

Title: BookIt

Who: Carson Gross, Brandon Warren, Gavin Zura, Dennis Pozidaev, Justin Nguyen
Zach Driver

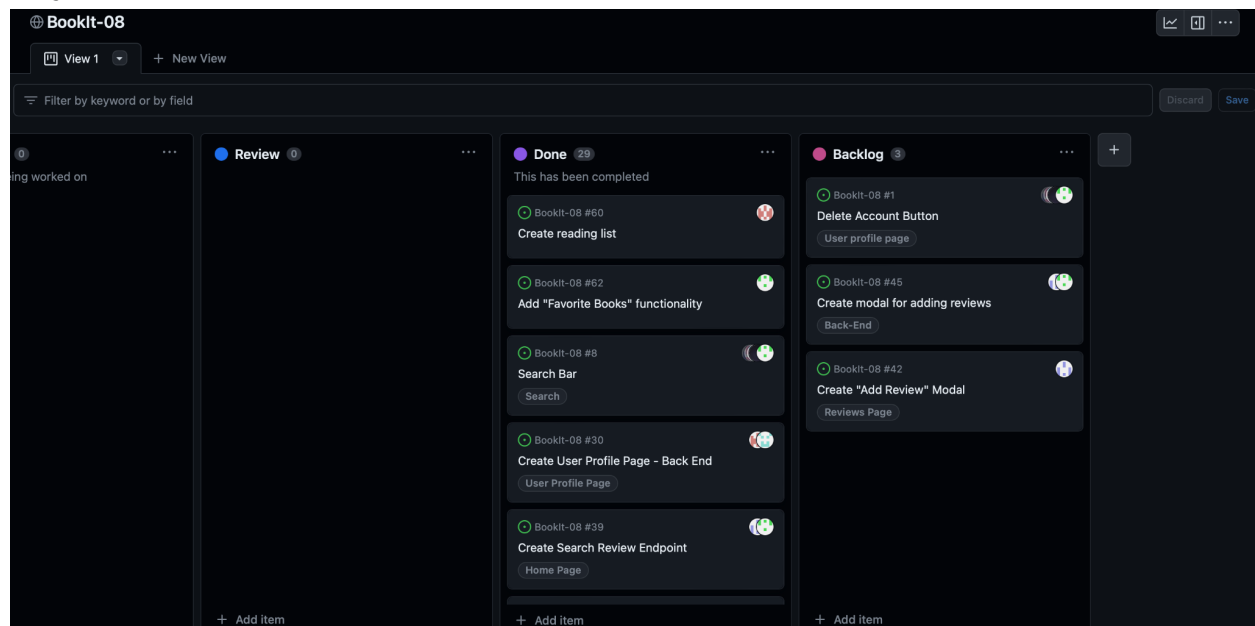
Project Description:

BookIt is a site for all book lovers. It allows its users to search for books they might like, add books to reading lists, and view reviews on books of different users. Users are able to post reviews for books available in the database (provided by an external API). These reviews will be searchable by the books they're reviewing, the user who posted the review, and the point-rating associated with the review.

Each user has a profile which they can update with new information. Profiles contain a username, age, location, favorite book, and an about me section. The User can also visit pages for their personal reviews of books, their reading list of books that seem interesting to them, and liked books that they have read in the past. The site itself displays a list of all books currently obtained from the external API. A user can click on a book they like to see more information about it such as its subject, the number of downloads, and its average user rating.

Users have to register and log in to utilize the functionality of the website.

Project Tracker:



Video:

<https://cuboulder.zoom.us/rec/share/5FoUnldzp7qJm04cSRy8JxHOdVhTMT2FVsaPLczeYp4PcsMy7F8teApHz7mmUkjW.Fa-xkplmxfo5EeXk?startTime=1683078771000>

VCS:

<https://github.com/carsongro/BookIt-08>

Contributions:

Carson Gross: I mainly worked on creating and styling the home page as well as the test cases for logging in and registering. I also created the initial database in the create.sql file. I also

contributed to the index page to create API calls to access data from the external book API. I also helped teammates debug and manage source control.

Gavin Zura: I worked on the search functionality and the book page for Bookit. The search feature allows the user to search for books based on their title and/or author and will list all the results for the user. The book page allows users to view a specific book, see its information, see all the reviews, and even add a review from the page.

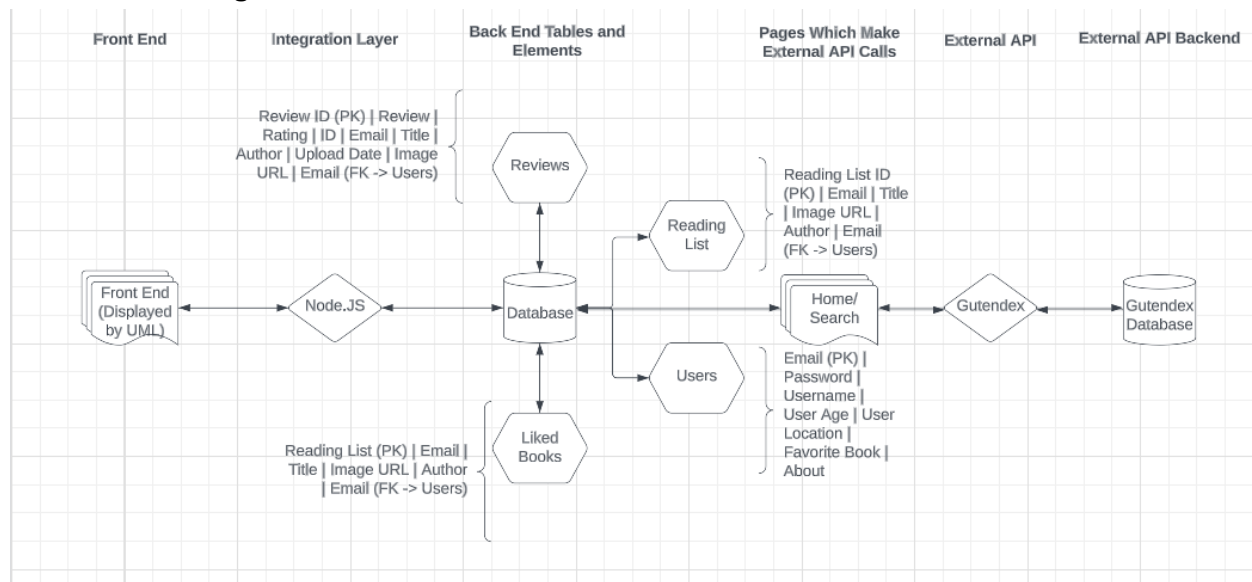
Brandon Warren: I implemented the reviews database, as well as the reviews page. I also implemented a feature that allows the user to search the reviews database using any combination of the following fields: the book, the book's author, the user who posted the review, and the book's review score. Further, I implemented the liked books feature, which allows the user to track their favorite books. I also implemented our register feature's test cases, as well as doing a lot of bug fixing and edge-case testing.

Justin Nguyen: I worked on adding a reviews page and got the functionality working for users to be able to add reviews on their favorite books. The user is able to search for books and write reviews on any book they'd like. The information is saved into the database where all users can view. Furthermore, I also helped in the beginning with the login/register api's and I worked on styling the website.

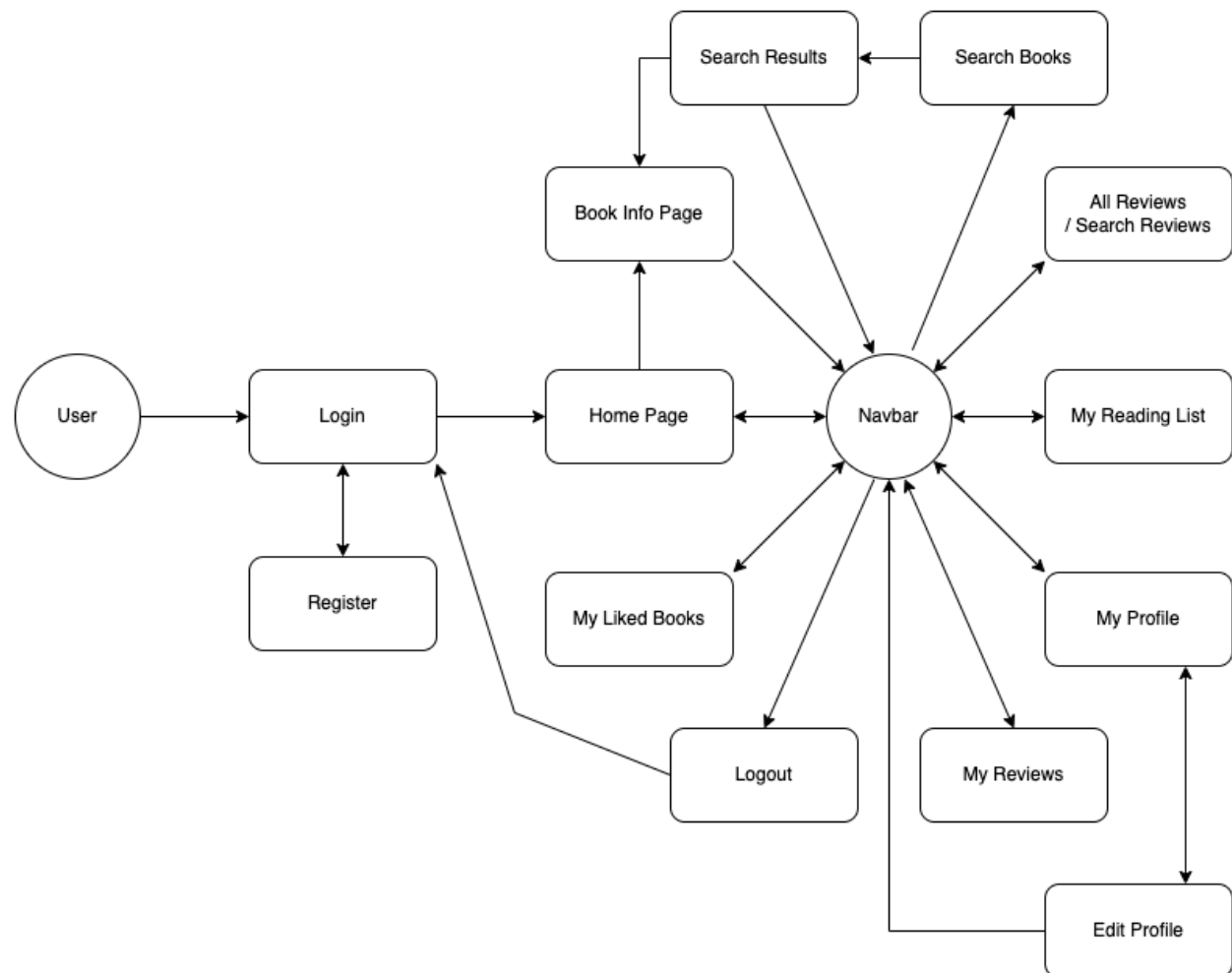
Dennis Pozhidaev: I mainly focused my work on the creation of users' profile page and the styling of that and other pages to fit the book theme we were going for. The profile page allows the user to make a custom username, set their location and age, write down their favorite book so far, and lastly write a little bit about themselves. The initial database was updated to fit in these fields and API calls to access elements needed to store and use the users' profile information were created in index.js. Along with this I helped with the login test cases when we first started the project to ensure that logging in actually works.

Zach Driver: I worked on the "my reviews" page and "reading list" page. On the review page, I added a system that sorted through all the reviews in the database and only showed reviews written by the user, as well as the functionality to delete a review. On the reading list page I created a database to store books that users were interested in, as well as an option to remove a book from your reading list. For the reading list I also added the ability to add a book to your reading list from the home page and the book page.

Architecture Diagram:



Use Case Diagram:



Test Results:

```
allprojectcode-web-1 | Server!
allprojectcode-web-1 | Database connection successful
allprojectcode-web-1 |   ✓ Returns the default welcome message
allprojectcode-web-1 | (node:54) [DEP0066] DeprecationWarning: OutgoingMessage.prototype._headers is deprecated
allprojectcode-web-1 | (Use `node --trace-deprecation ...` to show where the warning was created)
allprojectcode-web-1 |   ✓ positive : /register (108ms)
allprojectcode-web-1 |   ✓ negative : /register (65ms)
allprojectcode-web-1 |   ✓ positive : /login (1914ms)
allprojectcode-web-1 |   ✓ negative : /login (1634ms)
allprojectcode-web-1 |
allprojectcode-web-1 | 5 passing (4s)
allprojectcode-web-1 |
```

Deployment:

While the virtual machine is running, the site can be reached at:

<http://recitation-11-team-008.eastus.cloudapp.azure.com:3000>

The site is hosted through Microsoft Azure, following the instructions detailed in Lab 13.