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. Currently, we removed profile comments, profile pictures, and user reputations from project 2, but we plan to re-add them later.

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

Our data model has 5 classes - genre, book, bookinstance, author, and user. Genre has one field called name, where the user selects the genre from a dropdown menu. Book has fields title, author, summary, genre, for_class, and book_pic. Bookinstance has fields id, book, owner, instance_pic, book_condition, and comment. Author has fields first_name and last_name. User has fields id, first_name, last_name, university, bio, books_offered, and books_wanted. The URL routes that we implemented first included linking all of our html pages together where we indicated they would in the mock UI. Also, we had to include variables referencing things such as user.bio, book instance titles, etc. to our html files to represent mock data in our pages. ProfileOther has links to home, browse and profileSelf, profileSelf has links to home, browse, and itself, addBook, browse has links to home, itself, and profile other, addBook has links to home and browse, and Home(index) has links to itself, browse and profileSelf. The important UI views include a query over all book instance objects to fill the navigation arrows on the browse page, a view to filter through all book instances object under the user "Nate" for the profileOther page, and the same views used for the profileOther page along with a view to retrieve all of user Nate's wanted books for the profileSelf page.

Problems/Successes

In terms of successes, the best thing we did on this project was the decision to work on parts 1 and 2 together instead of divvying up work. Project 1 did not require as much coordination when it came to coding html/css pages as this project did. Eric proposed that we worked in the Elm classrooms since they have projectors and that worked well. Nate was our 'leader' during these two sessions, as his computer was hooked up to the projector, and we all worked through the coding together. Since we are a 6 person group, the maximum size, coordinating times to meet is harder than if we were a 4 or 5 man group. However, we still were able to get 5 people to meet together on Sunday and all 6 to work on Wednesday, so we were able to work around that.

In terms of implementation difficulties, we spent a good amount of time trying to figure out how to set up our books_offered and books_wanted models, and we finally got an answer for that from Tim. We also initially had a lot of trouble getting our css and images to load on our pages because we forgot to reference them statically. In addition, figuring out how to access our pictures in our database through view functions was problematic. One thing that we can definitely do to improve is further our understanding of git and take advantage of branching. This would allow us to get more work done at once and not have to worry about overwriting others' work. Also, we had a problem getting images to work properly with imageFields, but Tim advised us at his office hours that using static images was satisfactory for this submission. Lastly, it seems as though we underestimated the work that had to be done in part 2 of the project.

Individual Writeups

Eric Zhou: I shamelessly hoarded an entire classroom for our group to use and stole HDMI cords from nearby rooms for the projector. I helped think about the contents of our data model. I helped with debugging some issues (referencing static components, mapping the url, etc.). Reviewing certain sections of the mozilla tutorial proved to be helpful. I managed to use a for loop to print out a book's genre(s) - a many to many field. I helped Ohad fix his carousel. **Percentage of work: 16%**

Nick Ferneza: I was active in slack throughout our time working on this project and was present for the entirety of our two group working sessions. I helped organize the team and figure out when we could meet. I commented whenever I could on trying to debug issues or recalling information from the mozilla tutorials, and was a part of figuring out the initial data model. I set up our templates folder with everything from project1 and added the urls to urls.py. When we were having trouble getting information from the "Nate" user to show on our page I suggested running a for loop through our 'list' of only 1 element and that solved our problem for the time being. I also wrote the majority of the team write up and am responsible for turning it into a pdf when everyone finishes their write ups. I hope to be more involved in the coding aspect of the project for part 3, but I feel as though I am in an awkward spot right now because every other member has a 'page' that they initially created in project 1 that they are focusing on and I do not. This will change though, since we need to add more pages to the application and I plan on volunteering to do a lot of work on these. **Percentage of work: 16%**

Ohad Aharoni: I was available for assistance at all points of the group project and was present for our two meetings. During these meetings I assisted in creating and debugging the models page, including fixing an issue we had related to ForeignKeys and ManyToManyFields in our user model. I went to multiple TA office hours in order to get assistance and fix issues we couldn't figure out while all working together. I built the functionality of the browse page (along with some debugging assistance from Eric). At one point Jamie was having an issue where his buttons wouldn't link properly to other pages so I helped him fix that. **Percentage of work: 16%**

Tim McNamara: I started off the project by setting up the skeleton of the website. Then, the majority of the project we did together as a group, so I (along with everyone else) was present at 2 meetings to collectively figure out and modify the data model, urls.py, models.py, views.py, build the base_generic.html, and extending the base_generic.html in index.html. Then on my own, in the addBook page, I extended the base_generic.html along with fixing links and images to correctly represent their new static and dynamic locations. In addition to doing these I built the data model in Draw. Lastly, I added a very bare contact us page. **Percentage of work: 16%**

James Reedie: The majority of the work was done collaboratively. Although I was unable to attend the first meeting where we drew up our data models, I was still able to help with debugging and editing the design of them. Aside from helping people figure out what was wrong and how to fix it when they were having issues with their pages, individually I did the profileSelf template and view. In general I did my best to make sure everyone stayed on task and set goals to make sure we could finish on time. **Percentage of work: 16%**

Nathan Masi: Once Timmy set up the Django skeleton website, I worked along with my group mates in setting up our UML diagram and discussing which data types we would need in implementing our web application. Once we had decided on that, we met up in a classroom, populated some test data, and started to move our templates into the actual Django environment. Our main goal after our first meet up was to get all of our test data squared away, and to make sure our mock html files were the format we wanted.

We then met up a second time, and again I plugged into the HDMI (this is why most commits are from my machine for a while) and we started to get our urls working. Once we had all of the links we wanted, we started to replace our hard-coded data with the test data by making queries in our views file. After we got the hang of replacing our mock data with the test data, we split off and started to just make the necessary changes to each of the web pages that we had created for the mock UI. I worked on the otherProfile page (the page you see when looking at another user's profile). I would say I definitely contributed my fair share of work to this project. **Percentage of work: 20%**