PickaBook

Team 22 - Product Backlog

Logesh Roshan, Myeongsu Kim, Shobhit Makhija, Piyush Juneja

Problem Statement

The current environment around media consumption is primarily focused on audio visual systems in the form of games, TV shows and movies. This culture has lead towards a drop in literature consumption, general reading and in turn, the vocabulary skills among the general population. To combat this, our project aims to deliver a platform for exploring new and famous literature and books for readers with diverse interests to dive into, reviewing and discussing their favorite books, and creating a personalized recommendation system based on each individual's reviews. Our project will also provide the latest additions to the market to the users with the help of a crawler to update new book releases, all within an intuitive web framework.

Background Info:

Audience

The web has been used to provide new platforms for individuals to form communities and discuss their interests. Following this pursuit, our primary audience would be a community that consists of individuals, young or old, who fall anywhere within the range of a newbie book reader to literature enthusiasts.

Similar Platforms

There are multiple review based applications like PickaBook, examples being Amazon, IMDB and several others. Amazon primarily focuses on selling products, so their platform isn't geared towards discussing and reviewing books. IMDB, on the other hand, is a close to identical product, with the only difference being that their product focuses on movies and shows instead of books.

Limitations

A lot of the aforementioned platforms perform their services extremely well, but their focus is either toward making a product attractive and selling said products, or they do not deal with books at all. Furthermore, since these platforms provide a wide range of services, their UI is often cluttered; we aim to provide a simpler, more straightforward interface that will be easy to pick up.

Functional Requirements:

1. As a user, I would like to be able to Register and Create a new account.
2. As a user, I would like to be able to link my Google or Facebook account.
3. As a user, I would like to be able to reset my password.
4. As a user, I would like to be able to set my profile's privacy to control if other users can see my profile.
5. As a user, I would like to be able to view any book in the database.
6. As a user, I would like to view all my reviews.
7. As a user, I would like to be able to add books to my favorites collection.
8. As a user, I would like to specify my preferred genre.
9. As a user, I would like to be able to keep my profile private.
10. As a user, I would like to view other users' profiles.
11. As a user, I would like to be able to write reviews for books.
12. As a user, I would like to give ratings for books.
13. As a user, I would like to be able to view the collected works of authors.
14. As a user, I would like to be able to follow the author for any new publications.
15. As a user, I would like to view new published books.
16. As a user, I would like to view top rated books by year and genre.
17. As a user, I would like to be able to ask questions regarding particular books.
18. As a user, I would like to answer any unanswered question.
19. As a platform, I would like to learn different users' preferences.
20. As a platform, I would like to recommend books to users based on preferences.
21. As a platform, I would like to web crawl for new books to add *(if time allows)*
22. As a platform, I would like to provide users the ability to provide feedback on the website.

Non-Functional Requirements:

Architecture

The application will be developed as a decoupled unit consisting of a frontend and backend. This will allow us to easily focus resources on particular functionalities within sprints either in the front or backend. Decoupling will also resolve compatibility and communication issues between the backend and the frontend.

The backend will be developed with a RESTful API to allow easy frontend integration. The backend will be developed with Django, a powerful python web framework. Django has a wide variety of packages for web development and is highly scalable. It also comes with a well established Object Relational Mapping library for interaction with relational databases.

The front end will be developed using the ReactJS framework. React will be able to obtain data from Django by using the REST API, i.e. sending requests to Django's rest framework. Furthermore, decoupling the frontend from Django's complete framework and following REST API, allows us to develop in other platforms by simply deploying a frontend for the new platform.

Performance

Based on our hosting server, we expect to support about 500,000 book records with roughly 10,000 users. Furthermore, we guarantee a response time of at most 500 ms for any get request.

Security

Security will be a critical component of the *PickaBook* platform, as it will store sensitive user information - email addresses, telephone numbers, home addresses, et cetera. The Django framework already comes with built-in security features to prevent cross-site-scripting, SQL injection, and such. Within our database schema, we will implement roles and a permissions framework to ensure regulation of the information accessible to users.

Furthermore, we will allow users to set their profiles' privacy settings, to control whether other users can see their personal information or recommendations. All requests to the API will need to be authenticated in order to prevent abuse or damage to the program.

Usability

The user interface should be navigable, and the average user must be able to understand the design and layout fairly easily. A simplistic design for the UI would be crucial, as a poorly designed UI could hinder a fulfilling user experience. While platforms such as

Amazon and Barnes & Noble provide certain features similar to ours, our platform will differ insofar that our focal point is not product-selling.

Instead, emphasis will be placed on creating a recommender system that is accurate and thus, enhances the user experience by providing recommendations tailored to each user's needs. In that regard, our application will also double as an archive of books and literature available in the market (up to 500,000), with frequent updates.

Hosting/Deployment

Although the front and back ends will be disjoint for the most part, and only the backend (including the database and recommender system) will absolutely need to be hosted on the server, we plan to host/deploy both of them on an AWS server nonetheless.