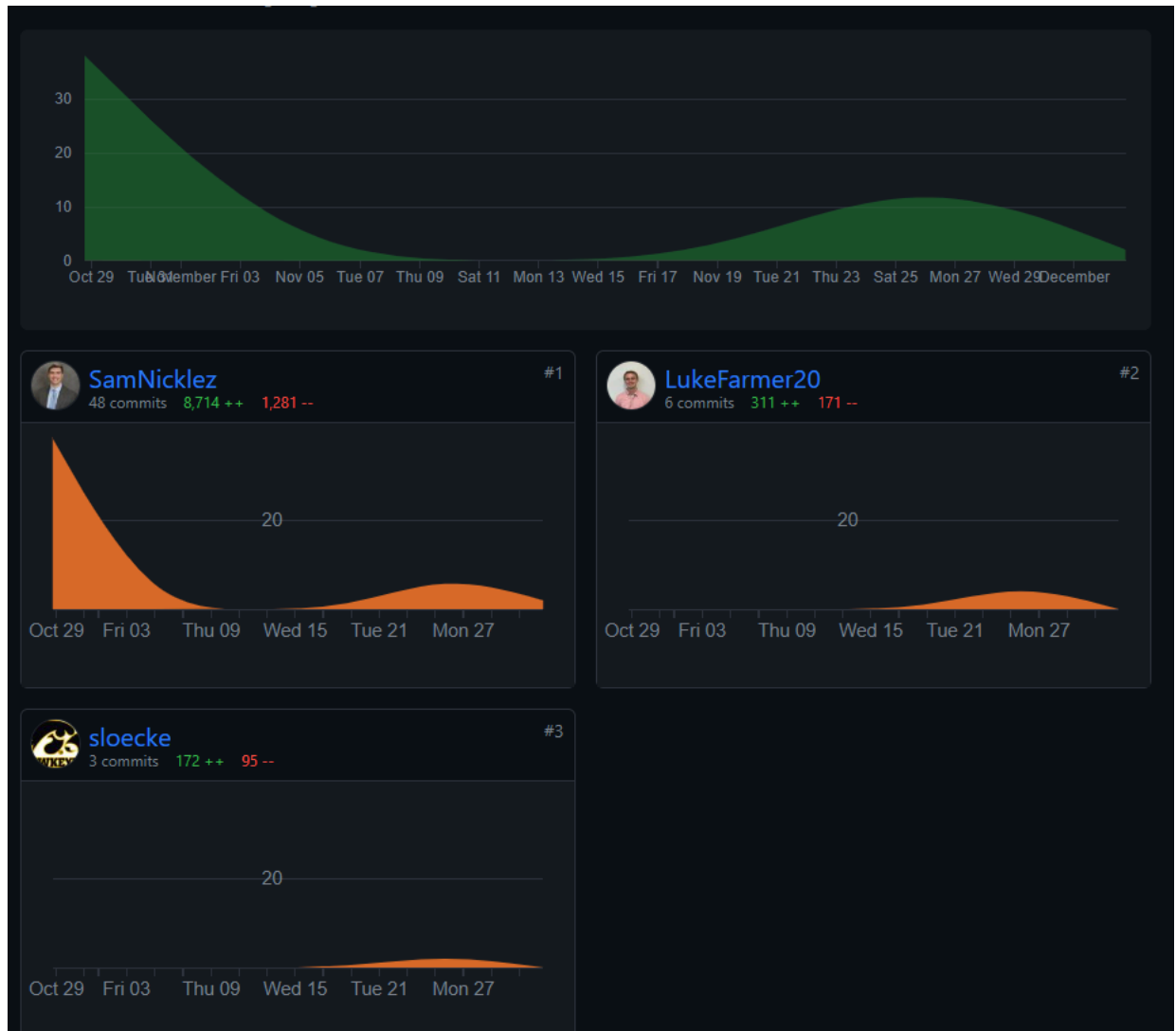
In lab 3 we successfully completed and fulfilled all requirements given to us while staying on time and on budget. We played to our strengths, as Sam Nicklaus and Cole Arduser are CSE and took the lead on this lab. Sam Nicklaus had experience using the Vue framework and the firebase database. He took the lead on teaching the rest of the group how to work with the new frameworks. Everyone completed the vue files for their own personal pages with their personalized content. We split the project up as follows:

Samuel Nicklaus - Team lead, project set up, comment feature, login implementation

Cole Arduser - UI design planning, personal page, database design and set up

Luke Farmer - UI design planning, personal page

Sam Loecke - UI design planning, personal page



**Figure 9: Code Snippet of GitHub Contributions Page**

For some reason, since Cole Arduser made a separate branch instead of committing directly to main, his commits are not showing up on the github contributions, Coles PR had 126 additions and 80 deletions. Feel free to verify the commits [here](#).

The work distribution for this lab was more lopsided since it was software heavy. CSE individuals worked heavily on the backend implementation of the comment and login features as well as the firebase database while the EE group members learned from the CSE members and completed their personal pages with unique content. We implemented a Waterfall methodology as we were already given strict requirements and guidelines. This helped a lot as we started our

project by having a long meeting going over how we wanted to design the project. What we were most concerned with was the knowledge gap between individuals for web design and backend server implementation.

Our biggest challenge for this lab was learning a new framework for everyone besides Sam Nicklaus. We got through this challenge by reading the documentation and learning from Sam Nicklaus who had experience with the Vue framework as well as using firebase. It was a good experience for everyone to learn a new tool to use in the future. Another big challenge we faced was managing our images when trying to build and deploy the project. When run locally the program ran fine, but when we tried to build and deploy it we had errors with how we were saving our images. Eventually, we overcame this issue by searching through Vue documentation and some trial and error.

	Nov 5th - 10th	Nov 10th - 15th	Nov 15th - 20th	Nov 20th - 25th	Nov 25th - 30th	Nov 30th - 3rd
<i>Planning</i>						
<i>Design</i>						
<i>Implementation</i>						
<i>Testing</i>						
<i>Lab Report</i>						

**Figure 10: Gantt Chart for Timeline**

For the most part, we were able to stick to our plan and timeline. Actually we were ahead of schedule for the majority of the lab with more time to add a few more features on each team member's personal pages.

## Conclusion

Overall, the goal of this lab is to create a publicly hosted website with a portfolio for our team. Each team member had to have their own private portfolio page with at least one embedded figure. We had to have a login/logout function to see the private pages and a comment feature on each member's page. We accomplished this goal and were able to meet all of the specifications. In the process, we all improved our ability to make publicly hosted websites and work with Github in an efficient manner.

## Appendix & References

"Vue Router." Vue Router, <https://router.vuejs.org/>

"Vue Test Utils." Vue Test Utils, <https://test-utils.vuejs.org/>

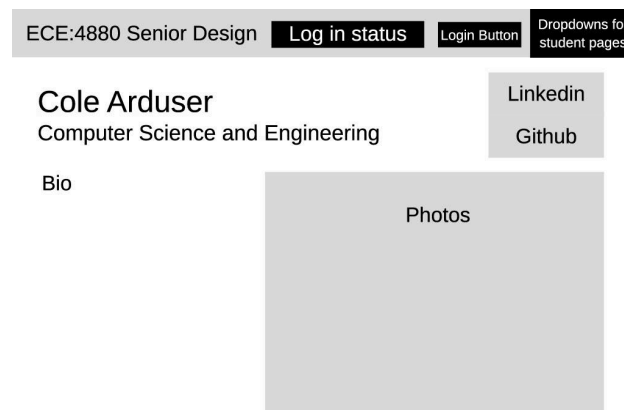
"Vue.js Devtools." GitHub, <https://github.com/vuejs/devtools>

"Vuetify." Vuetify, <https://vuetifyjs.com/>

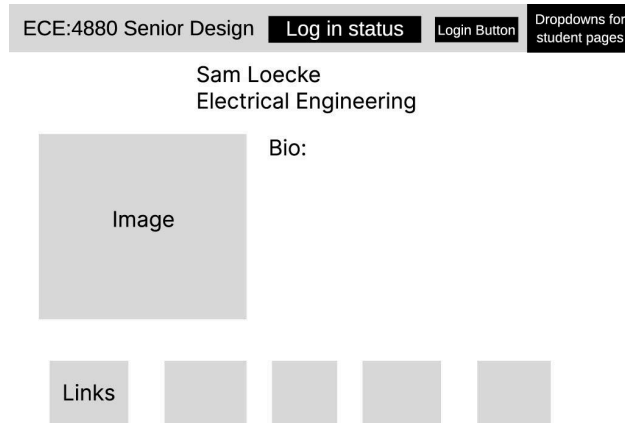
"Vue.js." Vue.js, <https://vuejs.org/>

"vue3-cookies." npm, <https://www.npmjs.com/package/vue3-cookies>

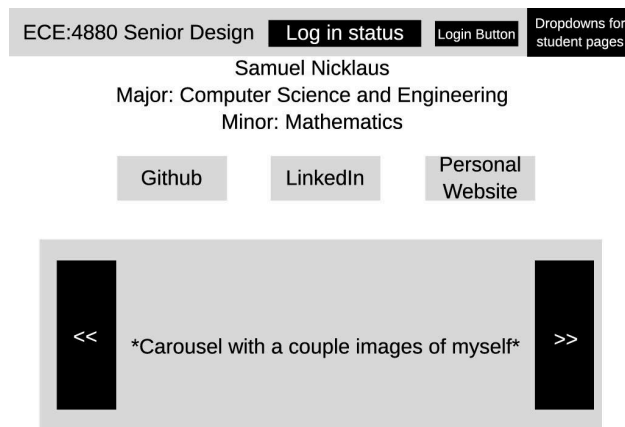
"Vitest." Anthony Fu, <https://vitest.dev/>



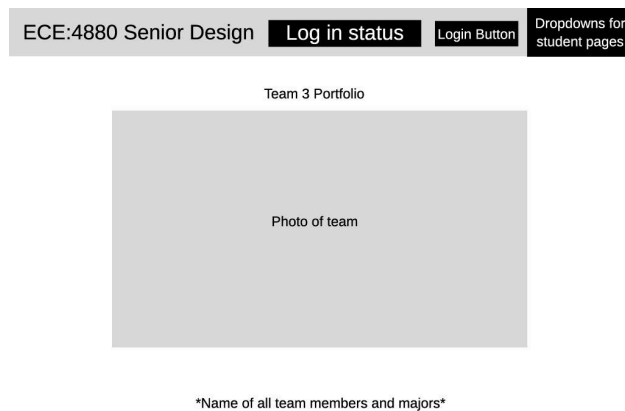
**Figure 11: Cole Arduser personal page initial figma design**



**Figure 12: Sam Loecke personal page initial figma design**



**Figure 13: Sam Nicklaus personal page initial figma design**



**Figure 14: Home landing page initial figma design**

Username

Password

LOG IN

CREATE ACCOUNT

***Figure 15: Login page initial figma design***