**Book Recommendation System**

**A PROJECT REPORT**

**for**

**Mini Project (KCA353)**

**Session (2023-24)**

**Submitted by**

**Radhika Gupta**

**2200290140119**

**Shafali Gupta**

**2200290140139**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Mr. Prashant Aggarwal**

### Associate Professor



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(December 2023)**

**DECLARATION**

I hereby declare that the work presented in report entitled “Book Recommendation System” was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University of Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, that are not my original contribution. I have used quotation marks to identify verbatim sentences and give credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

**Name:** Radhika Gupta

**Roll No.:** 2200290140119

**Name:** Shafali Gupta

**Roll No.:** 2200290140139

**(Candidate Signature)**

**CERTIFICATE**

Certified that Radhika Gupta (2200290140119) has carried out the research work presented in this thesis entitled “Book Recommendation System” for the award of Master of Computer Application from Dr. APJ Abdul Kalam Technical University, Lucknow under my/our (print only that is applicable) supervision. The thesis embodies results of original work, and studies are carried out by the student himself/herself (print only that is applicable) and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Prashant Aggarwal Signature

Associate Professor

Date :

**ABSTRACT**

The world of book recommendations has grown a lot recently. Smart systems use different tricks to suggest books you might like. Some systems look at what others similar to you enjoyed (collaborative filtering). Others check what you liked before and suggest similar stuff (content-based filtering). There are cool math tricks too, like breaking down your book tastes into secret ingredients (matrix factorization). The new superhero in town is deep learning, which uses super-smart computer networks to understand your bookish desires.

To check if these systems are doing a good job, they have special tests. It's like when you have a new recipe and you check if it tastes yummy - but for book suggestions! There are challenges, though, like when you're new (cold start) or there's not much info on what books people like (data sparsity).

Researchers are also making sure these systems play fair and are super private. They're making them more personal and explaining why they suggest a book, like a friend saying, "Hey, you might like this because...". These systems are getting even better at suggesting books from different areas too.

In real life, these systems are working well and making people happy. As they keep growing, they're becoming more tailored and smarter, making our book adventures even more exciting!

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, Mr. Prashant Aggarwal (Associate Professor) and Mr. Prashant Agrawal (Associate Professor) for their guidance, help and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions have guided me a lot in completing this project successfully.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Radhika Gupta (2200290140119)

Shafali Gupta (2200290140139)

|  |  |
| --- | --- |
|  |  |
| **TABLE OF CONTENTS**  Declaration | **Page No.**  2 |
| Certificate | 3 |
| Abstract | 4 |
| Acknowledgements | 5 |
| Table of contents | 6 |
| Chapter 1: Introduction | 7 |
| 1.1 Overview | 7 |
| 1.2 Data Collection and Data Preparation | 8 |
| 1.3 Algorithm Used | 9 |
| Chapter 2: Literature Review | 10 |
| Chapter 3: System requirement and specification | 11 |
| 3.1 System Requirement Specification | 11 |
| 3.2 Hardware specification | 11 |
| 3.3 Software specification | 12 |
| 3.4 Functional Requirement | 12 |
| 3.5 Non- Functional Requirement | 12 |
| 3.6 Performance Requirement | 13 |
| Chapter 4: Related Work | 14 |
| 4.1 Recommendation System | 15 |
| Chapter 5: Result | 16-19 |
| Conclusion | 20-21 |
| Future Scope | 22 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to Buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy, or anything else depending on industries).

Recommendation systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. The main objective is to create a book recommendation system for users.

Since, here we are trying to recommend books to users based on their past purchases or ratings the user gave previously, we are basically trying different models like Popularity based recommender system, Collaborative Filtering based recommender system (user-item or item-item) etc. We will be using the Popularity based recommender system to deal with the cold start problem, where we do not have history of past purchases of a particular user or where the user is totally new.

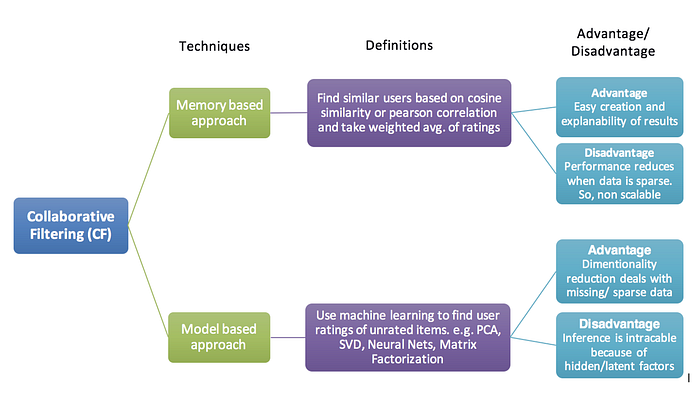


Fig 1.1 Collaborative Filtering Model (CF)

* 1. **DATA COLLECTION AND DATA PREPARATION**

**Data Summary:** We are using Book-Crossing dataset to train and test our recommendation system. [Book-Crossings](http://www2.informatik.uni-freiburg.de/~cziegler/BX/) is a book ratings dataset compiled by Cai-Nicolas Ziegler. It contains 1.1 million ratings of 270,000 books by 90,000 users. The ratings are on a scale from 1 to 10. The Book-Crossing dataset comprises 3 files.

**Users:** This .csv file contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.

**Books:** Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavours (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon website.

**Ratings:** Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1–10 (higher values denoting higher appreciation), or implicit, expressed by 0.

**Data Collection**: Before building any machine learning model, it is vital to understand what the data is, and what are we trying to achieve. Data exploration reveals the hidden trends and insights and data preprocessing makes the data ready for use by ML algorithms.

* 1. **ALGORITHM USED**

For the scope of our project, we used the K-Nearest Neighbours algorithm. KNN is a machine learning algorithm to find clusters of similar users based on common book ratings, and make predictions using the average rating of top-k nearest neighbours. We convert our table to a 2D matrix, and fill the missing values with zeros (since we will calculate distances between rating vectors). We then transform the values(ratings) of the matrix data frame into a SciPy sparse matrix for more efficient calculations.

**CHAPTER – 2**

**LITERATURE REVIEW**

[1] A common and cold e-commerce problem is the suggestion framework. Recommendation system per fin multiple ways including faculty members based on quality, suggestions for reciprocal filtering, and sung the combination techniques. This article Proposes a collective suggestion filtering system focused on the naïve Bayesian approach. The current recommendation method does have a great performance, according to both the undertaken experimentation, then numerous prior implementations, including the praised k-NN Algorithm being used by suggestion, especially at a longer length.

[2] There is a downside to the existing suggestion process Great performance, according to both the undertaking Experimentation, then numerous prior implementations, including the praised k-NN Algorithm being used by suggestion, especially at a longer length System of suggestion progressively is used in fields, such as films, traveling, songs, books, etc. Increasing social acts have amplified the user commending programs in persons and community Recommending programs. Community suggestion structures also address the issue of cold starting which Occurs within a person's recommendation engine this work offers a report on the most recent technologies. Relevant to several areas of community optimization algorithms as for their accumulation and customer obvious sign designs.

The literature on book recommendation systems reflects a dynamic and evolving field with diverse approaches to address user preferences. Collaborative filtering, a cornerstone, leverages user interactions to generate recommendations. Memory-based methods, relying on user or item similarities, and model-based approaches, employing machine learning for prediction, have been extensively explored. Content-based filtering, emphasizing item features like genre or author, complements collaborative filtering.

**CHAPTER-3**

**SYSTEM REQUIREMENTS AND SPECIFICATION**

**3.1 System Requirement Specification:**

System Requirement Specification (SRS) is a fundamental document, which forms the foundation of the software development process. The System Requirements Specification (SRS) document describes all data, functional and behavioral requirements of the software under production or development. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two- way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and non-functional requirements only. It doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements.

**3.2 Hardware specification**

➢ RAM: 4GB and Higher

➢ Processor: intel i3 and above

➢ Hard Disk: 500GB Minimum

**3.3 Software specification**

➢ OS: Windows or Linux

➢ Python IDE: python 2.7.x and above

➢ Jupyter Notebook

➢ Language: Python

**3.4 Functional Requirements:**

User Registration and Authentication: Users should be able to create accounts securely, log in, and the system must ensure robust authentication mechanisms to protect user data.

Recommendation Engine: The heart of the system lies in its recommendation engine. Depending on the chosen approach, be it collaborative filtering, content-based filtering, or a hybrid method, the engine should generate accurate and personalized book recommendations for users.

User Interface: An intuitive and user-friendly interface is essential for users to interact seamlessly with the recommendation system. The interface should allow users to explore recommended books, provide feedback, and navigate effortlessly through the application.

Feedback Mechanism: Users should have the ability to provide feedback on recommended books. This feedback loop is crucial for refining the recommendation algorithms over time and improving the overall accuracy of the system.

**3.5 Non-Functional Requirements**

**Scalability**: The system should be designed to handle an increasing number of users and books without compromising performance. Scalability is crucial for accommodating growth and ensuring a consistently responsive user experience.

**Performance:** Quick response times and low latency are paramount for user satisfaction. The system should efficiently process requests and deliver recommendations in real-time to provide a seamless and enjoyable experience.

**Reliability:** High system reliability is essential to minimize downtime and ensure continuous availability. Users should be able to access the system reliably, without disruptions.

**Security:** The security of user data and system infrastructure is of utmost importance. Measures should be in place to protect against unauthorized access, data breaches, and other security threats.

**3.6 Performance Requirement:**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system onceit has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

**CHAPTER-4**

**RELATED WORK**

**4.1 Recommendation System**

Recommender system is defined as a tool that helps users in search something which is related to their tastes or as a strategy of choice for users under complex information environments. Recommender systems handle the problem of overload of information that users find, by providing them personalized and targeted content. For building this systems, distinct approaches have been developed.

They can be collaborative filtering, content-based filtering, or hybrid filtering. Collaborative filtering recommends items by identifying other users with similar tastes. On the other hand, Content-based filtering recommends elements that are similar in content to products that user appreciated in the past or matched to user attributes. The hybrid method, as said the name, combines both collaborative and content-based techniques to reduce and overcome some limitations of this approaches.

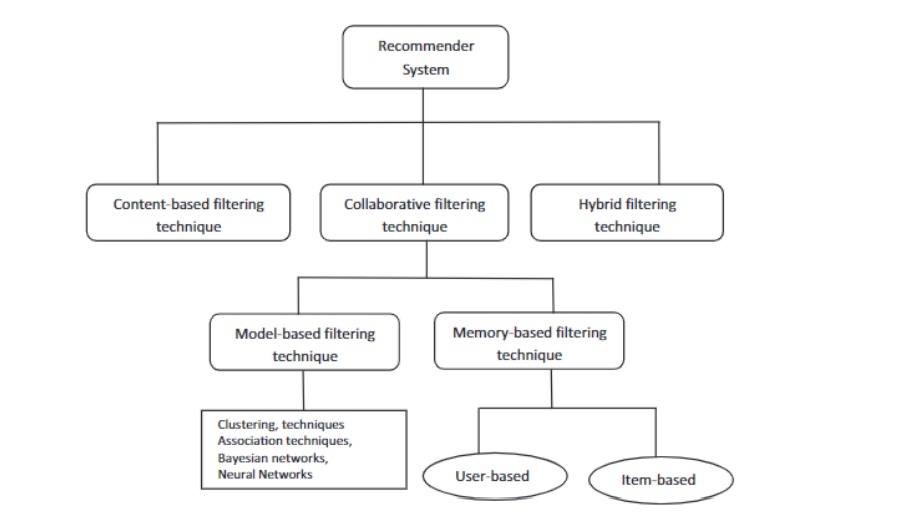


Fig 4.1 Recommender System

**Collaborative Filtering:**

Collaborative filtering is based on the assumption that the user’s future preferences can be predicted by their past preferences (acquired via feedback), in other words what they appreciated in the past will be appreciate also in the future. These past user-item interactions, build through user-item matrix, are sufficient to find matches and to detect similarity between users and/or items and make predictions according on these estimated distances. User-item matrix defines the match data of m users by n items that contains the user’s ratings over items. So, each entry (i, j) in the matrix represents the interaction between user i and item j Two items are considered to be similar if most of the users that have interacted with both of them did it in a similar way.

For measure the closeness, many algorithms are used like the K-nearest neighbour (K-NN) approach, the Cosine similarity, or the Pearson Correlation. I explained the details about the similarity metrics in the next subsection on Similarity Computation. Collaborative filtering methods, as we can see in Figure 2.1, are classified as memory-based or model-based. Memory based approaches works directly with values of stored interactions and these techniques can be of two types: user-based or item-based.

Depending on if we want to determine similarity between users or items Respectively

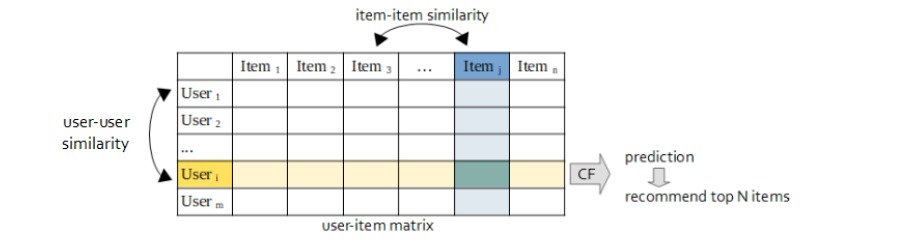
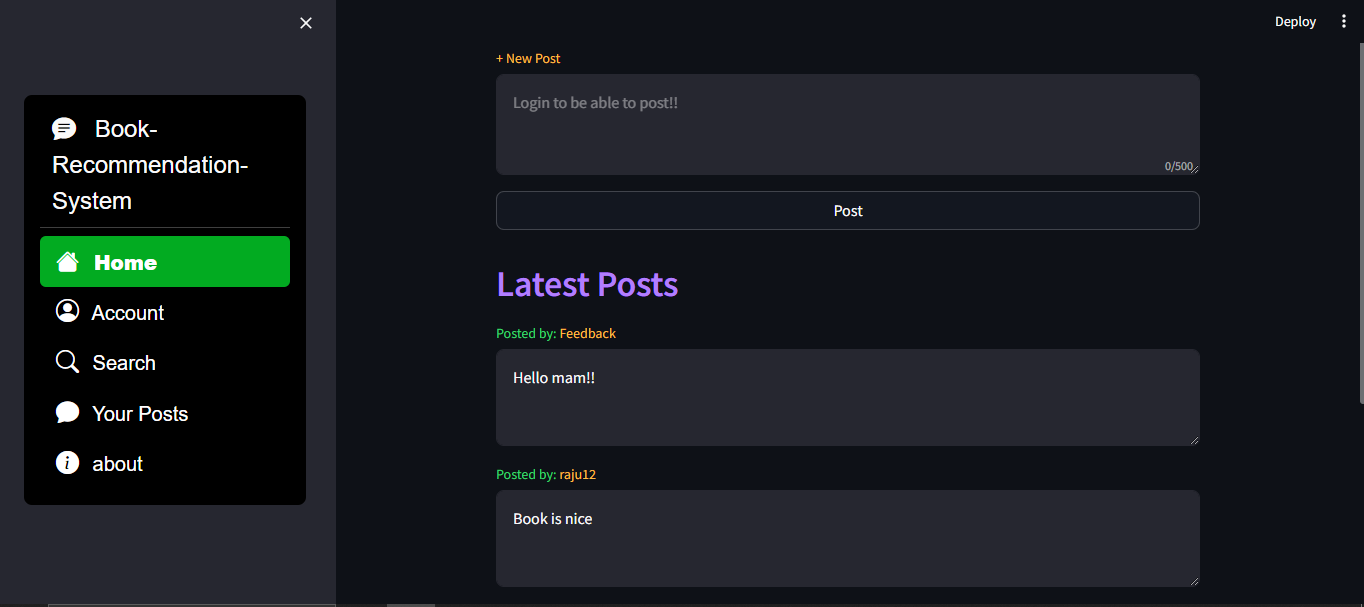


Figure 4.2: Collaborative filtering process

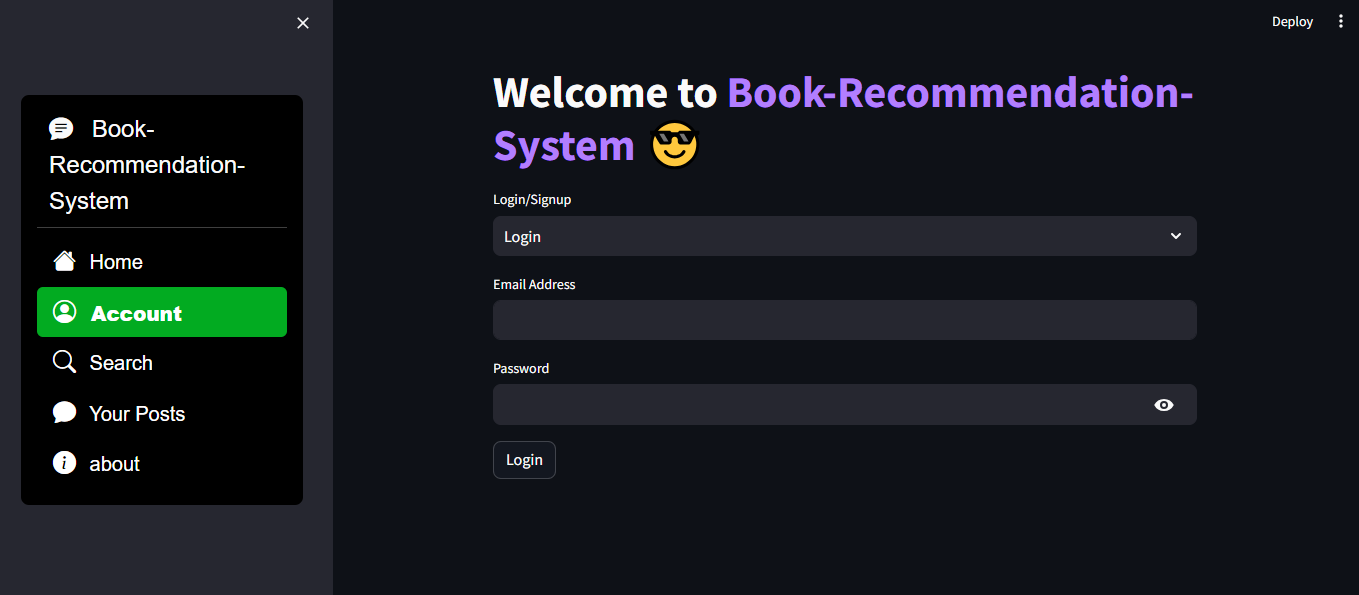
**CHAPTER-5**

**RESULT**

**5.1** This is the page where user can post anything about the book what he/she likes about it

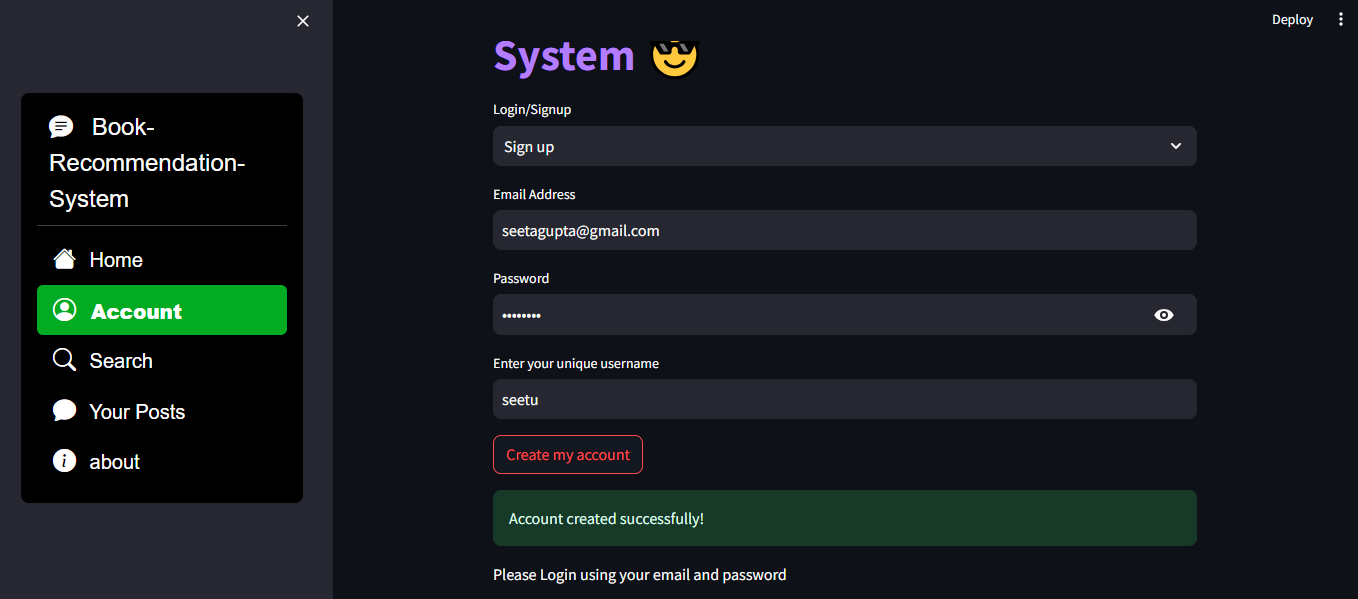
**Fig 5.1 Post**

**5.2** This is our login/Signup page where when a new user visits the site he/she signup or if an old user visits then he/she logins.



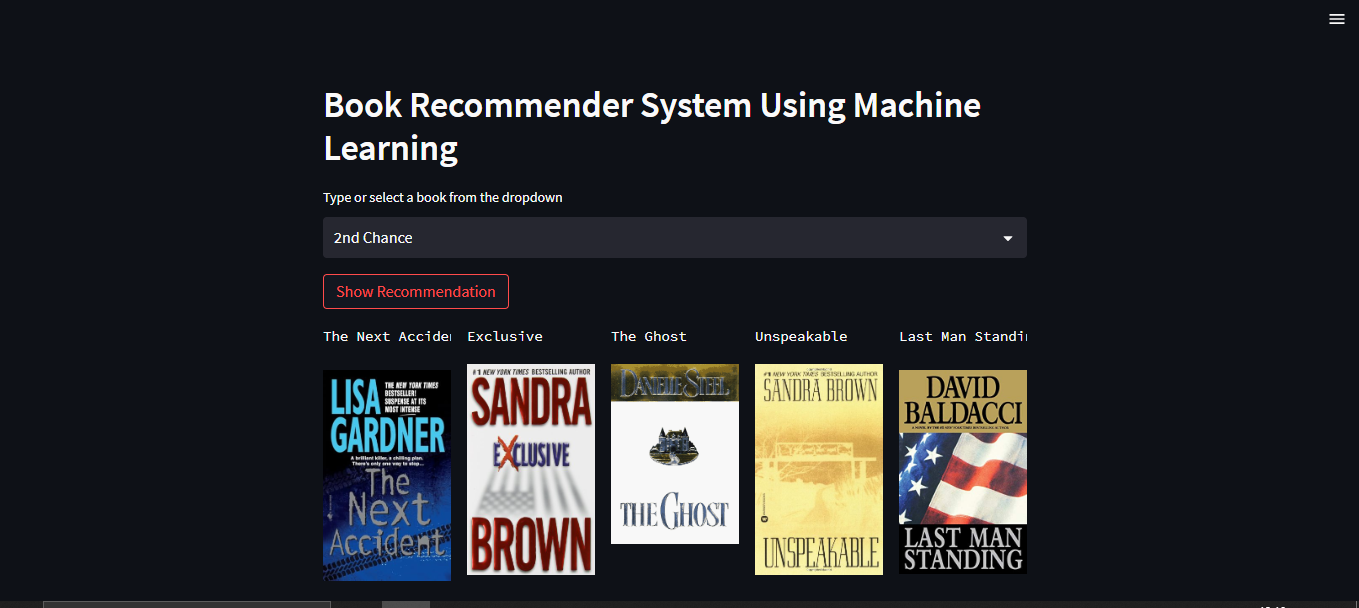
**Fig 5.2 Login Page**

**5.3** This is the demo of Signup page and it is done with appropriate Email, Password, and unique username.



**Fig 5.3 Signup**

**5.4** This is page of book-recommendation site where user can choose any book based upon his/her choice.



**Fig 5.4 Book-Recommendation Page**

**Advantages of Book Recommendation Systems:**

**Personalization:**

Book recommendation systems analyse user preferences, reading history, and behaviours to provide personalized suggestions. This enhances the user experience by offering books tailored to individual tastes.

**Discovery of New Content:**

Users are exposed to a diverse range of books, including genres or authors they may not have discovered on their own. This promotes exploration and broadens their reading horizons.

**Time-Saving:**

Recommender systems save users time by presenting them with a curated list of books that are likely to match their interests. This eliminates the need for extensive manual searching and browsing.

**Increased User Engagement:**

Users are more likely to engage with a platform that understands their preferences and consistently provides relevant recommendations. This can lead to increased interaction and prolonged usage.

**Enhanced User Satisfaction:**

When users find the recommended books aligning with their preferences, they are more likely to be satisfied with the platform. This positive experience contributes to user loyalty and retention.

**Improved Business Performance:**

For book retailers or platforms, an effective recommendation system can lead to increased sales and customer retention. By guiding users towards books, they are likely to purchase, these systems contribute to revenue growth.

**Cross-Selling Opportunities:**

Recommender systems can suggest books from different genres or categories, creating cross-selling opportunities. This encourages users to explore and purchase books beyond their usual preferences.

**Adaptability to User Changes:**

Recommendation systems can adapt to changes in user preferences over time. As users evolve in their reading habits, the system adjusts its recommendations accordingly, ensuring continued relevance.

**Efficient Content Curation:**

Manual curation of vast book catalogues can be overwhelming. Recommendation systems automate the process, ensuring that books are effectively categorized and presented to users based on their preferences.

**Community Building:**

Some recommendation systems incorporate social features, allowing users to see what books their friends or like-minded readers enjoyed. This fosters a sense of community and encourages social interactions within the platform.

**Conclusion**

Recommendation engines basically are data filtering tools that make use of algorithms and data to recommend the most relevant items to a particular user.

Recommendation system can be categorized into the following categories:

**a) Collaborative filtering:**

This type of recommendation system makes predictions of what might interest a person based on the taste of many other users. It assumes that if person X likes Snickers, and person Y likes Snickers and Milky Way, then person X might like Milky Way as well.

**b) Content-based filtering:**

This type of recommendation system focuses on the products themselves and recommends other products that have similar attributes. Content-based filtering relies on the characteristics of the products themselves, so it does not rely on other users to interact with the products before making a recommendation.

**c)Demographic-based recommender system:**

This type of recommendation system categorizes users based on a set of demographic classes. This algorithm requires market research data to fully implement. The main benefit is that it does not need a history of user ratings.

**d)Knowledge-based Recommender System:**

This type of system makes suggestions based on information relating to each user’s preferences and needs. Using function knowledge, it can draw connections between a customer’s need and a suitable product.

**e) Hybrid filtering:**

This type of recommendation system can implement a combination of any two of the above systems.

The recommendation system made in this project is able to recommend movies for a particular user-provided its user id is given. Our program fetches the Movie lens dataset, and then create and train a model using WARP loss function. It uses a hybrid approach that is the content-based and collaborative approach to recommend movies for a user appropriately. For the evaluation of our results, we can use two metrics of accuracy: precision and ROCAUC.

Both are ranking metrics: to compute them, we will be constructing recommendation lists for all our users, and checking the ranking of known positive books.

The need of recommendation system is: With the growing amount of information on the internet and with a significant rise in the number of users, it is becoming important for companies to search, map and provide them with the relevant chunk of information according to their preferences and tastes.

**Application of recommendation system:**

Almost nowadays all web service-based business uses recommendation system. Examples of popular recommendation systems are that of Netflix, Amazon, YouTube, Gaana Music App, Flipkart, eBay etc.

**Advantages of using recommendation system**: Recommendation systems can significantly boost revenues, CTRs, conversions, and other important metrics. Moreover, they can have positive effects on the user experience as well, which translates into metrics that are harder to measure but are nonetheless of much importance to online businesses, such as customer satisfaction and retention.

**FUTURE SCOPE**

The future scope of book recommendation systems holds considerable potential as technology continues to advance. Here are some aspects of its future development:

**Personalization and User Engagement:**

Enhanced personalization: Future systems are likely to become even more adept at understanding individual user preferences, incorporating factors like reading history, user feedback, and real-time contextual information to offer highly personalized recommendations.

Improved user engagement: Integrating features such as interactive interfaces, multimedia content, and social elements could make book recommendations more engaging and enjoyable for users.

**Integration of Emerging Technologies:**

AI-driven conversational interfaces: Integrating natural language processing and AI-driven chatbots can create more interactive and conversational book recommendation experiences.

Augmented Reality (AR) and Virtual Reality (VR): Future systems might leverage AR and VR technologies to create immersive reading experiences, allowing users to explore book recommendations in virtual environments. Social integration: Integrating with social media platforms and leveraging user-generated content for recommendations could become more prevalent, allowing users to share and discover books within their social circles.

**Continuous Learning and Adaptability:**

Real-time learning: Systems could evolve to adapt in real-time based on user interactions and changing preferences, ensuring that recommendations remain relevant over time.

Integration of user feedback: More sophisticated feedback loops could be established to continuously improve the accuracy of recommendations based on user input.