Book Hooked

Helen Dupree, Ben Burkhalter, Zach Trainor, Joseph Sembower, Delan Huang, Sayat **Project Description:**

Book Hooked is a way to keep track of what books you own and in what formats. It also has a basic way to look for new books you might want to read. Everyone can track their books in their own unique account. The home page shows books based on user preferences that they can browse. In the library page users can add new books to the library with just title and author. Then users can see the cover and other information about the books sorted by the format they own the book in. On the user page the user can see some of the information they signed up with as well as all of their books sorted by newest added. Included in Book Hooked is a GUI where the users can see books in a way that displays the title and book cover in a way that is similar to Netflix.

Project Tracker: N/A

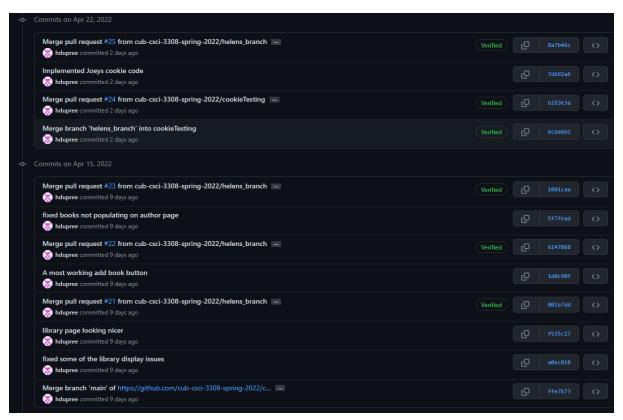
Video: Video

VSC: Github Repository

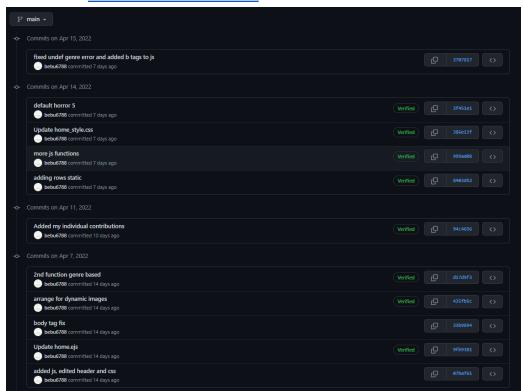
Contributions:

Helen: github contributions

I worked on a few different things. I created all of the userpage and a majority of the library page. A large amount of the work for both pages was creating the partial for showing book information. The books partial involved getting book data from the database and creating calls to the Google Books api. For the library page created the backend for the 'add books' button, allowing new books to be added to the user's profile. I also helped with the nodejs code for the home page. I finished implementing the cookies that Joey started.



Ben: Github Main Contributions



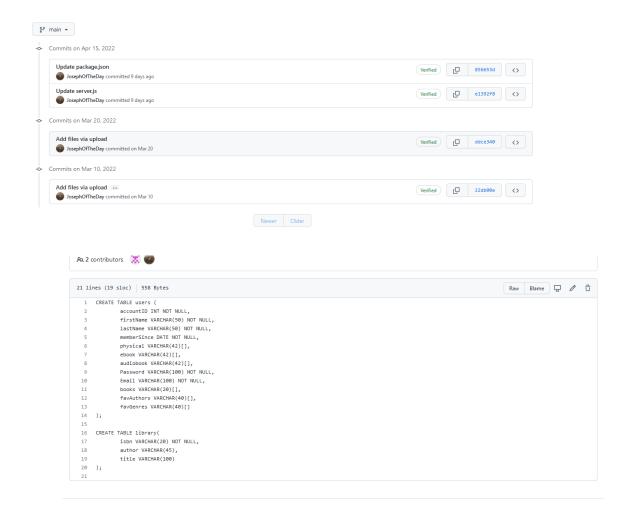
My big contribution to this project was the home page. This page was my pride and joy! I used HTML bootstrap to implement a Netflix style all-in-one user experience with rows of sliders displaying book covers. When the user sees a book they're interested in they can click the image and a new tab will open with a link to purchase it. Eventually the html became dynamic through JavaScript functions that populated the home page based on favorite genres, or favorite authors using AJAX Google Books API calls. After these functions were working, the homepage was converted to EJS in order to talk to the NodeJS server to access user information from the database. This allows a user to input a couple of their favorite authors and genres and after signing up, the homepage will populate books based on their input preferences. Helen helped EJS talk to node, since backend stuff was a major struggle throughout the semester. But other than that, the home page is all me!

Zach: github contributions

I worked on creating the landing page and getting that up and running. I worked on creating the server.js, converting everything to ejs, restructuring file directories, creating header, footer, and navbar partials, assisted with database commands. I also helped create the library page foundation. I worked with node.js, HTML/CSS, plenty of JavaScript and Postgres.

Joseph: github contributions

I worked on starting the database and working through bugs we had throughout the project. This included talking with the group about how the database should work and taking steps to implement it. However, these latter steps were more based on Helen and Zach and taking the time to implement what needed to be done with the final steps of the database. Beyond that, I took a lot of time trying to get cookies to work for the website but could never come to a proper conclusion. This part of the project was finished by Helen. During the whole process I did take the time to setup work based plans such as trello other milestones, but this couldn't have been done without the help of my other team members.



Sayat: github contributions

Nothing

Delan: github contributions,
Architecture diagram

Deployment: Unfortunately we were unable to get the website functioning properly on Heroku. To run the code navigate to the all project code components folder and docker-compose up.