Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT REPORT

On

Book Recommendation System

Submitted to

Department of Computer Application

NIMS College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by
Susan Budhathoki
Reg. No. : 6-2-756-11-2019
2024/11

Under the Supervision of

Nirmala Timilsena

Tribhuvan University

Faculty of Humanities and Social Sciences

NIMS College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by Susan Budhathoki entitled "**Book Recommendation System"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Nirmala Timilsena

SUPERVISOR

BCA Department

Kanibahal Lalitpur, Nepal

Tribhuvan University

Faculty of Humanities and Social Sciences

NIMS College

LETTER OF APPROVAL

This is to certify that this project prepared by Susan Budhathoki entitled "**Book Recommendation System**" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

SIGNATURE of Supervisor Nirmala Timilsena Supervisor Nims College Kanibahal Lalitpur, Nepal	**SIGNATURE of HOD/ Coordinator** Shyam Kumar Thapa Coordinator Nims College Kanibahal Lalitpur, Nepal
Signature **Internal Examiner**	Signature **External Examiner**

Abstract

In today's digital age, everyone depending on reviews by others in many things such as selecting a movie to watch, buying products, reading a book. Recommendation systems are used for that purpose only. A recommendation system is a kind of filtering system that predicts a user's rating of an item. Recommendation system recommends items to users by filtering through a large database of information using a ranked list of predicted ratings of items. Online Book recommendation system is a recommendation system for ones who loves to read a book. When selecting a book to read, individuals read and rely on the book ratings and reviews that previous users have written. In the project, Collaborative techniques such as Clustering are used in which data-points are grouped into clusters. Algorithms such as K-nearest neighbour (KNN) are used. In this method the Users similarity with all other users is determined using some similarity measure like Euclidean distance/Pearson Correlation/Cosine Similarity based on the common items they have rated. Top rated items are recommended to the user.

Keywords: Book Recommendation System, Cosine Similarity, Machine learning

Acknowledgement

Firstly I would like to express my special thanks to my teacher **Nirmala Timilsena** and **Raj Kadel** for helping me to develop excellent vision and idea for this project. I would also like to thank **NIMS college** and **TU** who gave me the golden opportunity to do this project, which helped me in doing a lot of research and I came to know about so many new things related to this project.

I am very thankful towards my friends who helped me a lot during this project. Without their help and support I wouldn’t have complete this project.

I would also like to thanks my parents and teachers for the support and encouragement they have given while finalizing this project within the limited time period.

Table of Contents

List of Abbreviations

Table 1: List of Abbreviation

Abbreviations	Full Form
BRS	Book Recommendation System
CF	Collaborative Filtering
KNN	K-Nearest Neighbour
SDLC	Software Development Life Cycle
TU	Tribhuvan University
UI	User Interface

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

A recommendation system is a type of information filtering system which predicts the rating or preference that a user would give to an item. A book recommendation system is a type of recommendation system where we have to recommend similar books to the reader based on their interest. Now a days online rating and reviews are playing important role in book sales. Readers only buy or read books depending upon the review and rating by others. So this recommendation system focuses on the reviews and ratings by others and filters the books.

In this project collaborative filtering recommendation system is used. Collaborative filtering is a popular techniques in recommendation system which assumes that the users who have similar tastes in the past will continue to have similar preference in the future also. User ratings are used to determine the user or item similarity in collaborative filtering. In collaborative filtering, recommendations are based on similarities between the active user and other users, or the similarities between items ratings. Collaborative filtering has become the most commonly used method to recommend items for users. Collaborative filtering is a technology to recommend items based on similarity. There are two types of collaborative filtering: User-based collaborative filtering and Item-based collaborative filtering. User-based collaborative filtering algorithm is an effective way of recommending useful contents to users by exploiting the intuition that a user will likely prefer the items preferred by similar users. Therefore, at first, the algorithm tries to find the user's neighbors based on user similarities and then combines the neighbor user's rating score by using supervised learning like k-nearest neighbors algorithm and k-means algorithm. Item-based collaborative filtering algorithm fundamentally has the same scheme with user-based collaborative filtering in terms of using user's rating score. Instead of the nearest neighbors, it looks into a set of items; the target user has already rated items and this algorithm computes how similar items are to the target item under recommendation. After that it also combines user's previous preferences based on these item similarities.

A book recommendation system is a powerful tool that helps users discover new books based on their interests and preferences. The primary goal of a book recommendation system is to enhance the user's reading experience. With the help of machine learning techniques and collaborative filtering methods, book recommendation systems play a vital role in recommending books to the users.

1.2 Problem Statement

The web recommendation system have many challenges. When a user first joins the recommendation website, system does not know anything so it struggles suggesting books to the user. As many people join the platform and more books get added, the system takes long time to find suitable recommendations. Users may worry about their privacy when it comes to sharing personal preferences. Some users may give fake ratings to certain books, making recommendation less accurate. And also the system promotes popular books neglecting other books.

Solution

The system use a mix of collaborative filtering and popular recommendation to recommend books to the users according to the rating and user preferences. By using KNN(K-Nearest Neighbors) algorithm the system speed up the process and recommends the books to the users. The system also filters the fake rating and helps in recommending the proper books to the user according to their similarities and interest. By regular checking and updating the algorithm the system provides the accurate books to the users.

1.3 Objectives

The objectives of a book recommendation system can vary depending on the specific context and goals of the system. The main objective is

- To recommend books to the users based on similar rating.
- To recommend relevant books using Machine Learning Model.

1.4 Scope and Limitation

The scope of Book Recommendation System is that the system recommends books to the user according to the the rating and their interest. First the user have to register and login to the system. Once the user login to the system then they can access the system which creates the relation between the user and the system. This system is a online system so it provides online platform to the user, which can be accessed through internet and can view and search books then the system will recommend books according to the user search. Then the user can search for the books according to their interest and the system will recommend the books to the user according to the rating similarities. The system is easy to use and does not require vast or skilled knowledge to use it. This system provides login facilities to the user due to which they can view and search book as their interest.

The limitation of Book Recommendation System is that it is collaborative filtering based recommendation and it recommends books to the user according to the rating similarities only. This system only works when the user rates the books and it is dependent on user rating. Sometimes the user may not rate the book so the result may not be accurate.

1.5 Development Methodology

In this project, this system has used Waterfall model for the Book Recommendation System development because it works effectively for small-scale projects with consistent and well-defined criteria. The project was conceived in the first phase, during which the objectives of the Book Recommendation System were specified. In the analysis phase, user needs were thoroughly examined, and clear details were defined about what the system should do (functional requirements) and how it should perform (non-functional requirements).

The system architecture, database structure, and user interface were carefully developed during the design process, using Flask because of its efficiency and simplicity. The actual implementation of the intended system took place during the ensuing building phase. This included the use of Python and Flask to develop the Book Recommendation System. Testing was essential to ensure the system's quality. Unit testing checked individual components, integration testing verified how modules worked together, and system testing ensured the entire system worked as expected. After testing, the maintenance phase began to fix any bugs, consider user feedback for improvements, and plan future updates.

Before finishing the project, a final review ensured all requirements were met, and users received a fully functional Book Recommendation System. Detailed documentation was maintained throughout each phase, providing a clear record for future reference. The Waterfall Model worked well for this small project, allowing a step-by-step and organized development process.

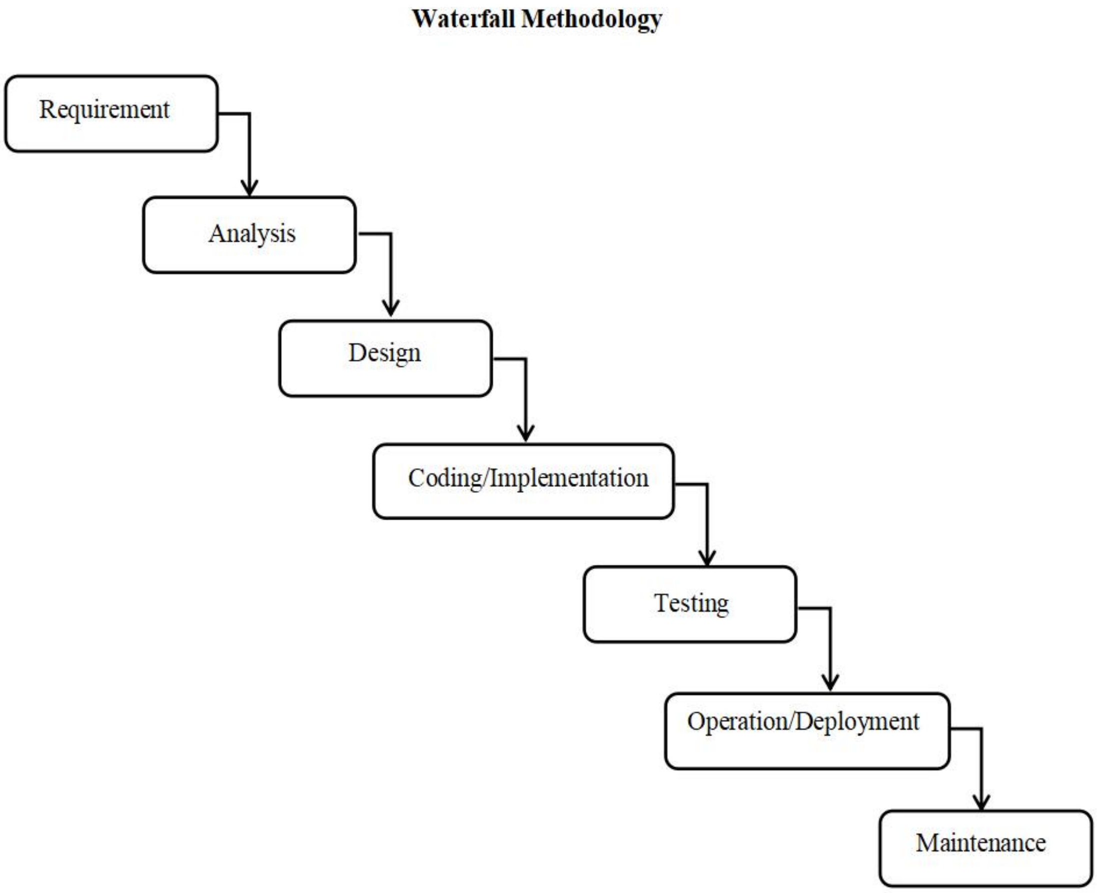


Figure 1: Chapter 1.5 Waterfall Methodology

1.6 Report Organization

Chapter 1 Introduction

This chapter provides an overview of the research project. It introduces the problem statement, the objectives of the study, and the development methodology. It also outlines the structure of the report and provides a brief summary of each subsequent chapter.

Chapter 2 Background Study and Literature Review

In this chapter, a comprehensive review of relevant literature is presented. It explores existing research, technologies, and methodologies related to book recommendation system. The chapter discusses various approaches, their strengths, weaknesses, and gaps in the field, providing a foundation for the current study.

Chapter 3 System Analysis and Design

This chapter delves into the analysis of requirements and design considerations for the book recommendation system. It outlines the system's functional and non-functional requirements, user needs, and potential challenges. The design phase covers the architecture, components, and algorithms chosen for the system, detailing how they will address the identified requirements.

Chapter 4 Implementation and Testing

Here, the practical implementation of the book recommendation system is described. The chapter covers the development process, coding practices, and technologies used to bring the system to life. It discusses the testing methodologies employed, including unit testing, and system testing, to validate the system's performance and accuracy.

Chapter 5 Conclusion and Future Recommendations

The final chapter wraps up the report by summarizing the findings and outcomes of the research. It discusses the achievements, challenges, and limitations encountered during the project. Additionally, this chapter provides insights into potential future enhancements, recommendations for refining the system's accuracy and usability, and its applicability to real-world scenarios.

Overall, the report follows a logical structure that takes the reader through the journey of the research project, from the initial introduction to the final conclusions and suggestions for future work. Each chapter contributes to building a comprehensive understanding of the book recommendation system and its significance in the context of agriculture and technology.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Information has always been an inseparable part of nature. To understand the essence of information we had to wait until the 20th century. Shannon has also discovered that information can be transformed to numbers, or precisely to two state systems, thus creating the basic unit of the new, information-based world order. Considering its content one bit is 0 or 1, yes or no, exists or not, but its power is that with the chain of these we can code anything. By digitizing things mankind has achieved that information became not only lasting, but can be stored in virtually endless quantities, and at the same time we are able to distribute it extremely fast.

In parallel with the spreading of computers used for civilian purposes satisfying user expectations on a continuously growing scale has gotten into the focus of development companies and researchers.

Behind the rapidly increasing popularity of machines extremely serious efforts lie, which has been made to reduce “friction” between man and machine. Increasingly comfortable solutions have been created because attempts have been made to understand people’s needs and personalize the services provided by computers.

The basics of recommender systems were founded by researches into cognition science and information retrieval, and its first manifestation was the Usenet communication system created by Duke University in the second half of the 1970s, where users were able to share textual content with each other. These were categorized into newsgroups and subgroups for easier search, but it was not directly built on or targeted the preferences of users.

The first known such solution was the computer librarian Grundy, which first interviewed users about their preferences and then recommended books to them considering this information. Based on the information collected the system allocated the user into a stereotype group using a rather primitive method, thus recommending the same books to all persons in the same group. For more information about the results of Grundy’s solution and its popularity among users, see Rich’s 1979 article. This approach may seem a little outdated today, but at the time it was a paradigm shift in automated services, since it was personalised. It is important to note that this milestone has not been reached by all webshops, even now.

However, Grundy's solution quickly gained a lot of critics in the scientific world. Nisbett and Wilson state that "people are very weak in the study and description of their own cognitive processes". According to their studies, people often highlight their attributes which make them stand out from the rest of a particular group, making stereotyping efforts more difficult. Of course, it could happen that we simply want to create a different image of ourselves.

As Heli Vainio, the manager of one of Northern Europe's largest shopping malls phrases a little harshly in her earlier interview: "People respond to questionnaires in a way that makes them look better. I don't care about lies. I'm interested in facts". That's why she's equipped her shopping centre with Wi-Fi equipment that can track visitors within and in the immediate vicinity of the building with an accuracy of 2 meters. The goal is that instead of the visitors their actions should talk.

Basically two very different directions of recommender systems have evolved over time: collaborative filtering and content-based filtering. The former attempts to map the taste of users and offers content to them that users with similar preferences liked. The content-based filtering is about knowing the dimensions of the entity to be recommended (for example, a musical content recommendation system can consider the following dimensions: style, artist, era, orchestration, etc.) and the user's preferences for these dimensions or characteristics. Thus, every time a user likes another song, this new information is added to their profile.

The first example of collaborative filtering and also the origin of the term was the Tapestry system developed by Xerox PARC, which allowed its users to take notes and comment on the documents they were reading (initially in binary form: liking or disliking it). Therefore, users could not only use the content of the documents to manually narrow their search, but on the basis of notes and reviews from other users, which once reaching an appropriate number of users, was able to rank the thematic documents rather well on the basis of their relevance and usefulness.

GroupLens, which started in 1992, was already able to make automated recommendations for Usenet articles if the user had already evaluated some articles in the system. In the following years, many thematic recommender sites were developed, such as Ringo developed at MIT, later the Firefly music recommender pages, or the BellCore movie proposer system.

The first solution, which not only tried to encompass narrower topics, but no less than the Internet itself was Yahoo!, under a different name then. The two Stanford students created a thematic website catalog with indexed pages, which quickly gained popularity and meant easier searches for millions on the Internet, and based on the Alexa rankings it is still ranked as the fifth most visited website.

The roots of content-based filtering have to be looked for in the information retrieval field, from which many techniques have been transferred. The first documented solution came from Emanuel Goldberg in the 1920s (if not including the Jaquard loom presented in 1801, the predecessor of the Hollerith punchcard), which was a "statistical machine" that attempted to automatically find documents stored on celluloid tapes by searching for patterns.

In the 1960s, thanks to a team of researchers at the University of Cornwall gathering around Salton, a model for automatic indexing of texts has been created over nearly a decade, which forms the basis of the text mining processes we know today. The procedure is very simple, documents are classified along certain predetermined criteria that are collected into a vector as indices. The more similar two documents are to each other, the smaller the angle of the vectors describing them are.

The next milestone was the CITE online catalogue system developed in 1979 by Tamás Doszkocs for the National Library of Medicine, which not only allowed users to search for books by category, but sorted them by relevance based on search terms.

Content-based filtering has gained raison d'étre relatively late in the 1990s as a division of information retrieval. The main reason for the delay was that creating a well-functioning content-based filtering system, even in a particular subject, is a huge challenge, since the task is nothing less than to "understand" the examined topic and the factors influencing the relationship of the users to the discipline.

One of the first and quite successful research on this subject was the Music Genome Project in 1999, which aims to "understand" and capture music through its properties. To this end, more than 450 such properties have been revealed and their relations have been described using an algorithm. The basis of the procedure is that when the user likes a song, positive values are assigned to its specific properties (such as style, era, artist, orchestration, beat, etc.). Songs with similar properties will then be further promoted on the preference list and brought to the user's attention. The huge advantage to collaborative filtering is that very little information is sufficient at startup, while the former unfortunately requires a lot of users and feedback to identify people with similar tastes. However, the disadvantage is that usually it can hardly, or not at all make recommendations outside the user's music lists, since it is not building on the similarities between users, only the “understanding” the properties of music as an entity. Pandora Internet Radio, which has 250 million users, is based on this project even today.

The first solution to combine collaborative and content-based filtering solutions was Fab developed by Stanford students, presented in 1994. They point out that their objective with the hybrid system is to eliminate the disadvantages of the two procedures which became known by that time. Their model consists of two basic processes: first they collect content for specific topics (such as websites or articles about financial topics), then for each individual user they select those items collected from specific subjects which highly likely will interest them specifically and finally these contents will reach the user.

Combining the two approaches can be conceived in several ways, one procedure can be embedded in another, as Fab’s example shows, or it is possible to give a joint recommendation as a result of the two procedures, as Netflix does. Netflix’s algorithm, CineMatch was the most successful recommender system for online movie sales in the early 2000s. It was a serious catalyst for such research and the scientific field – which only started its independent existence in the 90s – started rapidly to develop. The 2006 Netflix Award’s challenge was to create a recommender algorithm based on the 100 million film reviews made available by them, which makes recommendations at least 10% better than the results of CineMatch. The 1 million dollar prize in 2009 was awarded for a solution that included 107 different algorithms and mixed their recommendations depending on the circumstances. We can’t omit the biggest example of online referral systems today, amazon.com, which recommends products to the user based on a cooperative filtering technique, taking into account previously browsed and purchased products and what they are currently viewing.

This technique is now used by many webshops in order to improve their sales figures. The purpose of the recommender systems operating in the online customer space is to customize the storefront according to the current taste and need of the customers. This creates an almost unbeatable competitive advantage for online shops over traditional brick-and-mortar store purchases, whose only forte is that the product can be touched by hand or tried on. Nowadays, it is more typical that after a fitting in the shop the products are ordered online, or using the product return option are bought without tryout.[1]

2.2 Literature Review

Case 1:

Ms. Praveena Mathew, Ms.bincy kuriakose and Mr. Vinayak hedge [3] proposed a Book Recommendation System (BRS) through the combined features of content based filtering (CBF), collaborative filtering (CF) and association rule mining to produce efficient and effective recommendation. The existing systems lead to extraction of irrelevant information and lead to lack of user satisfaction. So, they proposing a hybrid algorithm, which combines two or more algorithms, to help the recommendation system to recommend the book based on the buyer's interest. They use association rule mining algorithm, ECLAT (Equivalence class clustering and bottom up lattice traversal). ECLAT will helps to find out the frequent item set. It uses depth first searching technique. In one scan, it will categorise. Cosine similarity is used for the similarity measuring in content and collaborative filtering. They use item-item filtering in collaborative filtering. The basic finding that achieved through this proposed work is to recommend the books based on the buyer’s interest and increase the productivity and credibility. Using association rule mining algorithm to finds interesting association and relationship among large data set of books and provides an efficient recommendation for the book.

Case 2:

Yongen Liang and ShimingWan [4] proposed a method, which can mine products by understanding the user’s preferences. It is a personalised technology with collaborative filtering. It is book recommendation system, which is for a university library. Here only provide the recommendation service to the registered users. The collaborative filtering uses both user-user filtering and item-item filtering. The important job of the collaborative filtering is to calculate the similarity of the books and users or reader then, recommend. Cosine similarity is using for the similarity measuring in collaborative filtering. Then find out or predict the rating for the particular book, which the targerted user may like or give. One of the most important problems of collaborative filtering is cold start. That is, when a new user joins then they have no data about that user. They have no previous purchase history or borrow history. Therefore, here they propose a solution that Expert and new book recommendation. Expert and new book recommendation module will recommend the books as if Best-selling, newbooks arrived, classical books... in short, it will recommend the books at the top rating or popular books.

Case 3:

Dharmendra Pathak, Sandeep Matharia and C. N. S. Murthy [4] proposed an efficient and best unique hybrid recommendation algorithm, by providing the recommendation more satisfying the user's desire. Here the hybrid recommendation is a combination of collaborative, content and context based recommendation algorithms. The main input of collaborative filtering is rating i.e, votes of so many people, content based data that is the information about the users like their interest, date of birth, priorities... and the context based data that is the behavioural datas like date, taste, mood,weather... Cosine similarity is using for the similarity measuring. There are subject priorities according to the user's previous history. If they purchase a book then check, the purchased book is different subject priority from the subject priority has already set? If yes, then reset the subject priority3 and then subject priority 2. The subject priority1 will not change. Based on calculations and results they concluded that the proposed Hybrid book recommendation algorithm is best among the others.

Case 4:

Adli Ihsan Hariadi, and Dade Nurjanah [5] proposed a hybrid-based method that combines attribute based and user personality based methods for book recommender system. In this paper, they are implementing the MSVMSL (Most Similar Visited Material to the Most Similar Learner) method, and they are saying that, it is the best method among hybrid attributes based methods. The personality factor is used to find the similarity between users when creating neighbourhood relationships. The hybrid attribute will calculates the recommendation scores of rated books from neighbors using the similarity scores between a target book and its neighbors and between the active user and that user's neighbours. The score of book b from user u, denoted as score_b. This is for finding the Most Similar Visited Material to the most Similar Learner. It uses the values from both content and collaborative. Then use the result of hybrid as recommendation. That is the Most Similar Visited Material to the most Similar Learner.

Case 5:

Anand Shanker Tewari, Abhay Kumar and Asim Gopal Barman [6] proposed a book recommendation system based on combined features of content filtering, collaborative filtering and association rule mining. When a buyer search for a book, then it will be store as a purchase history or a search history. When the buyer is offline the recommendation perform some filteration for reccommending to buyer and the results are stored in the buyers web profile. When the buyer comes online next time, the recommendations will be generated automatically. In content based filtering, web Usage Mining (WUM) is used to provide relevant information to the buyers. web Usage Mining (WUM) typically extracts knowledge by analyzing historical data such as web server access logs, browser caches, or proxy logs. It helps to possible to model user behavior and, therefore, to forecast their future movements. web Usage Mining stores the user's behavior on the internet and processes that data. Item based collaborative recommendation Algorithm is using and Cosine similarity is using for the similarity measuring. Intersect the results from the association rule mining and the content, collaborative filtering.

Chapter 3: System Analysis and Design

3.1 System Analysis

A system analysis is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that users will have with the system. In addition to use cases, the system requirement analysis contain functional requirements, which define the internal workings of the system: that is, the calculations, technical details, data manipulation and processing, and other specific functionality that shows how the use cases are to be satisfied. It also contains non functional requirements, which impose constraints on the design or implementation (such as performance requirement, design constraints).

3.1.1 Requirement Analysis

Requirement analysis for a book recommendation system typically involves understanding the goals, functionality, and constraints of the system.

i. Functional Requirements

Functional requirements for a collaborative filtering book recommendation system typically involve specifying the features and capabilities that the system must have to fulfill its purpose effectively. Here are some key functional requirements for a collaborative filtering book recommendation system:

1. **User Interaction:** Users can rate books on a scale 1 to 5
2. **Collaborative Filtering algorithm:** The system analyzes user behavior to identify similar users or items for recommendations.
3. **Privacy and Security:** User information is kept safe, and the system makes sure to follow privacy rules.
4. **Search and filtering Options:** User can search for specific books or authors. It implements filters such as genre, author, or publication date to enhance search capabilities.
5. **Book Database:** The system keeps a big list of books with details like author, genre, and when they were published.
6. **Get Presonalized Recommendation:** User get suggestions for books based on what similar users liked.

7. User Interface: The system allows user-friendly interface that allows user to browse books, and receive recommendation.

Use Case Diagram

A use case diagram is a visual representation that depicts the interactions between actors (users or external systems) and a system under consideration. It provides a high-level overview of the functionality and behavior of a system, focusing on the external interactions rather than the internal implementation details.

Admin: Represents a person responsible for managing the book-related content in the system.

User: Represents an individual who uses the system to browse, search for, or get recommendations for books.

Register: User have to register to access the system.

Login: Both Admin and User can log in to access their respective functionalities.

Add Book: The Admin can add new books to the system.

Update Book: The Admin can modify the details of existing books.

Delete Book: The Admin can remove books from the system.

Book Search: Both Admin and User can search for books in the system.

Recommend Book: The system provides book recommendations to the User.

Logout: Both Admin and User can log out from the system after completing their tasks.

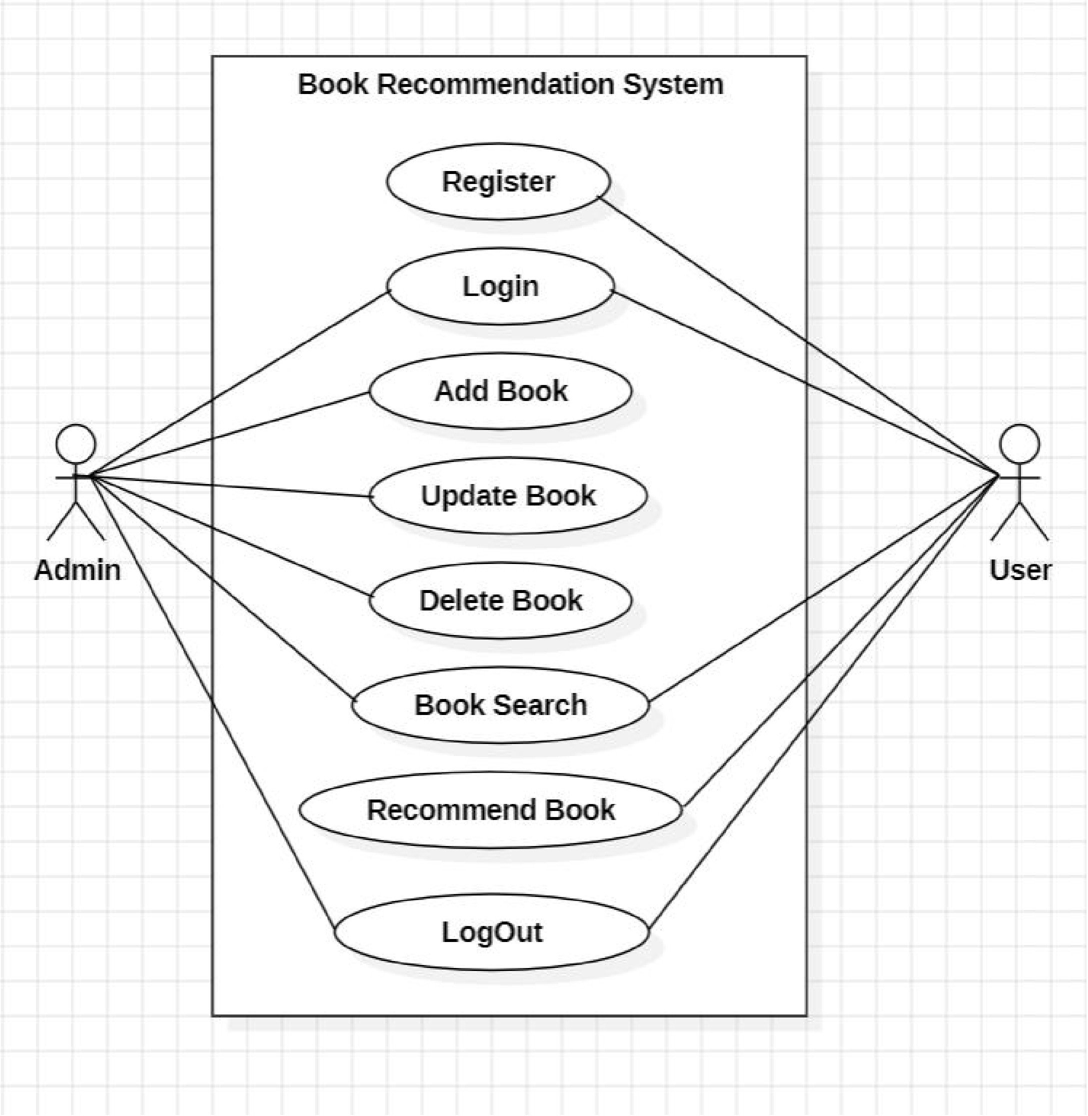


Figure 2: Chapter 3.1.1 Use case

ii. Non-Functional Requirements:

Non-functional requirements, also known as quality attributes or system qualities, describe the overall behavior, performance, and characteristics of a software system.

Performance: The system should respond quickly to user requests, providing recommendations in a timely manner. The response time should be optimized to ensure a smooth user experience, even when handling a large number of concurrent users or processing extensive datasets.

Scalability: The system should be capable of handling an increasing number of users and a growing book catalog without significant degradation in performance.

Availability: The system should be highly available, ensuring that users can access book recommendations at any time.

Reliability: The system should provide accurate and reliable recommendations consistently. It should handle errors and exceptions gracefully, avoiding crashes or incorrect results. It should have backup and recovery mechanisms to ensure data integrity.

Security: The system should protect user data and ensure privacy. It should implement authentication and authorization mechanisms to control access to user profiles and prevent unauthorized access or data breaches.

Usability: The system should have a user-friendly interface that is easy to navigate and understand. The recommendations should be presented in a clear and intuitive manner, allowing users to provide feedback and refine their preferences.

Maintainability: The system should be designed and implemented in a modular and maintainable manner. It should support easy maintenance and updates, enabling the addition of new features.

3.1.2 Feasibility Study

i. Technical Feasibility

The technical feasibility of collaborative filtering book recommendation system is evident as it is implemented in Python as backend programming language and Flask as the frontend framework. All codes and modules can be easily executed. The system not only supports the existing architecture but is also compatible with proposed enhancements. Its user-friendly nature and ease of implementation make it a technically feasible solution.

iii. Operational Feasibility

A collaborative filtering book recommendation system is quite practical to use. It smoothly fits into existing platforms and people like it because it gives book suggestions based on what they like. It can handle more users and books without causing too many problems, and the costs to run it are reasonable. By listening to user feedback, the system gets better at making recommendations over time. It can be customized to fit different needs, making it flexible for different situations. Even when dealing with new users or books, it still works well. It does not need a lot of complicated technical support, and it keeps your information safe. Overall, it is a user-friendly and effective way to help people find books .

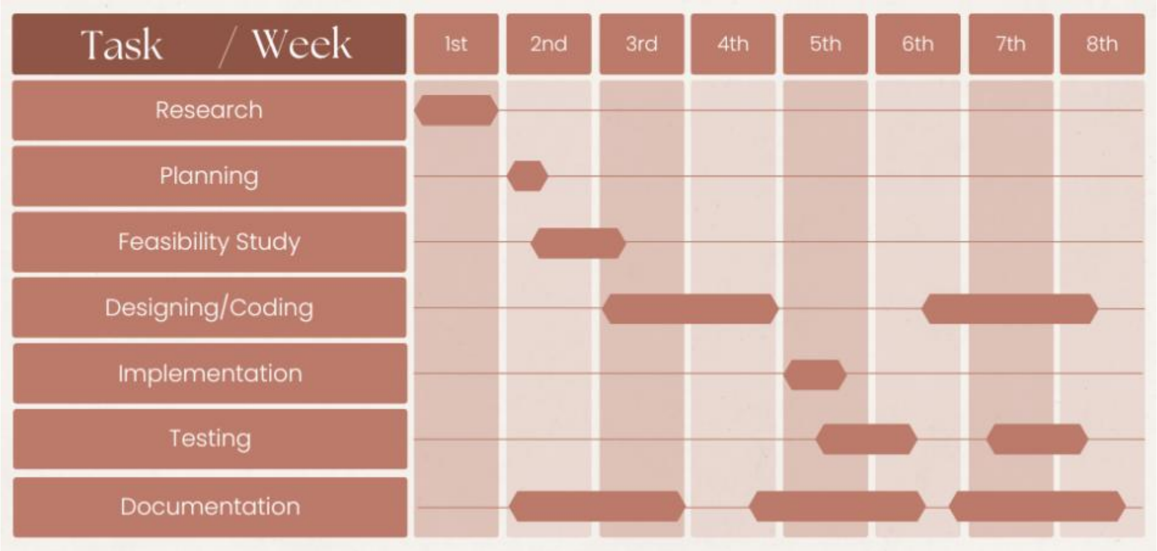
iii. Economic Feasibility

A collaborative filtering book recommendation system makes good economic sense. It helps people discover books they're likely to enjoy, which can increase sales for bookstores or online platforms. While there might be some initial costs to set it up, like hiring developers or using specific software, the benefits usually outweigh these expenses. People are more likely to buy books when they get personalized recommendations, leading to increased revenue. The economic feasibility lies in the system's ability to enhance user experience, fostering customer retention and over time.

iv. Schedule Feasibility

From a schedule perspective, implementing a collaborative filter book recommendation system is feasible. It is most important for completion of the project on time. The process involves steps like collecting data, creating algorithms, and testing recommendations. The simplicity of the approach contributes to a manageable schedule, making it feasible to deploy the recommendation system in a reasonable time-frame. Gantt chart is a project management tool that illustrates work completes over a time period of time in relation to the time planned for the work. It includes start and end dates of task.

Table 2: Chapter 3.1.2 Gantt Chart



Task / Week	1st	2nd	3rd	4th	5th	6th	7th	8th
Research	●							
Planning		●						
Feasibility Study		●	●					
Designing/Coding			●	●		●	●	●
Implementation					●			
Testing					●	●	●	●
Documentation		●	●	●	●	●	●	●

3.1.3 Object Modelling using Class and Object Diagrams

The User class represents each user in the system who can rate books. The Book class stores information about each book. Rating is used to capture user ratings for each book. The Recommendation Engine interacts with user preferences and data to generate book recommendations. The Collaborative Filter class performs user similarity calculations to identify similar users and produce recommendations based on other users preferences. This setup enables collaborative filtering, with Recommendation Engine and Collaborative Filter working together to provide recommendations based on user similarity and preferences.

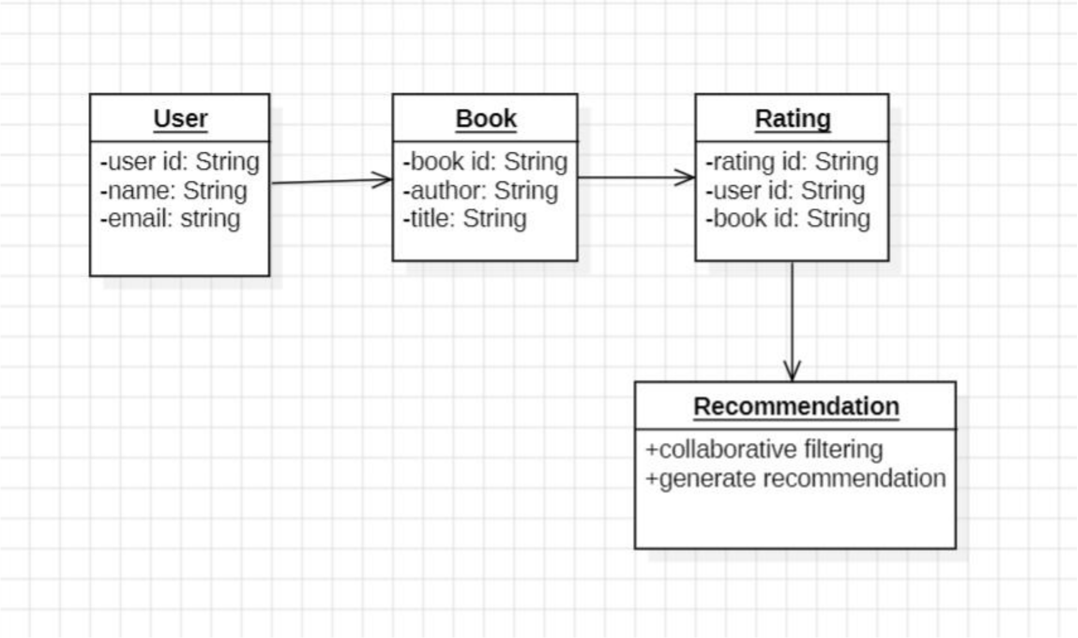


Figure 3: Chapter 3.1.3 Class and Object diagram

3.1.4 Dynamic Modelling using State and Sequence Diagrams

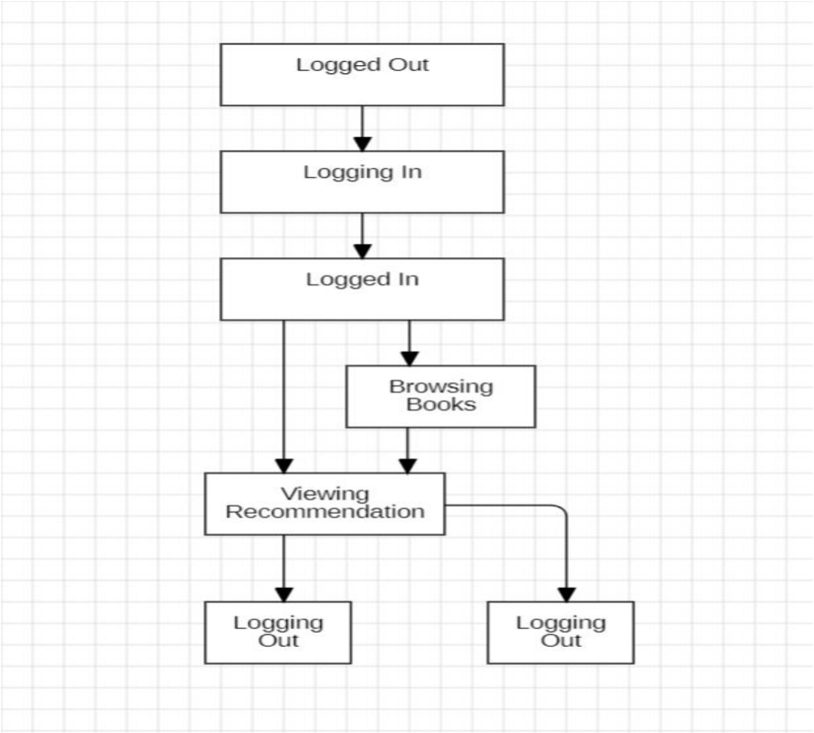


Figure 4: Chapter 3.1.4 State diagram

In State diagram first the user is not logged in. Then user is attempting to log in. After the user is successfully logged in. The user is viewing available books. The user is viewing book recommendations and at last the user is logging out of the system. The state diagram of book recommendation shows the different states of a User object when interacting with the system.

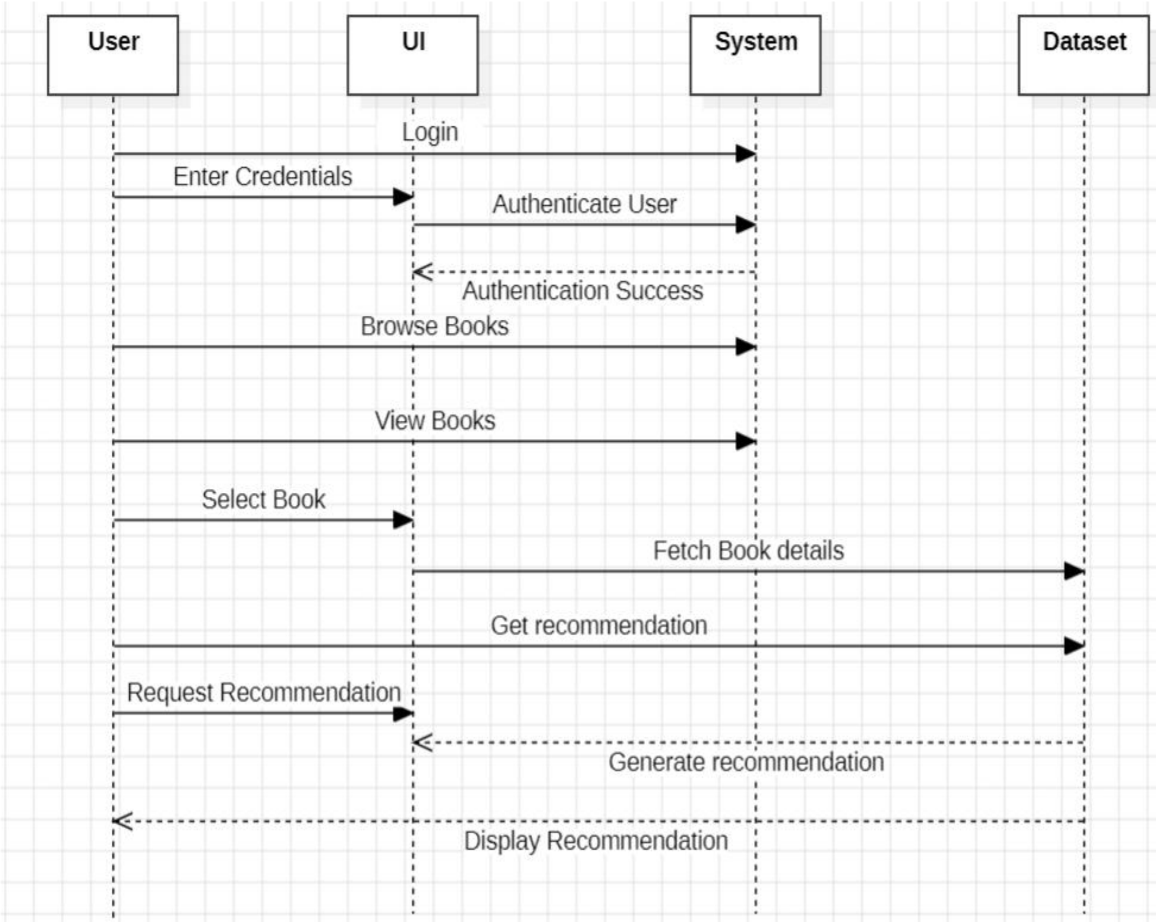


Figure 5: Chapter 3.1.4 Sequence diagram

In sequence diagram of book recommendation it show the interactions where a user views books and gets recommendations. The user logs into the system, entering credentials and getting authenticated. The user browses through the available books. The user selects a book to view its details. The user requests book recommendations based on existing ratings data, and the system retrieves and displays these recommendations from the dataset.

3.1.5 Process Modelling using Activity Diagrams

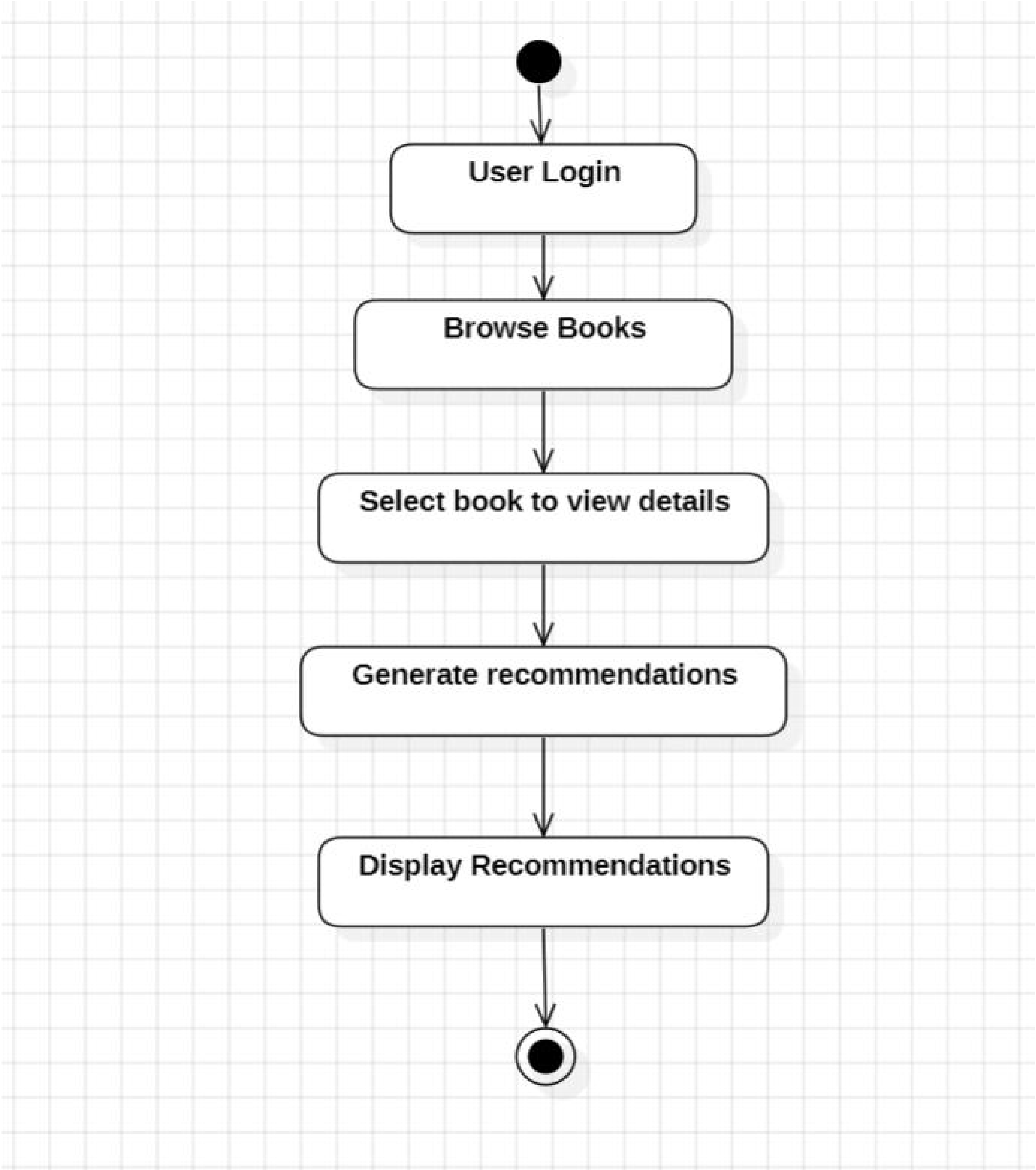


Figure 6: Chapter 3.1.5 Activity Diagram

In collaborative book recommendation system, users log in by entering their credentials. The system authenticates the user and after successful login, allows them to browse available books. Users can select a book to view its details. Based on the user's previous ratings stored in the dataset, the system generates and displays book recommendations. The process flow continues until the user decides to log out or exit the system.

3.2 System Design

3.2.1 Refinement of Class, Object, State, Sequence and Activity diagrams

The Class Diagram defines the core structure of the system, identifying key classes such as User, Book, Rating, and Recommendation. It outlines their attributes (like userID, bookID, rating) and methods (such as login(), rateBook(), generateRecommendation()), as well as the relationships between these classes, showing how they interact with one another. The Object Diagram: This offers a snapshot of the system at a particular instance, representing specific objects like an individual user and their rated books. It shows how these objects are instantiated from their respective classes and how they relate in a given context, such as user Alice having rated books "1984" and "Brave New World". The State Diagram illustrates the different states a User object can be in, such as Logged Out, Logging In, Logged In, Browsing Books, Rating a Book, and Viewing Recommendations. It also shows the transitions between these states based on user actions and system responses. The Sequence Diagram details the interactions over time between the user, system, and dataset. It follows a user through logging in, browsing books, selecting a book to view details, and requesting recommendations, illustrating how these components communicate and the order of operations. The Activity Diagram outlines the flow of activities within the system, from user login, through browsing and viewing book details, to generating and displaying recommendations using the KNN algorithm.

3.2.2 Component Diagrams

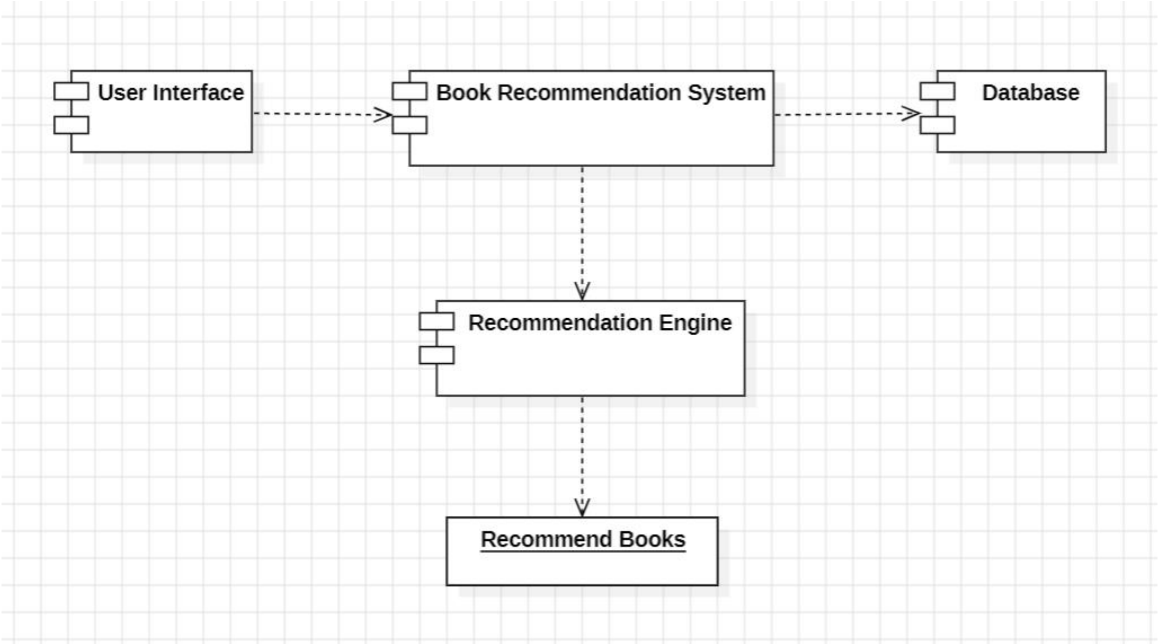


Figure 7: Chapter 3.2.2 Component Diagram

The User Interface component represents the part of the system that user interact with. It manages user interactions, including login, registration, browsing books, and viewing recommendations. The Recommendation Engine component is responsible for the core functionality of recommending books. It contains algorithm, data processing, and logic to generate book recommendations using the provided dataset. The Database manages access to user ratings data. The Dataset contains user ratings data, including userID, bookID, ratings.The arrows connecting the components represent the dependencies and interactions between them.

3.2.3 Deployment Diagram

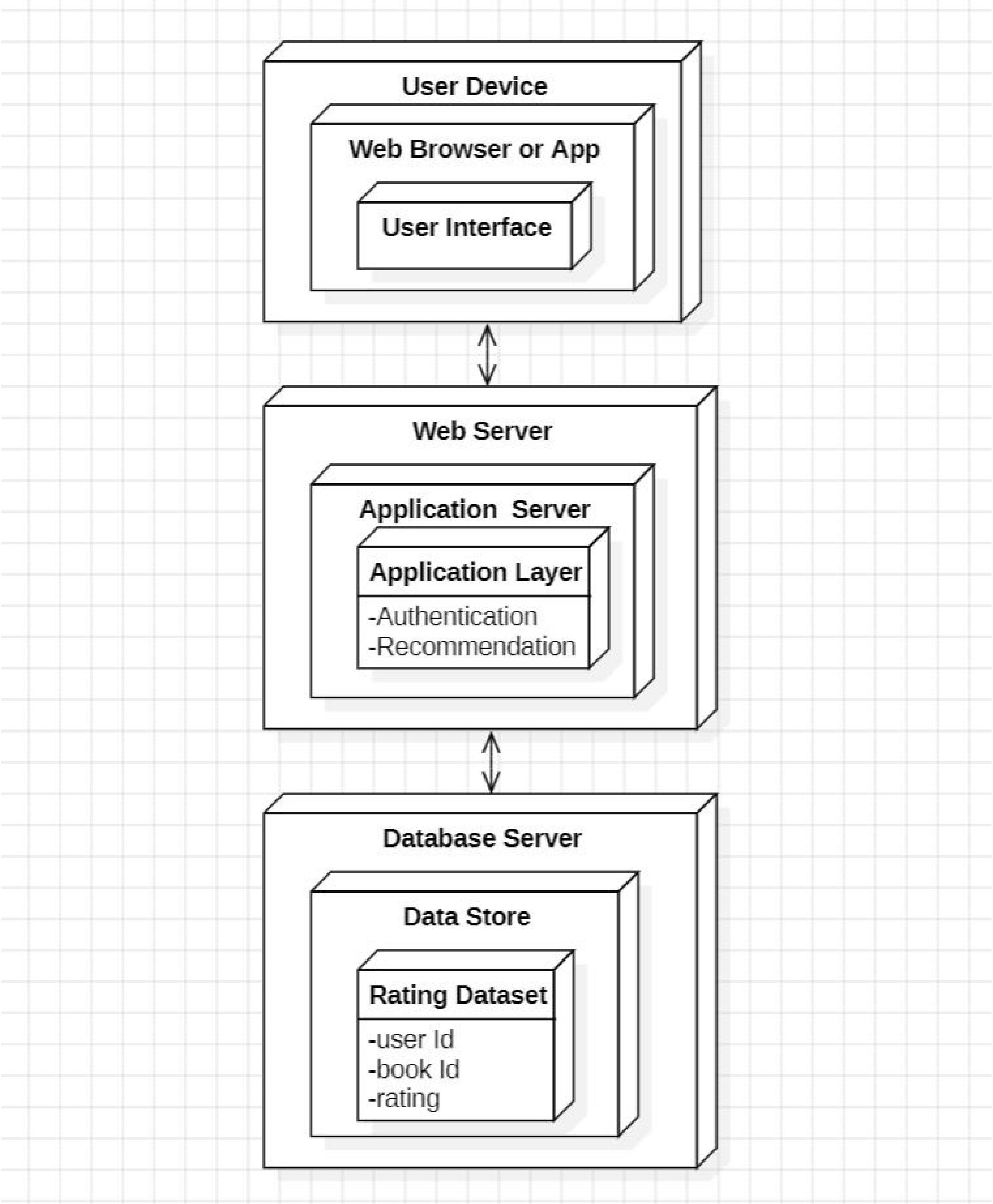


Figure 8: Chapter 3.2.3 Deployment Diagram

In the above figure the User Device represent the front-end interface through which the users interact with the system. The Web Server hosts the application logic, including the authentication and recommendation controllers. The Database Server stores and manages the systems data, including information related to books, ratings, recommendations. This deployment diagram shows how the component of a book recommendation system are distributed and interact across different hardware and software environment.

3.3 Algorithm Details

Collaborative Filtering using k-Nearest Neighbors (kNN) is a straightforward and intuitive approach for building recommendation systems. Below are the algorithmic details of a collaborative filtering book recommendation system using kNN:

1. Data Representation: The first step is to represent user-item interactions in a matrix form. Each row of the matrix corresponds to a user, each column corresponds to a book, and the entries represent the ratings given by users to books. If a user has not rated a book, the entry is typically left blank or filled with a default value.

2. Similarity Calculation: The similarity between users (or items) is calculated using a similarity metric such as cosine similarity, Pearson correlation, or Jaccard similarity. Cosine similarity is a common choice for its simplicity and effectiveness. It measures the cosine of the angle between two vectors representing the ratings given by users to books.

3. Neighborhood Selection: For each user (or item), the k most similar users (or items) are selected based on their similarity scores. This creates a neighborhood of users (or items) who have similar preferences.

4. Prediction Generation: To predict the rating of an item for a user, a weighted average of the ratings given by the k nearest neighbors is computed. The weights are typically determined by the similarity scores between the user and each neighbor Similarly, for item-based collaborative filtering, the prediction is based on the ratings of similar items.

5. Recommendation Generation: Once the predictions are generated for all items not yet rated by the user, the top-N items with the highest predicted ratings are recommended to the user. These recommended items represent books that the user is likely to enjoy based on the preferences of similar users.

6. Evaluation: The performance of the recommendation system is evaluated using standard evaluation metrics such as precision, recall, mean absolute error, or root mean squared error. These metrics measure how accurately the system predicts ratings or recommends items compared to the actual ratings or preferences of users. Parameter

7. Tuning: The value of k in kNN, which represents the number of neighbors to consider, is a hyperparameter that needs to be tuned for optimal performance. This can be done using techniques such as cross-validation.

Overall, collaborative filtering with kNN is a simple yet effective approach for building recommendation systems, particularly suitable for scenarios where explicit user-item ratings are available. However, it may suffer from scalability issues when dealing with large datasets due to the computational cost of calculating similarities between users or items. [2]

Chapter 4: Implementation and Testing

4.1 Implementation

4.1.1 Tools Used (Case tools, Programming languages, Database platforms)

CASE Tools

Table 3: Chapter 4.1.1 CASE tools

Used for	Tools and technologies
Analysis Tools	StarUML
Design Tools	StarUML balsamiq X
Programming Tools	python jupyter PC
Prototyping Tools	balsamiq
Maintenance Tools	PC
Diagram Tools	StarUML X
Process Modeling Tools	StarUML X
Project Management Tools	X
Documentation Tools	WPS Office

Configuration Management Tools	WPS Office
Web Development Tools	Google Chrome Microsoft Edge PC MySQL Workbench
Quality Assurance Tools	WPS Office

StarUML

StarUML is a modular and open tool which provides frameworks for extending the functionality of the tools. In my project, I have used StarUML to design the figures like flowchart, Use-Case Diagram, DFD etc.

Balasamiq cloud

Balasamiq Cloud is a web-based interface design tool mostly used for creating wireframes. I have designed my project UI using this web for prototype.

Microsoft Excel

Excel is a spreadsheet program. I have used Excel to make gantt chart which shows the starting ending and working time period of the system.

WPS Office

WPS Office is a free office software which involve Writes, Spreadsheet, Presentation and PDF. With the help of this software I have made my documentation of my project.

Google Chrome

I have used google chrome for searching various information related to my project. I have used it for downloading the python, pycharm, jupyter notebook etc. With the help of google chrome I was able to search and gather the information related to my project.

Microsoft Edge

Microsoft edge has got the co-pilot features for free. So, I used it as code fixers and QA (Quality Assurance) of my project code. Whenever I face error in code. I simply copy and paste it in co-pilot to find out the main errors.

YouTube

YouTube is a video sharing social media platform. While doing my project YouTube has been very useful platform for me through this I have fixed my code and get the knowledge about the algorithm also.

Table 4: Chapter: 4.1.1 Programming tools

Used for	Tools and technologies
Programming languages	python jupyter PC

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. By using this language I have developed by project.

Jupyter Notebook

The Jupyter Notebook is an open source web application. In this I have done my code of the book dataset. First I have arranged all the dataset and then used algorithm in this system.

Pycharm

PyCharm is a dedicated Python Integrated Development Environment (IDE). In this I have done all my coding section and developed my project.

4.1.2 Implementation Details of Modules (Description of procedures/functions)

In a collaborative book recommendation system, the user module plays a crucial role in managing user interactions and preferences. The user module typically includes features for user like login,search, ratings, preference collection, etc.

Login

Login features allow the user to enter into the system if the user is already registered in the system.

Register

Register feature allow the user to register the new user by their name along with email. Registered user can enter into the system which allows user to be part of the system.

Search

In search feature the user can search the books of their interest and get the recommendation through the system.

User Ratings

In this feature the user will rate the book they have read and rate it according to their interest.

4.2 Testing

4.2.1 Test cases for Unit Testing

Table 5: Chapter 4.2.1 Unit testing

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
1.	Home Button	On click Home Button	Shows most rated books	Showing the most rated books	Pass
2.	Search Bar	On click search Bar	Able to write	Able to write	Pass
3.	Search Button	On Click search Button	Shows the recommendation books	Showing the recommendatio n books	Pass
4.	Submit	On click submit	Display data	Display data	Pass

	Button	Button	from the datasets	from the datasets	
5.	Add Button	On click Add Button	Insert data in database	Insert data in database	Pass
6.	Update Button	On click Update Button	Update data in database	Update data in database	Pass
7.	Delete Button	On click Delete Button	Delete data from Database	Delete data from Database	Pass

4.2.2 Test cases for System Testing

Table 6: Chapter 4.2.2 System testing

S.No	Action	Inputs	Expected Output	Actual Output	Test Result
1.	Launch BRS	127.0.0.1	BRS system	BRS system	Pass
2.	Login(valid email & password)	susan@gmail.com Password: ****	Login successfully	Login success	Pass
3.	Login(invalid email & valid password)	susn@gmail.com Password: ****	Incorrect email & password	Incorrect email & password	Pass
4.	Login(valid email & invalid password)	susan@gmail.com Password: ******	Incorrect password & email	Incorrect password & email	Pass
5.	Login(invalid email & invalid password)	susn@gmail.com Password: ******	Incorrect email & password	Incorrect email & password	Pass
6.	Popular book datasets	Store most rated books	Shows most rated books	Shows most rated books	Pass
7.	Recommend book	Book name:"1984"	Recommend the book	Recommend the book	Pass

4.3 Result Analysis

The collaborative book recommendation system works by analyzing user ratings to suggest books that people might enjoy. By comparing the ratings from different users, it finds patterns and similarities to make personalized recommendations. The system's effectiveness can be seen in how well users respond to these suggestions—if they rate recommended books highly and continue using the system, it's a good sign that the recommendations are relevant and helpful. Overall, this system helps users discover new books they are likely to enjoy, making their reading experience more enjoyable and engaging.

Chapter 5: Conclusion and Future Recommendations

5.1 Lesson Learn / Outcome

Book Recommendation System is my project. In this project I have faced many problems and also learned many things. At first I didn’t have knowledge and idea to develop the website. Even I was confused while selecting the topic to develop the website. With the help of my teacher guidance I got the idea about the topic and chose Book Recommendation System. Then I started searching about the recommendation system and finalized collaborative filtering recommendation system as my project. While doing documentation and coding my teacher helped a lot in the errors and problem. From YouTube also I got many ideas of coding and with the help of internet I completed the project. This project helped me a lot in gaining the knowledge and experience and to develop executable system. As every project is an opportunity to gain new experience. Through this project I have gained many new experience related to IT and was capable of coding on my own and also learnt how to deal with the errors. Doing coding on my own have helped me to be confident and to gained experience.

I develop this system with many features which can be useful in books field and can bring well systematic change. So it is the first step for me to learn that I’m able to develop the demo type websites.

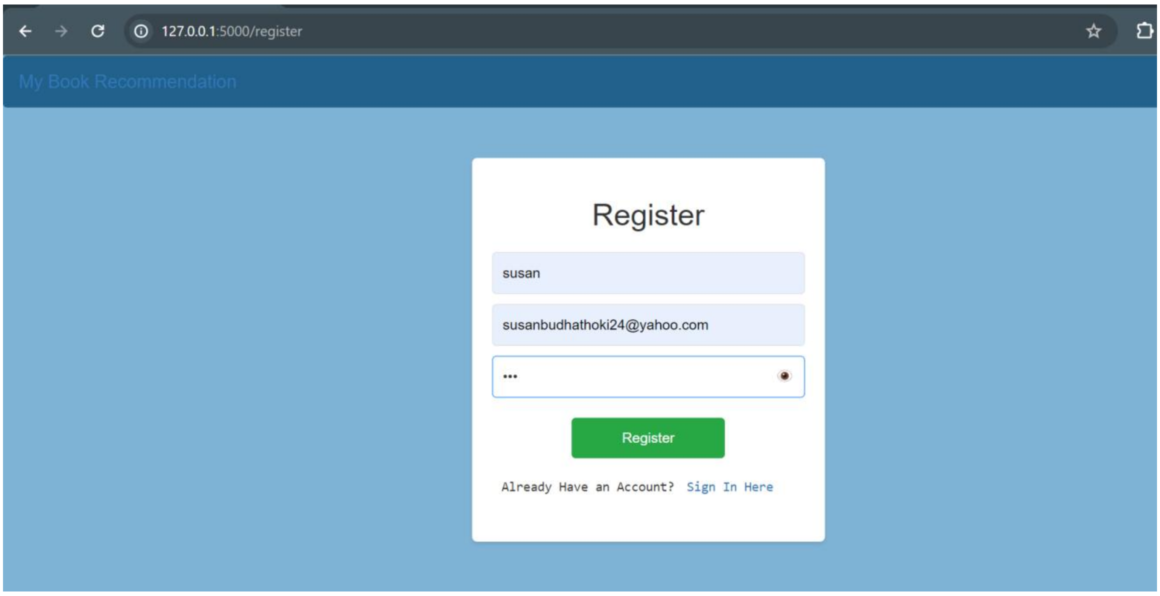
5.2 Conclusion

Finally, I have completed my project meeting all the guidance and requirements of TU. My project is book recommendation system which have features like it recommending books you might like. I have used a special method called collaborative filtering to recommend books to the user according to the rating similarities. The system worked well, and I found out that it was accurate in suggesting books. But, I also saw a few things we can make better, like implementing hybrid models, exploring matrix factorization techniques, Real-time recommendation system, continuous user feedback loop etc.to make the system even better.

5.3 Future Recommendations

In my project, I have made Book Recommendation System with the features like login and register to the user. In this the user can see the top average rated books and search for the books from the data-set. Now my project is collaborative filtering where the recommendation is based on the users rating only but in future the recommendation system will be the mixture of context based filtering and hybrid based recommendation. It will be more accurate in the future and it will recommend books to the user based on different algorithms.

Appendices



127.0.0.1:5000/register

My Book Recommendation

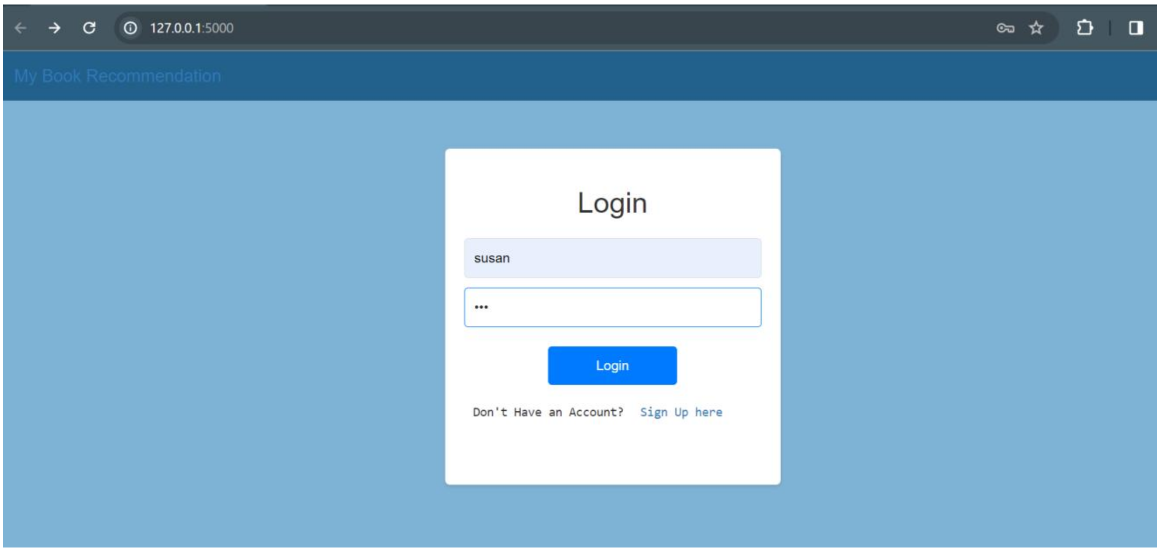
Register

susan

susanbudhathoki24@yahoo.com

•••

Already Have an Account? Sign In Here



127.0.0.1:5000

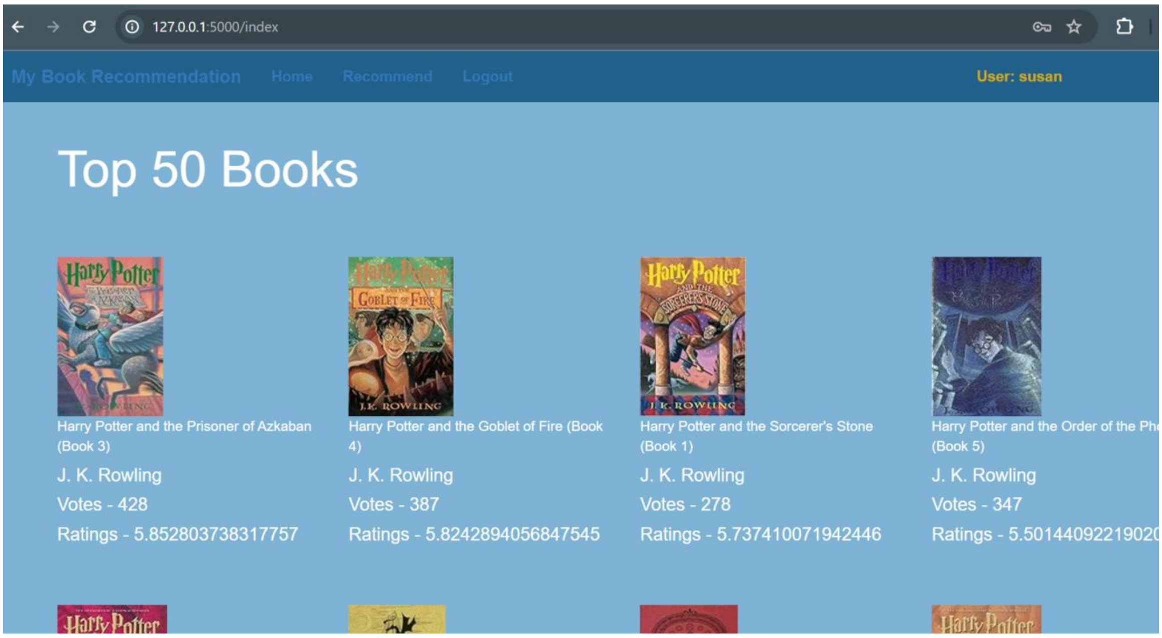
My Book Recommendation

Login

susan

•••

Don't Have an Account? Sign Up here



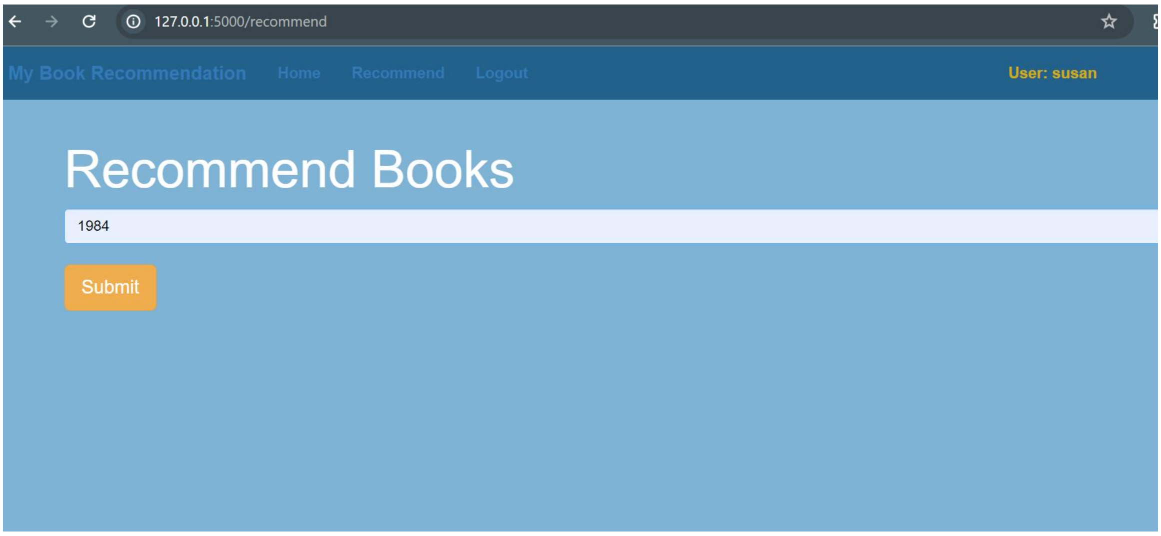
Top 50 Books

Harry Potter and the Prisoner of Azkaban (Book 3)
J. K. Rowling
Votes - 428
Ratings - 5.852803738317757

Harry Potter and the Goblet of Fire (Book 4)
J. K. Rowling
Votes - 387
Ratings - 5.8242894056847545

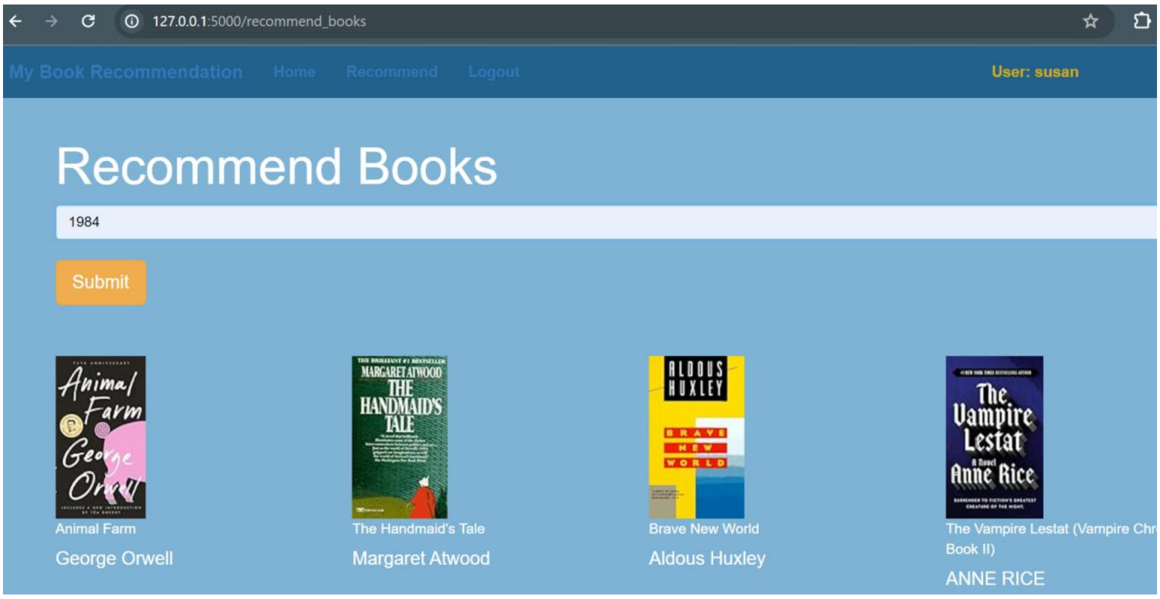
Harry Potter and the Sorcerer's Stone (Book 1)
J. K. Rowling
Votes - 278
Ratings - 5.737410071942446

Harry Potter and the Order of the Ph
(Book 5)
J. K. Rowling
Votes - 347
Ratings - 5.50144092219020



Recommend Books

1984



Recommend Books

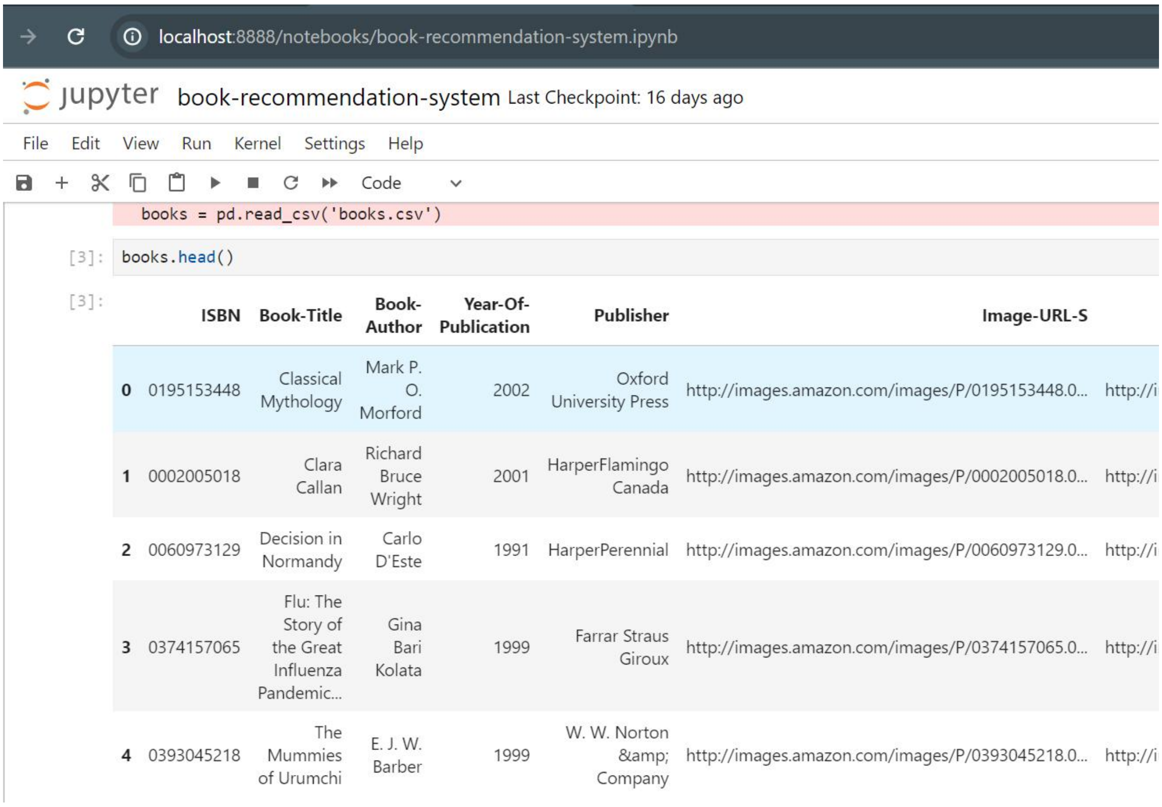
1984

Animal Farm
George Orwell

The Handmaid's Tale
Margaret Atwood

Brave New World
Aldous Huxley

The Vampire Lestat (Vampire Chr
Book II)
ANNE RICE

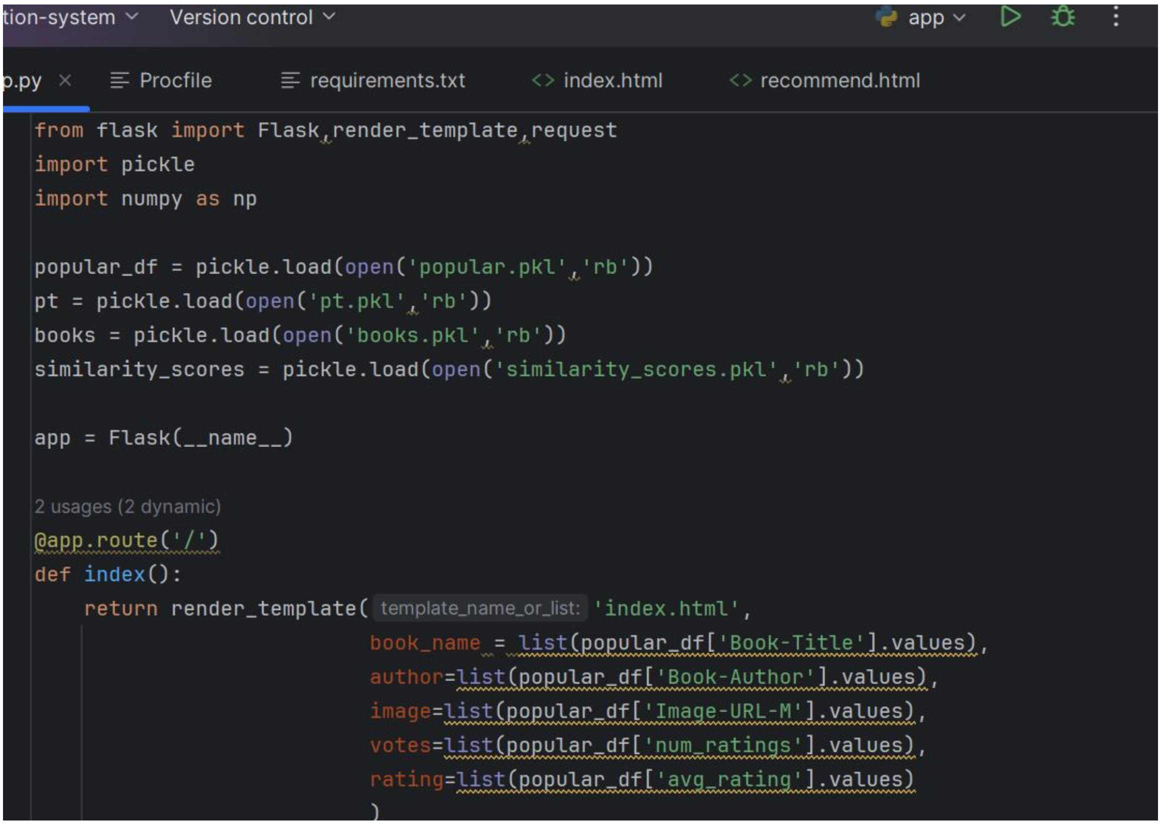

localhost:8888/notebooks/book-recommendation-system.ipynb
jupyter
book-recommendation-system Last Checkpoint: 16 days ago
File
Edit
View
Run
Kernel
Settings
Help
Code

```
books = pd.read_csv('books.csv')
```

```
[3]: books.head()
```

[3]:

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://i
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://i
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://i
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://i
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company	http://images.amazon.com/images/P/0393045218.0...	http://i

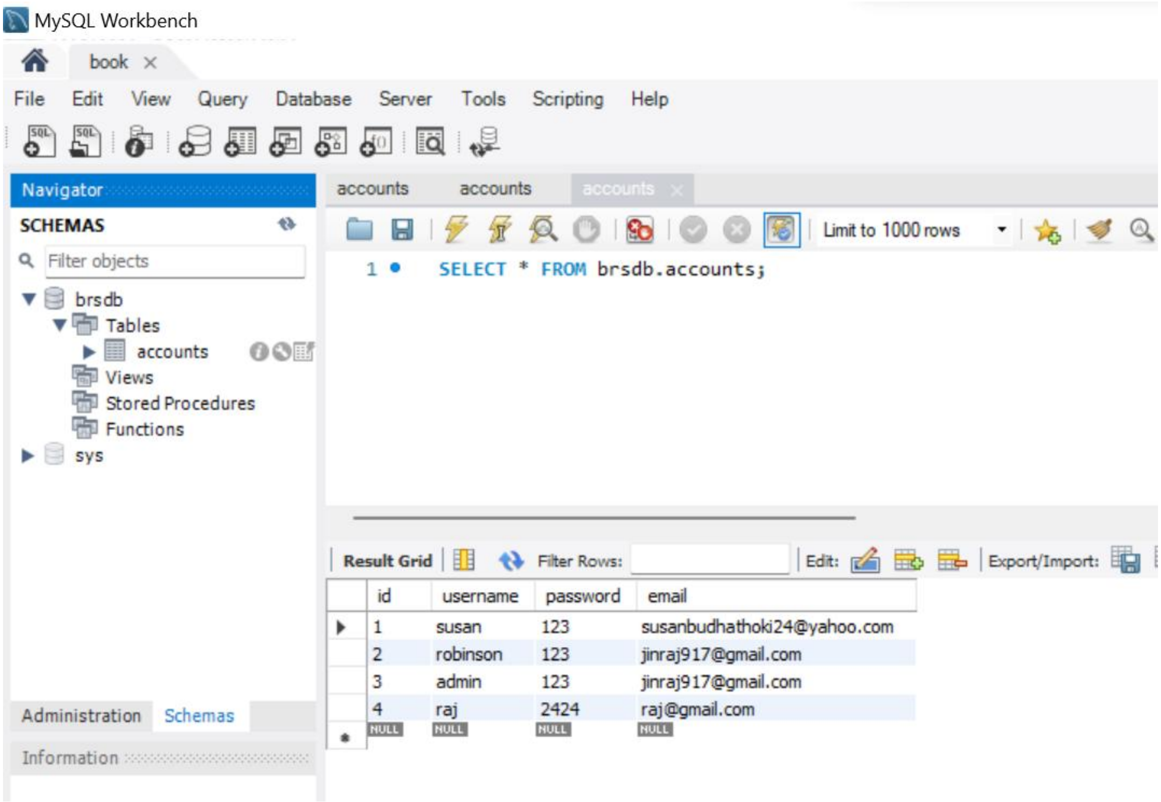


```
from flask import Flask,render_template,request
import pickle
import numpy as np

popular_df = pickle.load(open('popular.pkl','rb'))
pt = pickle.load(open('pt.pkl','rb'))
books = pickle.load(open('books.pkl','rb'))
similarity_scores = pickle.load(open('similarity_scores.pkl','rb'))

app = Flask(__name__)

2 usages (2 dynamic)
@app.route('/')
def index():
    return render_template( template_name_or_list: 'index.html',
                           book_name = list(popular_df['Book-Title'].values),
                           author=list(popular_df['Book-Author'].values),
                           image=list(popular_df['Image-URL-M'].values),
                           votes=list(popular_df['num_ratings'].values),
                           rating=list(popular_df['avg_rating'].values)
                           )
```



```
SELECT * FROM brsdb.accounts;
```

id	username	password	email
1	susan	123	susanbudhathoki24@yahoo.com
2	robinson	123	jinraj917@gmail.com
3	admin	123	jinraj917@gmail.com
4	raj	2424	raj@gmail.com
NULL	NULL	NULL	NULL

Reference

[1] "History of recommender systems," Onespire - SAP and IT services, 05-Jan-2021. .

[2] E. Ahmed and A. Letta, "Book recommendation using collaborative filtering algorithm," Appl. Comput. Intell. Soft Comput., vol. 2023, pp. 1–12, 2023.

[3] Researchgate.net. [Online]. Available: https://www.researchgate.net/publication/311610181_Book_Recommendation_System_through_content_based_and_collaborative_filtering_method. [Accessed: 11-Mar-2024].

[4] Ijcrt.org. [Online]. Available: https://ijcrt.org/papers/IJCRT2106360.pdf. [Accessed: 11-Mar-2024].

[5] Researchgate.net. [Online]. Available: https://www.researchgate.net/publication/323132248_Hybrid_attribute_and_personality_based_recommender_system_for_book_recommendation. [Accessed: 11-Mar-2024].

[6] Researchgate.net. [Online]. Available: https://www.researchgate.net/publication/271462577_Book_recommendation_system_based_on_combine_features_of_content_based_filtering_collaborative_filtering_and_association_rule_mining. [Accessed: 11-Mar-2024].

Supervisor Visit Log Sheets

Log for the Time management system	
Student Name	Susan Budhathoki
Email Id	budhathokisusan30@gmail.com
University Name	Tribhuvan University
College Name	Nims College
Supervisor Name	Nirmala Timilsena
Course & Code	Project-III (CACS452)

Meeting, Notes & Background Research		
Date	**Tasks**	**Signature**
2024/5/20	Submission of one-page proposal with the title of the project.	
2024/5/25	Submission of proposal with objectives, problem and solution.	
2024/6/20	Learned to use StarUML for object,dynamic and process modelling.	
2024/7/16	Conducted research and literature review related to the topic and learned citation.	
2024/8/10	Began coding Book Recommendation System.	
2024/9/18	Implemented the K-Nearest Neighbors(KNN) algorithm for generating recommendation.	
2024/10/09	Tested and refined the recommendation algorithm with sample datasets.	
2024/10/26	Presented project functionality to the supervisor and discussed errors.	
2024/11/14	Defence of Project by Supervisor.	
2024/11/24	Defence of Project by External Sir.	
2024/11/26	Improved documentation and finalized the project report based on feedback.	