Overview

Our web application is a college book swap. Students from the five colleges (UMass Amherst, Hampshire, Amherst, Mount Holyoke, and Smith) can trade books with each other. After making an account, users can browse and/or search for specific books they might want and communicate with other users to set up a trade. Users also have the option to keep a wishlist of desired books and a list of owned books. As mentioned in our project 2 overview, we still have not implemented comments but that is something we will probably get done for the final submission.

Team Members

Nick Ferneza (Ferneza), Ohad Aharoni (Alohad), James Reedie (jreedie), Tim McNamara (timothymcnam), Nase Masi (natejm47), Eric Zhou (ejzhou)

Github Repository

https://github.com/timothymcnam/BookSwap

Design Overview

On the navbar of every page of our application in the top right corner is the option to login/logout. If a user is logged in a button on the navbar to access the my profile page will show up along with a logout button. If the user is not signed in, they will only see a login button in the upper right corner of the navbar. On the my profile page there are buttons to add titles to the user's wishlist, edit their bio, and to add to their list of books owned. The button to add to books owned redirects to the add book form page, while the edit bio and add titles to wishlist page redirect to their own forms as well. These forms allow for modification (bio and wishlist additions) of data as well as addition (adding books) of data to our database. Also, our home page and browse pages are now populated with book instances from every user in the database other than the one that is currently logged in. When a user inputs information on the add book page or add to wishlist pages, a book will be created if one does not already exist with the given information.

Problems/Successes

Unfortunately, our team made the fatal mistake of naming our user model 'User,' which is the same name as Django's built in user model. In order to merge the two models we had to rename ours, and this caused a lot of problems. We resorted to deleting our database to get things to work, which was annoying but not a huge deal. We also ran into problems with our database being overwritten due to multiple people attempting to add data, but we solved this by starting over again and having me (Nick) be the only one allowed to modify data to avoid conflicts. We were able to successfully divvy up work and get most of the assignment done outside of a group setting, but we did have a few problems with overwriting past work in github. In the future, I think most of us could learn how to use github more effectively to prevent these problems from happening.

Team Choice

For the team choice component we are going to figure out how to store images in the database for books and book instances along with how to present them in our web application. We are also considering implementing messaging. While working on the last project we tried implementing pictures with pillow but ran into some issues doing so, and Tim told us we could delve into figuring out how this works and have it be our team choice component.

Individual Write Up

Eric Zhou: Created editBio form. Added styling changes to the editBio.html, login.html, addWishlist.html, and addBook.html to enhance user experience. Pointed out some minor bugs. **14%**

Nick Ferneza: Added login logout functionality and edited the navbar to be dynamic based on whether a user is logged in or not, re-populated our database with several users/profiles and many books/bookinstances. Created form for adding books to a user's wishlist along with the html page and view functionality required. Fixed a couple minor bugs on the home and browse pages. Wrote the team write up. **18%**

Ohad Aharoni: I created dynamic functionality for the browse page which shows every available bookInstances excluding the currently logged in user's ones, where each bookInstance links to the profile page of the user which owns that bookInstance. Helped set up more database entries in order for us to test our website. Helped fix some bugs other members pointed out. **13%**

James Reedie: Fixed models to change our user model to profile (with assistance from the rest of the group) so that we could connect our users with Django's provided user model. Edited profileOther model to work dynamically with any users. Edited profileOther and profileSelf views and templates to work dynamically with the user being viewed and the logged in user, respectively. Fixed issues with index view and template. Gave advice on or handled various bugs. **20%**

Nathan Masi: Worked with the group to adjust our models to include Django's built in user model, changing our user to profile. Made changes to our index page to have it dynamically display 3 book instances from random users in the database. Helped with bugs and small visual changes. **15%**

Timothy McNamara: For this part of the assignment my main task was to set up a form where the user can add a book. For this I created the addBook form, edited the addBook.html template (which was later changed by Eric to look better), and added the url in urls.py to direct to my new addBook page. Then for the majority of my work, I wrote the addBook request in views.py which for a post request retrieves the cleaned data from the form, queries to see whether the entered author and book exist, and then will either use the existing author and book, or create a new author and/or book, then redirect the user back to their profile. Then, for a get request, it will display the appropriate form. Lastly, I edited the database to include all the book genres, helped fix a few bugs, and did some git back-tracking when our master branch broke. **20%**