Book Recommendation Site with a Marketplace Implementation

James Karoly[1], Dhanasri Salla[2], Sai Sandeep Mandava[3], Praveen Raj Gujju[4],

Anusha Meka[5], Sairam Vinjamoori[6], Krushal Rama[7], Alekhya Bonnineni[8]

Seidenberg School of Computer Science and Information Systems

Pace University, New York, NY, USA

[1]jk52340n@pace.edu [2]ds28271n@pace.edu [3]sm22201n@pace.edu [4]pg41335n@pace.edu

[5]am67344n@pace.edu [6]sv16921n@pace.edu [7]kr97409n@pace.edu [8]ab97535@pace.edu

Abstract— **Recommendation engines power everything from social media and news sites to e-commerce sites. They are incredibly important to help users find things they might want to purchase or read. There is an inherent problem to these engines though in that they do not give users a diverse set of recommendations. They mainly settle on sending the most popular items that are more similar to what a user likes. We want to create an algorithm that takes diversity into account and give the users a more varied set of recommendations. This should help underrepresented and newer authors find an audience when regular recommendation engines might deprioritize these results.**

Keywords— **Machine Learning, PyTorch, Recommendation Engine**

I. Introduction

Today, people have many different streams of media and content to sift through it is hard to find good recommendations for new products or content. This is especially prevalent in the literature industry where thousands of books are printed yearly. This is where recommendation engines are put into play. Recommendation engines power everything from social media sites to news sites. They use a machine learning algorithm to take the user's likes and dislikes to recommend things to them and they are an integral part of the internet, and most websites use them in some form.

Since these engines play such a crucial role in the user's ability to navigate all the options of content that are out there. Our system will harness a custom machine learning algorithm and a modern UI to cater to both experienced and new readers alike. The algorithm will promote diverse recommendations to help the user discover books they would not have otherwise.

In addition to the recommendation engine, the site will also include a way for users to organize their reading habits into custom collections that they have total control over. It will also include a full marketplace implementation so our users will be able to purchase any of the books recommended to them. We want the site to be a one stop place for book lovers to organize their reading habits and find new books to read.

II. Literature Review

Recommendation engines for books are nothing new. They are utilized by e-commerce sites and custom sites specifically made for book recommendations. One of the more popular one is called GoodReads. GoodReads' recommendation engine is incredibly popular and research has even been conducted on it. A research team in 2017 even discovered that by analysing the GoodReads data they could predict what books would be on the Amazon best sellers list because they saw a spike in readers on GoodReads [1]. This shows that these sites along with their engines can accurately track reading habits and their dataset is highly detailed.

This informed our research and helped us choose a dataset. There have also been studies done based off GoodReads that tests different machine learning algorithms to find the best option [3]. Though these engines mainly focus on serving reader the most popular recommendations and don't really worry about the diversity of them.

We also researched recommender models in general, we found a great survey on current advancements in recommendation engines and used that to inform our ML algorithm [2].

III. Project Requirements

Our goal is to develop a web application that will give book lovers a place to find new recommendations. We want the site to be able to be run on all modern web browsers in use and should have an intuitive and easy to use design.

A. *Functional Requirements*

1) User Authentication: The system should allow users to create an account and log in with their credentials. The app should also store user data securely.
2) Generate Recommendations: The system should generate recommendations using a custom machine learning algorithm.
3) Recommendations Display: The system should display the recommendations to the user.
4) User Collections: The system should allow the user to create custom collections for their books.
5) Marketplace: The system should have a marketplace implementation to allow our users to purchase books

IV. SYSTEM DESIGN

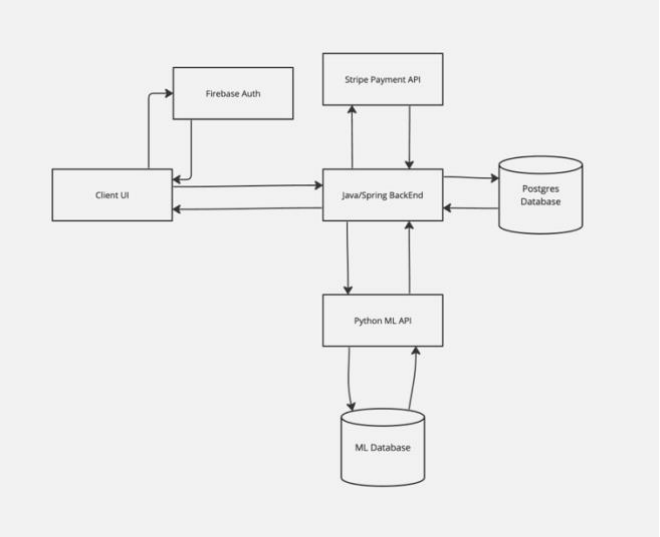

Fig 1. Conceptual Architecture Diagram

A. *Architecture*

Our system will have a decoupled front and back end. We will write a custom API to handle all backend tasks and to communicate with the front end. This architecture was chosen to make future maintenance and the addition of features easier. API driven design is also very popular in a lot of companies' tech stacks, so it was a way for our team to become more familiar with this architecture. The client UI communicates with two main components: Firebase Auth and our Java Backend API.

B. *Frontend*

The frontend of our website is written with ReactJS, some CSS styling and some HTML when needed. We chose this tech stack because it is highly compatible with all modern browsers and allows us to create an easy-to-use UI. We chose to use Firebase Auth as a way to manage passwords. This allows us to focus on the implementation of our site instead of focusing on things like password hashing and secure storage.

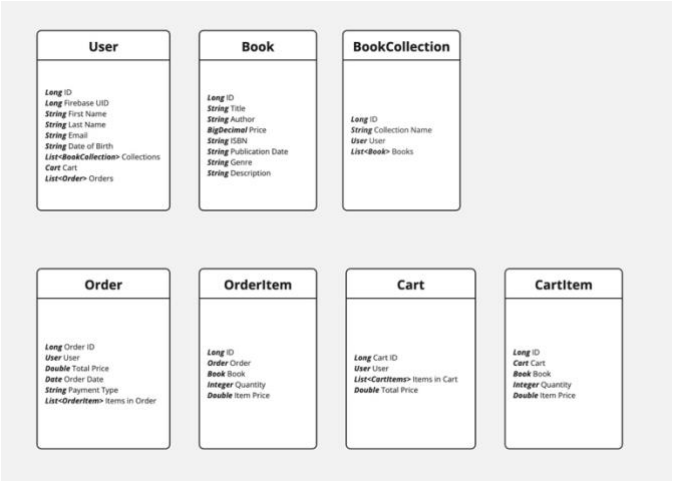

Fig 2. Class Diagram

C. *Backend*

Our backend is written in Java using the Spring Framework and uses Postgres as our database. We chose this based on our team's familiarity with the language, and Java is still widely used today in many companies. Our backend will communicate with two APIs, a payment API by the company Stripe and then the ML API written by our machine learning team. The backend then communicates with a Postgres database for storage.

As seen in our class diagram, we have 3 main entities for the book recommendation side: User, Book Collection, and Book. This lets us model the appropriate relations and allow our users to create their own collections with our custom-built logic. The marketplace contains 4 main entities: Cart, CartItem, Order, and OrderItem. These allow us to persist a cart between sessions and then convert that cart into an order when the user requests it.

D. *Backend API Details*

Our backend API follows the REST architecture. It handles 4 main areas: User Management, Book Collection Management, Book Management, and Marketplace Management. It follows the main CRUD operations that are defined in the REST architecture. For most of the request we return the entire object to the frontend, such as the entire Book object, but with the user we created a User Data Transfer Object as well, this allows us to just choose specific features we want to transmit to the frontend instead of sending the entire object. This will mainly be used to populate user details since it omits the database id and the firebase id, two things that will never get displayed to the user. This allows us to maintain the security of this data.

E. *Machine Learning API*

The project also includes a separate machine learning API created with Python. It is relatively simple with just two endpoints. The main endpoint will be used to generate and then return recommendations to the client. We also implemented a search endpoint to allow search access to our dataset.

Creating a separate API made sense for our project because it allowed the machine learning team to write in a language, they were more familiar with. It also allows for us to plan for the future scope of the project. In the beginning the algorithm will still be training from every user interaction, as the algorithm gets better, we could possibly sell access to just the ML API, and having it already designed as a separate system makes that easier.

V. MACHINE LEARNING

In book recommendation systems, cosine similarity is used to calculate the similarity between different books by treating them as vectors in a high-dimensional space. Each dimension represents a feature such as genre, author, or keywords

extracted from the text. By computing the cosine of the angle between these vectors, the system identifies books with similar content or themes, suggesting them to users with matching preferences. This approach enhances personalized recommendations by focusing on content similarity.

VI. Conclusion

Book Buddy is a web application that will allow users to organize their reading habits and generate recommendations for new books they want to read. It will implement a custom machine learning recommendation engine and a marketplace implementation. The initial offering of the application will include the ability to create and edit custom book collections as well as offering user registration.

After delivering the MVP for this project, we could continue implementing new features. The ability to share recommendations and custom lists with other users of the site and non-users would be the first new feature implemented. This could allow our users to recommend the site to their friends and allow them to show off some of the features of our site. We would also continue training and working on the machine learning model. We can gain more information from our users and incorporate that. The algorithm would just keep improving and allow us to refine the model and diversity metrics.

References

[1] S. K. Maity, A. Panigrahi and A. Mukherjee, "Book Reading Behavior on Goodreads Can Predict the Amazon Best Sellers," *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Sydney, NSW, Australia, 2017, pp. 451-454.

[2] Y. Li, K. Liu, R. Satapathy, S. Wang and E. Cambria, "Recent Developments in Recommender Systems: A Survey [Review Article]," in *IEEE Computational Intelligence Magazine*, vol. 19, no. 2, pp. 78-95, May 2024, doi: 10.1109/MCI.2024.3363984

[3] R. A. Wijaya, S. Staniswinata, M. Clarin, N. N. Qomariyah and I. B. K. Manuaba, "Prediction Model of Book Popularity from Goodreads “To Read” and “Worst” Books," *2023 10th International Conference on ICT for Smart Society (ICISS)*, Bandung, Indonesia, 2023, pp. 1-7, doi: 10.1109/ICISS59129.2023.10291619.