# CYBERSHELF

Celia Cameron, Owen Dyer, Jordan Okada, Leo Tellez, Ashton Wilbern
Section 011, Fall 2023

## Project Description

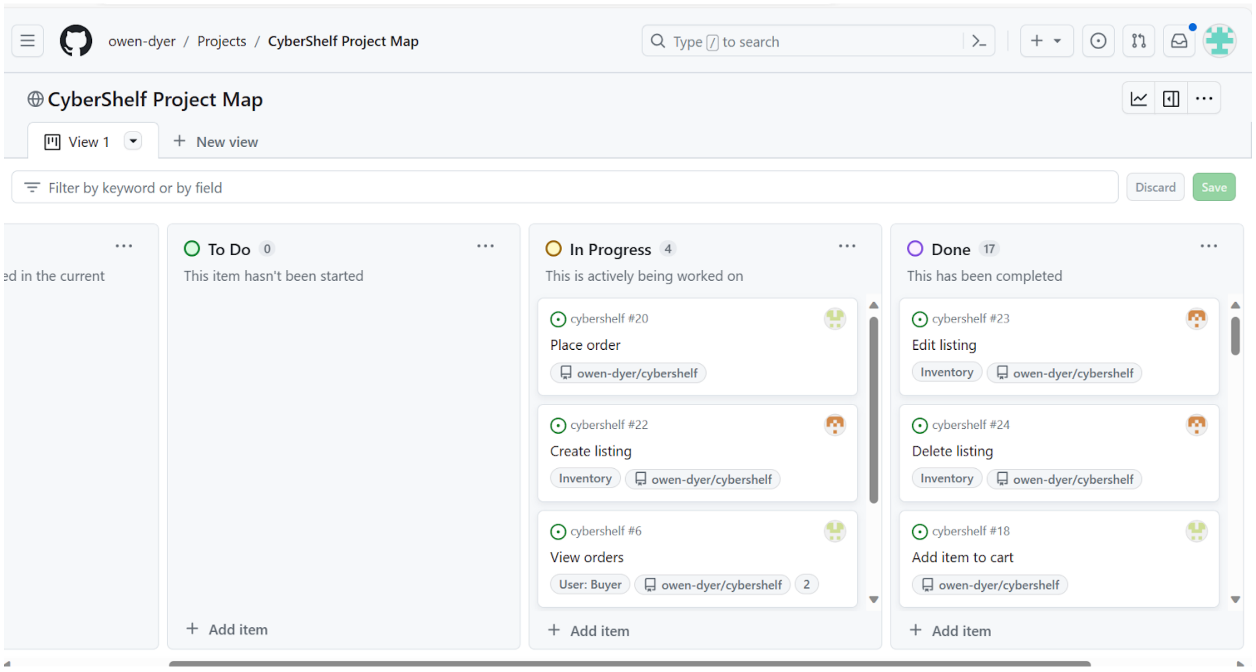
CyberShelf is an online storefront that allows users to navigate through and purchase a plethora of items. The platform contains a wide variety of products with prices that customers will love. As a guest user, customers can navigate through the inventory. If they wish to purchase any products, they must make an account with CyberShelf. They can then add products to their cart and checkout. They can change their account details and view their past orders at any time.

The website has a minimalist aesthetic which makes finding products an easy and efficient task. Without the ads and distractions which many other websites have, users can focus on the products CyberShelf provides. The variety of products that CyberShelf provides will continue to expand as more listings are added.

CyberShelf utilizes a microservice architecture, as opposed to a monolith architecture. Rather than having one centralized server manage all of the backend functionality, it utilizes multiple servers, as well as multiple databases. This allows each service to run independently, meaning that if one service were to go down, none of the other services would be affected, and the rest of the website would remain functional.

## Project Tracker

CyberShelf Project Map


owen-dyer / Projects / CyberShelf Project Map
Type / to search
CyberShelf Project Map
View 1
New view
Filter by keyword or by field
Discard
Save

ed in the current

To Do 0
This item hasn't been started
+ Add item

In Progress 4
This is actively being worked on

cybershelf #20
Place order
owen-dyer/cybershelf

cybershelf #22
Create listing
Inventory | owen-dyer/cybershelf

cybershelf #6
View orders
User: Buyer owen-dyer/cybershelf 2

+ Add item

Done 17
This has been completed

cybershelf #23
Edit listing
Inventory | owen-dyer/cybershelf

cybershelf #24
Delete listing
Inventory | owen-dyer/cybershelf

cybershelf #18
Add item to cart
owen-dyer/cybershelf

+ Add item

## Video

YouTube Link

## VCS

Github Repository

## Contributions

### Celia Cameron

My contributions to the project include work on the front end EJS pages. Initially, I did work with Bootstrap that was not used in the final product, as we chose to work with Tailwind instead. I drafted many pages as well as some routes for the order database. I designed the logo we used, and I helped to test the website, specifically through the registration and sign in test cases using Mocha and Chai. I also did a lot of work on the project presentation slides and the project report.

### Owen Dyer

My contributions to the project include configuring routes for NGINX, implementing functional code for user registration, authentication and authorization, as well as for the inventory, cart, and order servers, and implementing test cases for API routes. Also, I implemented JavaScript to run on the browser to handle interactions with various API endpoints, and to improve UX throughout the website. I also wrote templates for many of the pages on our website and styled them using TailwindCSS. Also, I helped deploy our application to Azure.

### Jordan Okada

My contributions to the group project consisted of mainly front end development. I worked on designs for the home page, account page, cart page, a thank you page, and a few other pages that were not included in the final product. I worked with tailwind and bootstrap to develop the pages. I also spent a majority of my time working with my TA and team members to debug our product, specifically to get our docker containers working properly with our advanced structure.

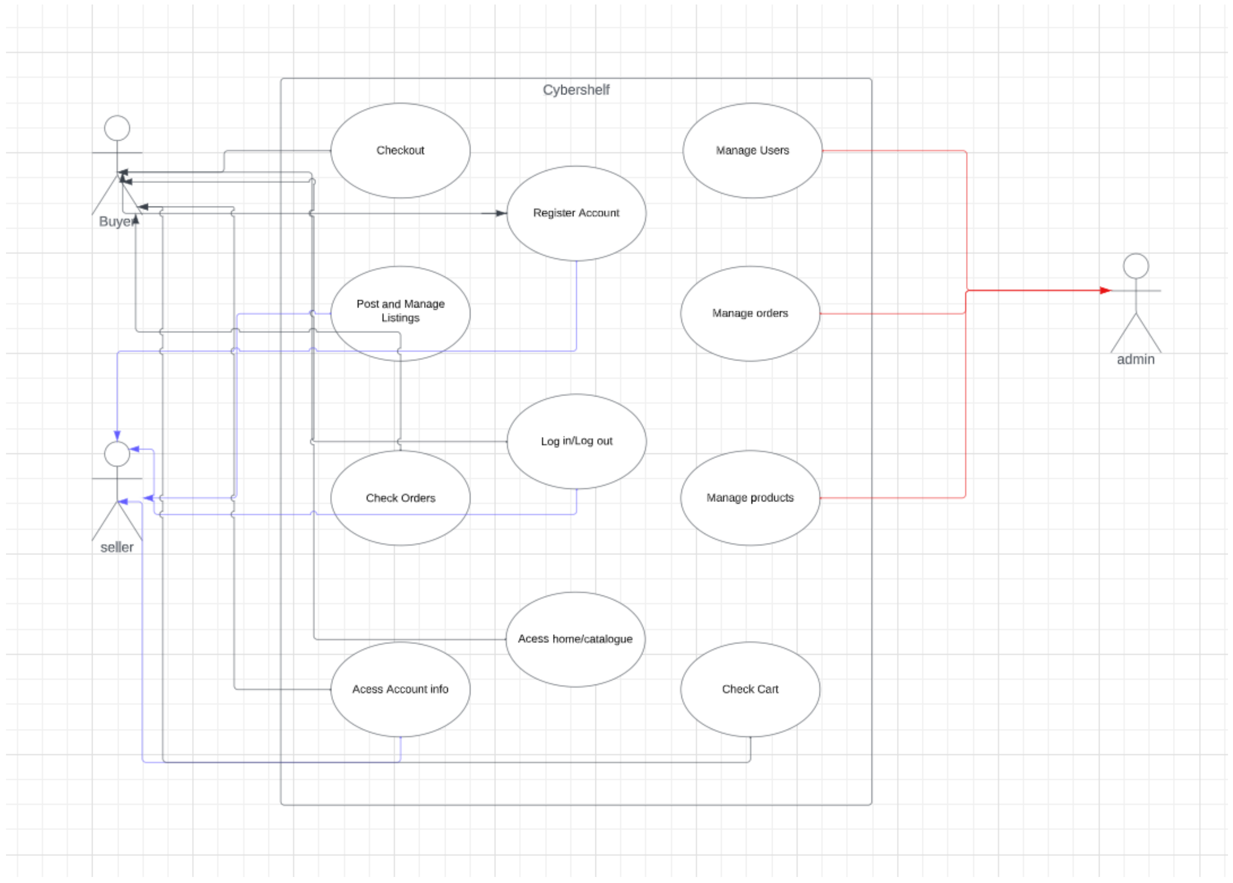
## Leo Tellez

My contributions involved creating the initial cart EJS page and SQL table, I also functioned as a scribe taking the meeting notes during every meeting and assisting with deliverables such as the Use case diagram, a Wireframe, and the creation of the video for the report, I worked to understand the micro-service system to the best of my ability and gained knowledge using these systems that I was able to explain how it works and what it allowed us to do during our presentation.

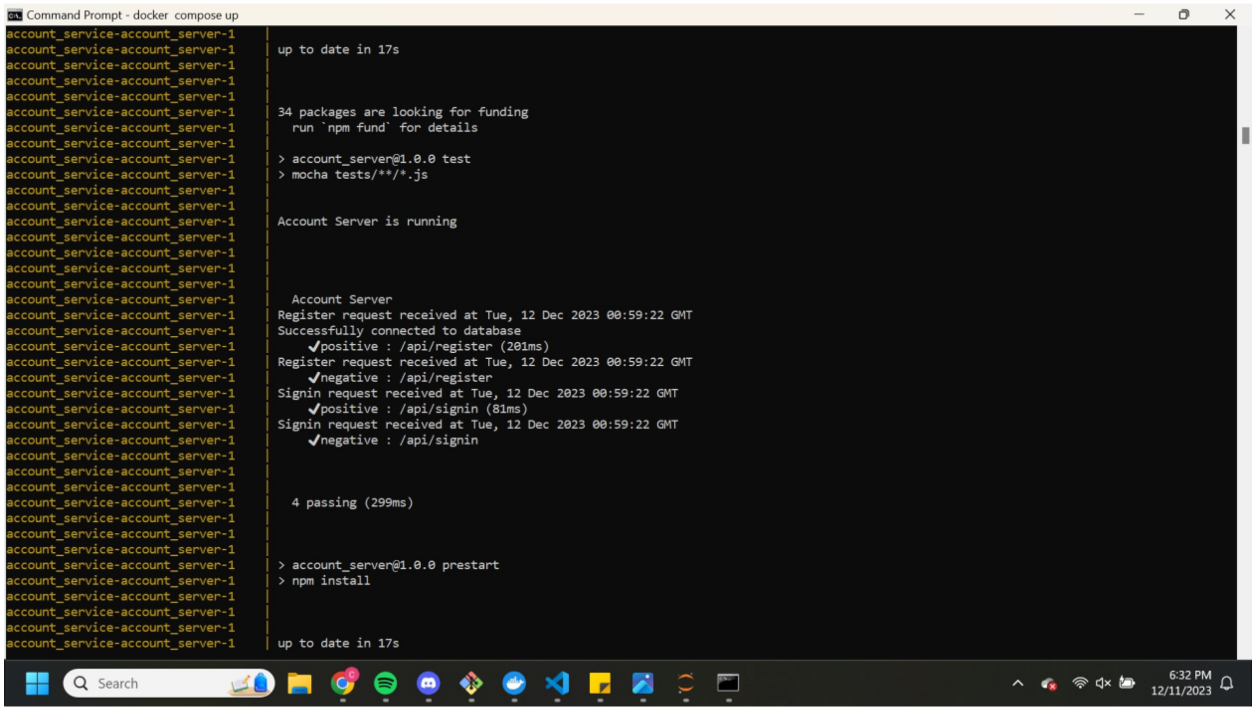
## Ashton Wilbern

My contributions to the group project include designing an accurate network architecture diagram, deploying our application to the virtual machine on Microsoft Azure for Lab 13, and initializing the SQL tables/databases for the order and cart services. Additionally, I took an instrumental role in setting up frontend foundations, focusing on the homepage and API routes, influenced by Lab 9 and transitioning to tailwind for styling. I also took part in extensive troubleshooting to get our separate docker containers to run properly to ensure that our application was able to function.

## Use Case Diagram



## Test Results



```
account_service-account_server-1  |
account_service-account_server-1  | up to date in 17s
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  | 34 packages are looking for funding
account_service-account_server-1  |   run `npm fund` for details
account_service-account_server-1  |
account_service-account_server-1  | > account_server@1.0.0 test
account_service-account_server-1  | > mocha tests/**/*.js
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  | Account Server is running
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |   Account Server
account_service-account_server-1  | Register request received at Tue, 12 Dec 2023 00:59:22 GMT
account_service-account_server-1  | Successfully connected to database
account_service-account_server-1  |     ✓positive : /api/register (201ms)
account_service-account_server-1  | Register request received at Tue, 12 Dec 2023 00:59:22 GMT
account_service-account_server-1  |     ✓negative : /api/register
account_service-account_server-1  | Signin request received at Tue, 12 Dec 2023 00:59:22 GMT
account_service-account_server-1  |     ✓positive : /api/signin (81ms)
account_service-account_server-1  | Signin request received at Tue, 12 Dec 2023 00:59:22 GMT
account_service-account_server-1  |     ✓negative : /api/signin
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |   4 passing (299ms)
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  | > account_server@1.0.0 prestart
account_service-account_server-1  | > npm install
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  |
account_service-account_server-1  | up to date in 17s
```

**Observations**

1. Users cannot checkout without logging into and/or registering an account.
- The user first attempted to click the add to cart button on a product, because they had no other options for attempting to purchase the product. A pop-up appeared reading, “Please sign in to add items to cart.” They then went through the process of registering, logging in, adding the item to their cart, and then filling out their information before hitting checkout.
- Their behavior is consistent with our use case. We wanted to ensure that the user had to go through the proper procedure before checking out, and when attempting to “cut corners” the application stopped them, providing the necessary information to checkout correctly.
- The behavior was consistent with our expectations, so we did not need to make any changes.
2. A customer cannot purchase items without filling out all of the mandatory fields in the checkout form.
- After adding their item to the cart and going to the checkout page, the user attempted to submit the order without filling out any information. They then attempted to purchase by only filling out one field. They then realized that every field must be filled out before submitting the order.

- They behaved like this to attempt to quickly cut corners to get to the end results, but in the real world, they would have missed out on important information that is necessary to provide.
- This test case worked exactly as we wanted it to and their behavior aligned with our use case perfectly. We tested if a user could check out by filling out none of the required fields, and if they could check out by having only filled out one of the required fields. This test produced the desired results: the user could only check out by having completed all of the required fields.
- In the future, we may consider implementing ways to ensure that users are inputting correct credit card information and other credentials, rather than just confirming that they entered information at all.

3. A seller cannot list items for sale without filling out all of the mandatory fields in the product form.

- Unfortunately we were unable to test this use case in our final application. We made a decision to omit the ability to list items on the website, and made it into an application that only allowed for users to purchase listed items.

4. A customer cannot add an item to cart without first selecting an item and a quantity

- The user completed this task very quickly. They simply added the item to their cart by clicking on the button “add to cart” listed under each item.
- They completed the action in this way because there was only one button listed under every item, so it was very easy to complete this task.
- Although it worked differently than we originally envisioned, the way that the user completed this task aligned with what we wanted. Originally we envisioned a button that would add the item to cart, and then prompt the user to ask for the quantity. Our final version automatically adds the item to the cart, and if the user wants to increase the quantity, they will simply press the add to cart button multiple times.
- We did change the final product from our original idea, which I mentioned earlier, however we did not deviate from our plan after completing the UAT testing, as they completed the task perfectly.

## Deployment

CyberShelf

- Deployed using Azure