

Login Page Test Plan

Unit testing:

- Each location users enter data can be unit tested to verify that we are only accepting response types we expect.
- This is important for testing any regex (like there might be for email entry)

Integration testing:

- Confirm that once logged in the user is directed to the home page.

User Acceptance Testing (the following all work):

- User can log in with valid credentials
- User cannot log in with invalid credentials
- Mandatory fields must be filled
- A cookie is correctly used to keep track of the session

Home Page Test Plan

Integration testing:

- If a book is added to to library confirm it is included in their profile library

User Acceptance Testing:

- A user can browse books from the home page
- They can add books they are interested in to "want to read"
- They can mark books they already read, as read.
- They can go to their profile
- They can edit their profile information like(name, age, where they live)

User Profile Test Plan

Unit Testing:

- Book form entry fields are checked to be valid

Integration testing:

- Confirm once a book is added via the add book form it is included in their library on their profile.

User Acceptance Testing:

- A user can view the books they have read on their user page.
- They can add books via the book form
- They can visit the home page.

Individual Contributions:

Helen Dupree: Created the book partial and linked it into the author/user page, this will be used in the library page also. Currently working on having it fetch data from the google books API based on values stored in the user table of the database. Also reworked the author page to be able to be used as a general user page if the general user page does not work out. And fixed the SQL commands to create the table. I also created the whole testing plan.

Last Commit:

<https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-02/commit/9bff93a8b7e421866e5983cade0d151912ded318>

Zach Trainor: Converting HTML files to EJS. Created header, footer, navbar template for easier use. Worked on Dockerfile and init SQL file. Created library page with CSS/JS. Currently still working on library page.

Last Commit:

<https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-02/commit/5e50bef5b3aec758366f1c5c33b9fe32c8ffca6b>

Delan Huang: I kinda haven't done anything for a while.

Sayat Toktarov: I haven't done a lot of things. I was mostly trying to learn how to convert HTML files to EJS and getting a better understanding in SQL. I feel like I'm a little behind my team in the knowledge so not to be confusing I want to catch up to my team.

Joseph Sembower: Haven't done as much as other team members. I have been more focused on the database sort of side of things. This includes SQL, PostgreSQL, and PHP stuff I am hoping to have done and implemented by Tuesday. This has all been done locally so I don't have a recent github push, still the same as the last one from Milestone 4.

Ben Burkhalter:

Implemented all HTML files into .ejs. Most importantly added javascript functions to dynamically populate each row of the home page based on a given criteria. Currently I have made two functions that populate one row each with 15 books using the google books API. The first uses a search query based on the author, the second based on genre. The home page now also runs on localhost.

Latest Commit:

<https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-02/commit/d17d9f3ea10fa4534edbc3da69c104b11330f932>