Project Reflection Report

By Jaywoo Jo, Ruhana Azam, John Shen, Alan Zhang

---

1. Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).
    - The direction of the project stayed almost entirely the same. The only change would be that we added a "Best Books/Authors" page that utilized our advanced queries to show the top books and top authors.
    - Our UI eventually was also slightly different from the UI mockup we gave in Stage 1, but that was expected as that was a rough sketch.

2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.
    - Our application achieved the main goals for functionality as detailed in our stage 1 proposal.
        - Users of this application are able to create/log into their own accounts using their @illinois.edu emails and modify their basic profiles (name, gender, introduction, etc.)
        - Users can search for books in our database, get all relevant information about the book (genre, author, awards, publisher, publishing date, etc.), and add them to their "My Likes" list
        - On top of searching for a certain book, they are also provided with contact information from other users with common interests and genres in the "Connections" page
    - An extra component we could have added if we had more time was allowing users to make comments on their favorite books and leave their own reviews. This could have added more usefulness to the application.

3. Discuss if you changed the schema or source of the data for your application.
    - We did not change the source of our data. All utilized data are from "Best Books Ever Dataset" as mentioned in the proposal. We used a random user generator for users when doing testing in early stages.

4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?
    - We only made minor adjustments in not utilizing all the attributes we had in our ER diagram. In our database design for Stage 2, we had a "description" attribute in Award, "averageRating" description in Author, and "description" in Genre. Despite being in our database, these attributes were not included in our final product as we felt it did not provide much usefulness for the way we developed our application.

We put most of our efforts towards making better functionalities: (1) being able to get book information for a searched book and (2) being able to see what other UIUC students share the same book interests as you. The few excluded attributes did not jeopardize the utilities of our application.

- To implement the stored procedure, a couple temporary tables were employed to keep track of 1) each book a user likes that is common to that of another user 2) each genre a user likes that is common to that of another user. These were created as an intermediate step to calculate the user recommendations. We did not initially include this in the ER-diagram because it came about in a later stage and at the time we did not know it would be helpful when calculating user recommendations.

5. Discuss what functionalities you added or removed. Why?

    - As each stage came along and more requirements came about, we moved things around from how they were described in our initial project description. For example, the "My Likes" page was moved from a section on a user's profile page to a separate page on our navigation bar.
    - We added a best books/authors page to visualize the books and authors with most awards and highest rankings. While this was initially created to satisfy the requirement of implementing our advanced queries, we found that the feature aligned with our vision of the final application as users would naturally be interested in the best books and authors of our database.
    - In general, the functionality of our application remained largely unchanged from how we planned it.

6. Explain how you think your advanced database programs complement your application.

    - **Stored procedures** were pivotal to our application because they connected users with each other. Connecting book lovers with each other is the main purpose of our application. Each time a user opens the recommendations page, a list of illini students are recommended based on their book likes using a stored procedure.
    - **Triggers** are important to filter input that complicate our systems. In our application, we used triggers to prevent creating a duplicate account or accounts that are not from UIUC (not ending with @illinois.edu). If errors occur due to one of the two situations, we will catch the error, give users feedback, and make users attempt another username.

7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.

- **Jay:** Implementing a log-in/sign-up page can be more complicated than you may think, but looking up resources and reading documents proved to be very helpful. One issue I faced was learning to create a custom middleware so that when a user logged in, the current user would be saved as the current session and then the current user could be accessed across all EJS templates. Another advice would be to not fear looking up issues you face and reading documentation and stackoverflow. I handled most of the front-end and faced an abundance of issues along the way that I was able to resolve or find different solutions by looking up and learning.
- **Alan:** One issue we faced was the challenge of all team members working on the VM at the same time. We were unsure if it was even possible and if doing so would overwrite each other's work. We solved this by researching how to connect to our GCP SQL server from a local environment and learning how to use GitHub for version control. This allowed us to each write code in separate branches and use pull requests to merge them.
- **Ruhana:** Creating the stored procedure was more time consuming than expected because it is difficult to know which lines of the stored procedure are causing an issue. The best solution was to comment out parts of the stored procedure. Then return and verify intermediate queries being made. But, even this was time consuming.
- **John:** Creating triggers for account sign-up can be a bit tricky because front-end and back-end need to work together for feedback during signup. In our case, when we find there is a duplicate account or the email is not from Illinois, the trigger returns an error and the server catches it and a warning box on the interface appears.

8. Are there other things that changed comparing the final application with the original proposal?
    - There are no significant changes. Our end product ended up matching what we stated in our original projection proposal description and usefulness.

9. Describe future work that you think, other than the interface, that the application can improve on
    - We can add security functionality to prevent multiple attempts from the same user with the wrong password.
    - We could implement something to allow users to upload an image of themselves in their profile to be viewed by others.
    - Currently, we need to input the exact book name and author in two separate bars. We can improve the search bar by implementing autocomplete predictions. In other words, making it easier to search such that as soon as you start typing a few letters, the search bar will start trying to guess what you're searching for and you have options to click on so you don't have to type out the whole thing.

- Some of our pages run a query or a stored procedure on page load. This is a little inefficient as if no changes are made to our databases, the results of these queries will stay exactly the same. One improvement we could make would be to store commonly accessed data in a cache to display on the frontend. We would then only need to run the query whenever we think it will return new data. This would make our application much more efficient and decrease the loading times of these specific pages.
- A cherry on top can be a "comments" section for each book, where users can look at how other Illini think of this book.

10. Describe the final division of labor and how well you managed teamwork.

As our original plan laid out, Alan and Jay focused on the front end, while Ruhana and John focused on the backend. The front end consisted of working on each page: home, login/signup, profile, book search, my likes, connections, best books/authors. Additionally, Alan and Jay were responsible for making sure users could login and they stayed logged in globally.

The backend work included writing queries and populating the database. Queries included: inserting, deleting likes, updating user info, creating new users, and searching books. Back end work also included created triggers and the stored procedure. Additionally, Ruhana and John were responsible for parsing, cleaning the original csv file.

Throughout the project, we maintained good communication and were responsive to each other members' messages. We frequently called to stay on the same page and to discuss how we would work on each stage of the project. Each member did the assigned work which was agreed upon during the meeting.