

Tech Library Project Report

Team Members: Andrew Connell, Ben Kohav, Charlotte Gorgemans, Roxanne Manthy, Ryan Hannigan, Sean Shi

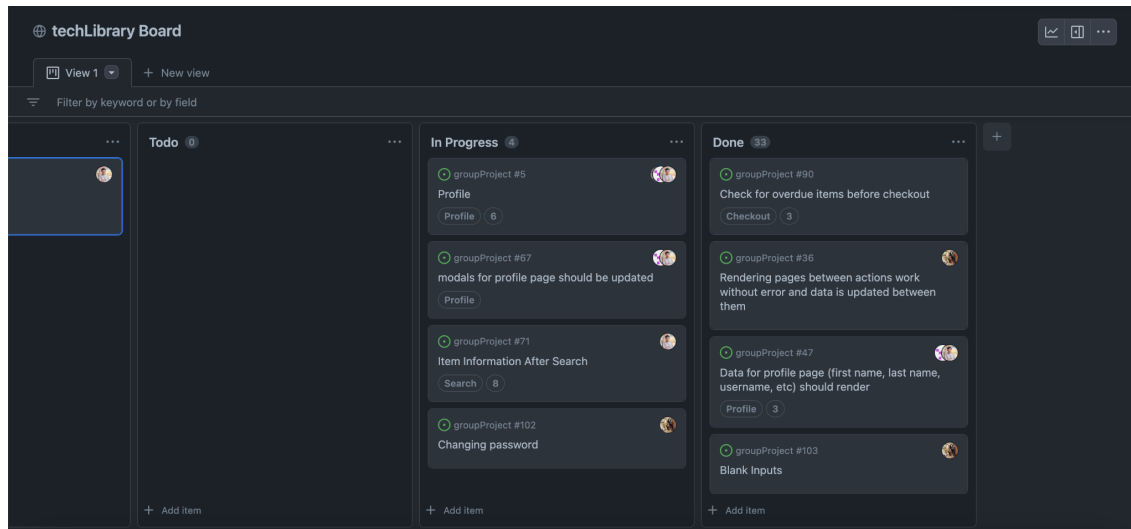
Project Description

Overview: Our project, an online tech library, provides a forum for students to borrow essential school materials and equipment for a day's use. Imagine the following scenario: Anna is a commuter student, and upon arriving on campus one day, she realizes that she left her computer charger at home. Our tech library allows Anna to connect with a resource center on campus to borrow a charger for the day.

Functionality: Once logged in, the user is redirected to our website's home page, where a list of popular products (i.e. charger, computer mouse, apple products) is available for browsing. A search bar in our nav bar proves for more targeted browsing, should the user want to search for a specific product. When the user has identified a product they would like to borrow, they can select the item, adding it to their cart, which redirects them to the separate cart page. The cart page allows the user to check out the product, and provides a location in which the item can be picked up. If the user loses an item, a list of suggested replacement options are available on our website for the user to select from. Finally, the profile page allows for the user to change their contact information (i.e. first/last name, email, school year).

Project Tracker

To keep track of each team member's tasks and our overall progress on the tech library's development, we implemented a Github project, connected to our GitHub repository. A link to our project board via GitHub can be accessed here: [\[GitHub Repo Link\]](#). A screenshot of our project board is also shown below:



Video

A video created by the developers further explaining our tech library's functionalities:

[\[Video Link\]](#)

VCS

A link to our GitHub repository, containing all of our project's code: [\[GitHub Repo Link\]](#)

Individual Contributions

Andrew: I created the page and api for the home page. I helped integrate our pages together, and find and fix bugs for the website. I created an entire cart system that used local storage. (Note: We later changed the cart system to storing information in the database.) I made a use case diagram and later updated it.

Ben: For this product, I contributed on various pages of the website throughout the course of our development. I helped create our login and registration page, as well as incorporate the database calls that allow for relevant information to be presented in the profile page. I added functionality to our search page allowing the user to filter their results by availability; updated search UI from a table of results to individual cards that neatly show each item and its descriptors. I also cleaned up the website UI on profile, cart, and search page, allowing for a cleaner experience for the user.

Charlotte: I established the base functionality and design of the profile page. I created the modals (intaking new information), and made revisions to the database. I worked with Roxanne and Ben to fine tune the profile page. I redesigned our login and register pages, by implementing a new UI to match all pages. I created our presentation and completed the application overview, tools/methodologies (with Sean), challenges, and future enhancements (with Ben) sections. I wrote our report, completing the project description, project tracker, test results (with Andrew), and deployment sections. I also created the sitemap, and logout, login, and registration wireframes.

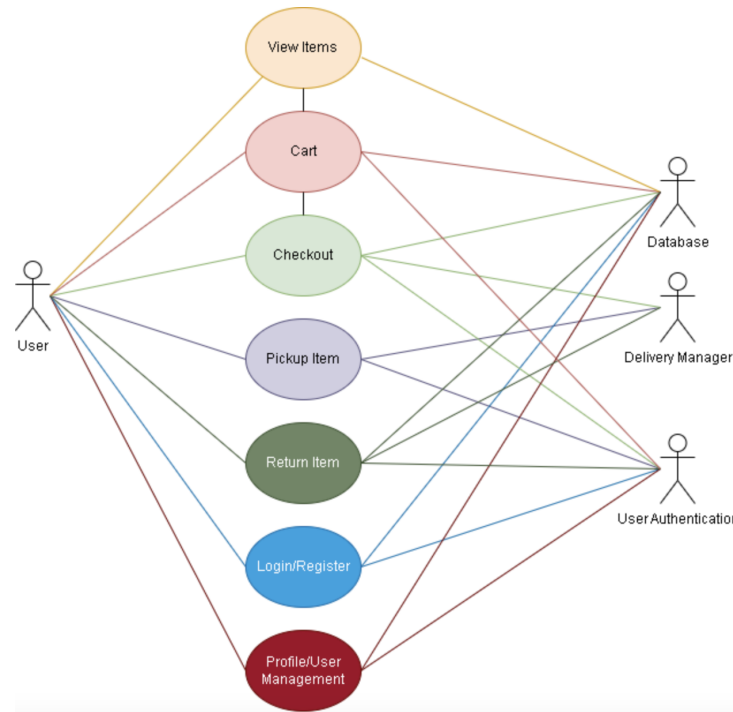
Roxanne: I created the layout of our website and ensured that there were no errors between rendering each page. I helped get our database working so that we could read data to the website. I was responsible for the editability of the profile page. I maked sure that the user could edit aspects of their profile. I created individual forms that allowed data to be saved within the session/database so that data is consistent throughout the session. Next I made sure that the user can edit their password securlyr. Finally I ensured the cleanliness/readability of our UI.

Ryan: I helped design the database along with the insert.sql. I collaborated with Sean to design the database ER diagram.

Sean: I created the api and query to search for matches in the database for the search page and also designed the initial search page. I also implemented the cart stored in the database and adding and removing items to/from it. I also added the cart/checkout page and put in return buttons on the profile page so users could return items. I also added a check for overdue items before checkout so the user could not checkout if they had overdue items and a check in cart so they would be notified if an item they wanted had already been checked out. I helped add tooltips to search results, fix bugs with insert.sql, and implement search filtering.

Case Diagram:

A case diagram is a visual representation of all the possible interactions a user can have with a system. This diagram is a high-level overview of the interactions between the system and the actors with the system. An image of our case diagram can be viewed below:



Our case diagram has also been pushed to our GitHub repository, and is accessible at this link:

[\[Case Diagram Link\]](#)

Test Results

Prior to formally testing our application, we established a test plan and implementation strategy. This report includes the edgecases that we brainstormed and experimented with during the testing process. [\[Testing Plan Link\]](#)

Register/Login/Logout

- When a user enters an input that is too long for the database to handle, it doesn't register the user and lets the user know they couldn't be registered.
- If nothing is entered into the register page and the user clicks register then the user is unable to register until they have filled out the required fields.

- When a user registers they are redirected to the login page, and the username and password they registered with allows them to log in
- When a user logs out, it will still save changed data within the session and the user is able to log in again, but the user does not have access to the rest of the website until they log in.

Checkout:

- If a user has an item in the cart and leaves their computer and logs into a different one, then their cart will be saved.
- If user B checks out an item in user A's cart, then user A cannot check out user B's new item.
- User A cannot add an item to cart that user B has already checked out. It will display "Item unavailable".
- If a user has overdue items, attempting to checkout items will return an error message until they return their overdue items.
- If a user checks out an item and returns it the use counter for that item goes up. This works even if the page is reloaded.

Profile:

- When the user is trying to change their personal information on the profile page (with the exception of password), it will reveal a modal which the user then uses to alter their data and the page then rerenders with updated information.
- If a user changes their password and then logs out, the user is able to log in with their new password. This is also the case for username

Search:

- If the user types in a string in the search bar which matches a name(ie laptop) the search page renders with all items which have that as part of their name
- If the user types in nothing to the search bar and hits enter, the search page will render nothing
- If the user types in an item not in the database, the search page will render no results.
- If the user chooses the only available option on the search page and hits search, the search page will be rerendered with only items which have not been checked out.

Home:

- Attempting to access the home page or any other page without logging in redirects the user to the login page.
- All pages contain a navigation bar with links to other pages which redirect them to these pages.
- When the user logs in, they are redirected to a home page with the most popular items types and buttons which redirect you to a page showing those items.
- The home page's about us (as tested by someone unfamiliar with the website) has instructions which effectively guide someone on how to use the website and its purpose.

Deployment

If we were to deploy the app, we were planning to follow the instructions of lab 12, using the CU Boulder's IaaS. We chose this form of deployment because it does not require any purchasing of additional hardware and is a cloud based server that allows for easy organization and development. The first step of this process would be to SSH into CU's available IaaS. Next would be to add permissions to this IaaS, downloading a rootless docker, and adding a .env file in order to access the website correctly. Following this process, we would need to make sure that our port is exposed to the public port so that it is accessible. Making sure that docker is running ensures that this process will be successful.

- *The app must be live, working, and accessible to your TA. We were instructed by our TA, Aishwarya Satwani, that was not necessary for us to deploy our website*