indiereads:

A Goodreads Alternative

Group 1 - Joseph Akers, Dillon Prescott, Jamie Winter

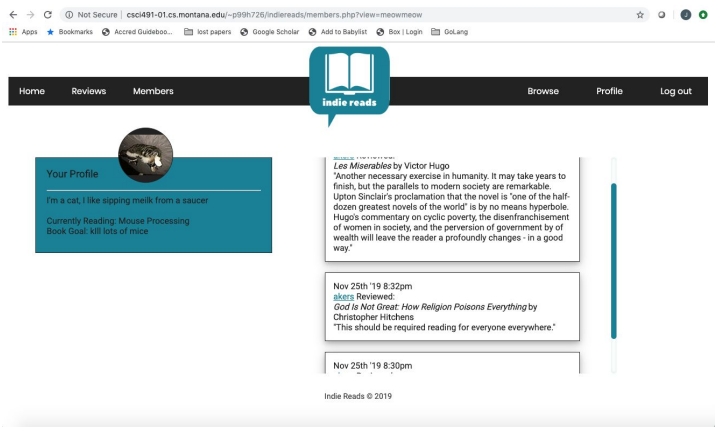
http://csci491-01.cs.montana.edu/~p99h726/indiereads

Objective:

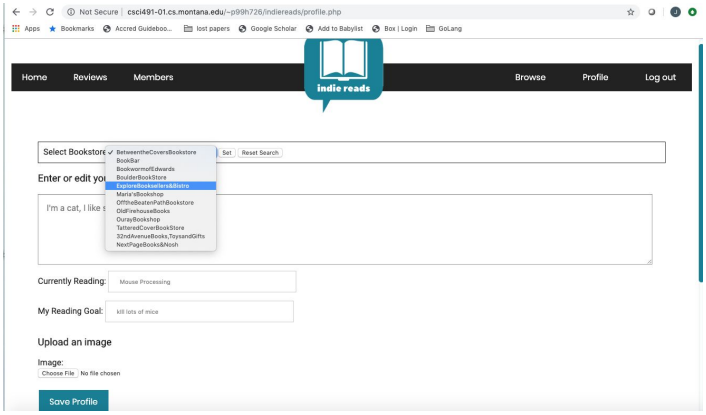
For those in the book industry, consumers and producers alike, Goodreads is an invaluable resource. It provides a platform for reviewing titles, discovering new titles, making connections with like minded individuals, and keeping track of what titles you have read and when. While incredibly useful, supporters of independent bookstores are torn between using Amazon owned Goodreads, benefiting indies greatest competitor, and losing out on the only book centric platform out there. Indiereads seeks to become the alternative to Goodreads with a focus on connecting with local independent bookstores while maintaining the features users love.

Summary:

Creating a social media platform is a large task. That being said, we chose to begin our project by using the LAMP stack assignment as our base. The LAMP stack utilizes the software stack of Apache, Mysql, PHP, Perl, and Python to create a website that had the functionality of creating accounts, log in/log out pages, connecting with other members, and writing messages already crafted for us.



To focus and emphasize local, independent bookstores, a search feature was added to the user's profile page. This allows users to find and select "their" bookstore. To find a store, a city and state is entered. A drop down menu is then returned and populated with all the local registered stores. After their location is set, a 'browse' tab in the navigation bar links them to the webpage of their given store. The name of the store as well as a link is also displayed on their homepage. This functionality was primarily accomplished via Javascript.



We needed to expand the database from the original LAMP stack assignment. New variables were added to existing tables and a completely new table was created to hold the information on the 440 independent bookstores. Creation and implementation of the bookstore table was a challenge as the original dataset was in a less than ideal format.

To achieve the look and feel of a modern social-media web application, the group decided it was essential to provide a "news-feed" feature. The primary functionality we wanted to demonstrate was the scrolling and message auto-loading commonly encountered on sites such as Facebook and LinkedIn. This took some tricky Javascript event-listening as well as JS ajax 'get' and 'post' calls.

To emphasize and encourage the book-centric nature of the application, we wanted to shift users away from the traditional personal message and instead focus on writing reviews of books they are reading or have read. Additionally, significant changes were also made to where, and which, certain reviews are shown. For

example, when a user opens their home page, only the reviews of their friends are shown. However, if they view another user's profile, only the reviews written by that user are displayed. Additionally, a user cannot directly message or write on someone else's page. This was done intentionally to emphasize the "review" nature of posts. At this point, reviews only allow text and fields for an author and title but follow on functionality would include a numeric rating system and well as provide the ability for users to chain, or add comments/ reviews to existing posts.

Members:

Joseph Akers
Github: https://github.com/jjg-akers
Email: jjg.akers@gmail.com

Joseph's primary responsibilities were the backend updates to the database and the corresponding PHP queries. This involved modifying many of the existing files as well as writing new PHP files. Additionally, Joseph was responsible for all of the Javascript necessary for the implementation of the news feed as well as the connection to and functionality behind the bookstore search and profile page updates.

Dillon Prescott
Github: https://github.com/dill01
Email: dillon.prescott@gmail.com

Dillon worked on getting the sheet of 440 bookstores into a readable format that could either be imported in to a mysql database via the connect engine that he found for MariaDB or be read as just a json file to be searched for a users Set store. He ran into some trouble with the connect engine when it came time to upload the database to the server as not all the required permissions had been assigned to the users to use the connect engine fully. This led to him getting the server Admin to install phpMyAdmin so that he could just import the database that he had created locally using the connect engine and the JSON file.

Jamie Winter
Github: https://github.com/jailarie
Email: jailarie.winter@gmail.com

Jamie contributed all of the design work for the project, from creating the wireframes using Balsamiq, the logo using Canva, and all CSS. The main hurdle she encountered was integrating CSS with PHP. Most problems ended up being solved by being extra specific when referencing classes and ids. She also headed the task management in Asana as well as the formatting of the PowerPoint presentation and portfolio.

Conclusion:

In the end, the group accomplished what we sought out to do in the scope of this project, but if we were to continue a different approach would be taken. While using the LAMP stack assignment as a starting point was convenient, we later realized that reworking the code to suit the specific needs of a fairly different project was difficult and messy. It was a good experience however, and a good model for what will likely be encountered in the workplace.

If we were to move forward with indiereads we would start from scratch, writing all of our own code. This would allow the group to organize and create everything precisely the way we needed it. We would also move away from LAMP and use Node.js for the backend and React.js for the frontend.

References:

AMPPS - http://ampps.com/

Balsamiq - https://balsamiq.com/

Canva - https://www.canva.com/

Goodreads - https://www.goodreads.com/