­­­­ ­­­

**TRIBHUVAN UNIVERSITY**

**Asian School of Management and Technology**

**Gongabu, Kathmandu, Nepal**

**A Project Report**

**On**

**“Book Recommendation System”**

Submitted By:

Rabin Neupane (24109/076)

Ritesh RC (24114/076)­­­­

Sushil Shrestha (24127/076)

A Project Report Submitted in partial fulfillment of the requirement of **Bachelor of**

**Science in Computer Science & Information Technology (BSc. CSIT) 7th Semester** of Tribhuvan University, Nepal

Chaitra,2080

# BOOK RECOMMENDATION SYSTEM

**[CSC 412]**

A project report submitted for the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer science & Information Technology awarded by Tribhuvan University.

**Submitted By**

Rabin Neupane (24109/076)

Ritesh RC (24114/076)­­­­

Sushil Shrestha (24127/076)

**Submitted To**

Asian School of Management and Technology

Department of Computer science

Affiliated to Tribhuvan University

Gongabu, Kathmandu



Chaitra,2080

Date:7th Chaitra,2080

# SUPERVISOR’S RECOMMENDATION

It is my pleasure to recommend that a report on “**Book Recommendation System**” has been prepared under my supervision by **Rabin Neupane, Ritesh RC and Sushil Shrestha”** in partial fulfillment of the requirement of the degree of Bachelor of Science in Computer Science and Information Technology (BSc. CSIT). Their report is satisfactory to process for the future evaluation.

…………………………………

Er. Chakra Narayan Rawal

Supervisor

Department of Computer Science & IT

Asian School of Management and Technology

Date:20th Falgun,2080

# CERTIFICATE OF APPROVAL

The undersigned certify that he has read and recommended to the Department of Computer Science and Information Technology for acceptance of report entitled “**Book Recommendation System**” submitted by **Rabin Neupane, Ritesh RC and Sushil Shrestha”** in partial fulfillment for the degree of Bachelor of Science in Computer Science and Information Technology (BSc. CSIT), Institute of Science and Technology, Tribhuvan University.

………………………………. ……………………………

**Mr. Anil Lal Amatya Er. Chakra Narayan Rawal**

**Principal Program Co-ordinater**

**……………………… ……………………………**

**Er. Chakra Narayan Rawal External Examiner**

**Supervisor**

# ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our project supervisor, Er**. Chakra Narayan Rawal**, in whose contribution stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of Asian School of Management and Technology, who gave the permission to use all required equipment and the necessary materials to complete the task. We are thankful and fortunate enough to get constant support from our seniors and every teaching staff of B.Sc. CSIT department which helped us successfully complete our project. We would also like to extend our regards to all the non-teaching staff of CSIT department for their timely support. We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advice. Our appreciations also go to each and every one of our colleagues for their encouragement and support in developing the project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere appreciation to all of them.

With respect,

Rabin Neupane (24109/076)

Ritesh RC (24114/076)

Sushil Shrestha (24127/076)

# 

# ABSTRACT

The problem with traditional Book Recommender System is that they are often not personalized enough to meet the unique preferences and tastes of individual users. Here we have used an approach to a book recommendation system using Cosine Similarity and using K-NN to recommend similar books based on the one chosen by the user. The books are converted into vectors using Collaborative Filtering and both Cosine Similarity and K-NN are used to find the nearest vectors(books) and recommend it to the user. The analysis done on the recommendation system has accuracy of 68% in both Cosine Similarity and K-NN. Additionally, a frontend interface is developed using HTML, CSS and Bootstrap while the backend using Python, powered by Flask. Hence the recommended system using collaborative filtering provides personalized recommendation to the users.

**Keywords:** *Cosine similarity, Book recommendation, K-NN*

# TABLE OF CONTENT

[BOOK RECOMMENDATION SYSTEM ii](#_Toc161764602)

[SUPERVISOR’S RECOMMENDATION iii](#_Toc161764603)

[CERTIFICATE OF APPROVAL iv](#_Toc161764604)

[ACKNOWLEDGEMENT v](#_Toc161764605)

[ABSTRACT vi](#_Toc161764606)

[TABLE OF CONTENT vii](#_Toc161764607)

[LIST OF ABBREVIATIONS ix](#_Toc161764608)

[LIST OF FIGURES x](#_Toc161764609)

[LIST OF TABLES xii](#_Toc161764610)

[CHAPTER 1: INTRODUCTION 1](#_Toc161764611)

[1.1 Introduction 1](#_Toc161764612)

[1.2 Problem Statement 1](#_Toc161764613)

[1.3 Objectives 2](#_Toc161764614)

[1.4 Scope and Limitation 2](#_Toc161764615)

[1.4.1 Scope 2](#_Toc161764616)

[1.4.2 Limitations 3](#_Toc161764617)

[1.5. Development Methodology 3](#_Toc161764618)

[1.6 Report Organization 4](#_Toc161764619)

[CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW 5](#_Toc161764620)

[2.1 Background Study 5](#_Toc161764621)

[2.2 Literature Review 6](#_Toc161764622)

[CHAPTER 3: SYSTEM ANALYSIS 7](#_Toc161764623)

[3.1 System Analysis 7](#_Toc161764624)

[3.1.1 Requirement Analysis 7](#_Toc161764625)

[3.1.1.1 Functional Requirement 7](#_Toc161764626)

[3.1.1.2 Non-Functional Requirement 8](#_Toc161764627)

[3.1.2 Feasibility Study 8](#_Toc161764628)

[3.1.2.1 Technical Feasibility 8](#_Toc161764629)

[3.1.2.3 Operational Feasibility 9](#_Toc161764630)

[3.1.2.3 Economic Feasibility 10](#_Toc161764631)

[3.1.2.4 Schedule Feasibility 10](#_Toc161764632)

[3.1.3 Analysis 11](#_Toc161764633)

[3.1.3.1 Class Diagram 11](#_Toc161764634)

[3.1.3.2 Sequence Diagram 12](#_Toc161764635)

[3.1.3.3 Activity Diagram 13](#_Toc161764636)

[CHAPTER 4: SYSTEM DESIGN 15](#_Toc161764637)

[4.1 Design 15](#_Toc161764638)

[4.2 Interface Design 16](#_Toc161764639)

[4.3 Process Design 16](#_Toc161764640)

[4.4 Dataset Description 17](#_Toc161764641)

[4.5 Algorithm Details 18](#_Toc161764642)

[4.5.1 Cosine Similarity 18](#_Toc161764643)

[4.5.2 K-Nearest Neighbor 19](#_Toc161764644)

[CHAPTER 5: IMPLEMENTATION AND TESTING 20](#_Toc161764645)

[5.1 Implementation Overview 20](#_Toc161764646)

[5.1.1 Programming Language Tools 20](#_Toc161764647)

[5.1.1.1 Front-end Tools 20](#_Toc161764648)

[5.1.1.2 Back-end Tools 20](#_Toc161764649)

[5.1.1.3 Drawing Tools 21](#_Toc161764650)

[5.1.2 Working Mechanism 21](#_Toc161764651)

[5.2 Testing 23](#_Toc161764652)

[5.2.1 Unit Testing 23](#_Toc161764653)

[5.2.2 System Testing 25](#_Toc161764663)

[5.3 Result Analysis 26](#_Toc161764664)

[5.3.1 Evaluating Accuracy 26](#_Toc161764665)

[CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION 30](#_Toc161764666)

[6.1 Conclusion 30](#_Toc161764667)

[6.2 Future Recommendations 30](#_Toc161764668)

[REFERENCES 31](#_Toc161764669)

[APPENDICES 32](#_Toc161764670)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
|  |  |
| API | Application Program Interface |
| CF | Collaborative Filtering |
| CSV | Comma Separated Value |
| CSS | Cascading Style Sheet |
| GB | Giga Byte |
| GHz | Giga Hertz |
| HTML | Hypertext Markup Language |
| IDE | Integrated Development Environment |
| ISBN | International Standard Book Number |
| K-NN | K-Nearest Neighbor |
| NumPy | Numeric Pythone |
| OOADM | Object Oriented Analysis and Design Methodology |
| OS | Operating System |
| RAM | Random Access Memory |
| SKLearn | Scikit-Learn |
| SQL | Structured Query Language |
| UML | Unified Modeling Language |
| VS | Visual Studio |
| WSGI | Web Server Gateway Interface |

# LIST OF FIGURES

Figure 1.1: Incremental Delivery ……………………………………………………….3

Figure 3.1: Use Case Diagram of Book Recommender ……………………………..….7

Figure 3.2: Gantt Chart ………...………………………………………………….…...11

Figure 3.3: Class Diagram of Book Recommender System …………………………...11

Figure 3.4: Sequence Diagram of Book Recommender System ……………………....12

Figure 3.5: Activity Diagram of Book Recommender System ………………………...13

Figure 4.1: System Design of Book Recommender System …………………………...15

Figure 4.2: Workflow Diagram ………………………………………………………...17

Figure 5.1: Accuracy of Recommendation System using Cosine Similarity …………..27

Figure 5.2: Accuracy of Recommendation System using K-NN …………..………….28

Figure 5.3: Train vs Test Graph of Cosine Similarity ………………………………….28

Figure 5.4: Train vs Test Graph of KNN …………...………………………………….29

Figure 5.3: Training vs Testing Accuracy………….. ………………………………….29

# LIST OF TABLES

Table 3.1: Schedule Table ………………………………………………………...…….10

Table 5.1: Test cases for Interface ……………………………………………………....24

Table 5.2: Test cases for Model Accuracy ………………………………………...…….24

Table 5.3: Test cases for Performance …………………………………………….…….25

Table 5.4: Test cases for Scrapping ………………………………………………….….26

# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

A recommendation system is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences, behavior and ratings, then use this information to improve their suggestions in the future.

Books are integral aspect of life, offering a diverse range of genres and serving various purposes, from entertainment to education. They encompass fiction, non-fiction, academic, and many other categories. Books can be distinguished by genres such as mystery, romance, fantasy, biography, and more. Additionally, attributes like publication year, language, author, and subject matter contribute to the differentiation of books. Book Recommendation Systems helps us to search our preferred books among all of these different types of books and hence reduce the trouble of spending a lot of time searching our favorable books. So, it requires that the books recommendation system should be very reliable and should provide us with the recommendation of books which are exactly same or most matched with our preferences.

A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Book Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, book recommendation systems also suffer with poor recommendation quality and scalability issues.

## 1.2 Problem **Statement**

In the vast landscape of literature, readers often find it challenging to discover books that suit their individual tastes and preferences. Traditional methods of book discovery may fall short in providing personalized recommendations, leading to a gap between readers and books that have the potential to capture the interest and engage the reader. To address this issue, the goal of the Book Recommendation System project is to design and implement an advanced recommendation system tailored to the unique reading habits and preferences of users.

## 1.3 Objectives

Book Recommendation System has the following objectives:

* To develop medium scale application.
* To provide recommendation on the basic of similarities calculated from book ratings given by users.
* To design responsive site.
* To perform search, filtering functionalities.

## 1.4 Scope and Limitation

### 1.4.1 Scope

This project has many scopes. They are:

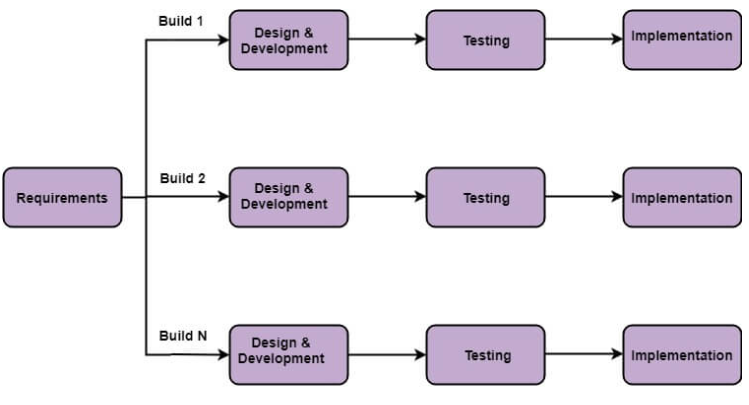
1. Personalization: A Book Recommender System should be capable of offering personalized recommendations tailored to each user's reading preferences. By analyzing user behavior, such as book ratings, reviews, and browsing history, the system should provide recommendations that align with the individual's interests and tastes.
2. Recommendation Engine: The Book Recommender System should incorporate a robust recommendation engine equipped with advanced algorithms and techniques. These algorithms should effectively analyze book attributes, user data, and interaction patterns to generate high-quality recommendations. Techniques such as collaborative filtering, content-based filtering, and hybrid approaches can be employed to enhance recommendation accuracy and relevance.
3. User Interface: The Book Recommender System should feature a user-friendly interface designed to facilitate seamless interaction and engagement. The interface should allow users to easily search for books, browse through recommendations, rate and review books, and provide feedback. Additionally, the interface should support personalized recommendations, enabling users to discover new titles that match their interests effortlessly.

### 1.4.2 Limitations

1. The system relies heavily on user behavior data. If users provide inaccurate data, the recommendations may not be accurate.
2. The system is limited by the availability and quality of data on books and user preferences. If there is insufficient data, the recommendations may not be relevant or useful.
3. The system may not be able to accurately recommend books for users with unique tastes that are not well-represented in the available data.

## 1.5. Development Methodology

The development methodology used for this project is incremental delivery, which is a kind of iterative software development process. The project is divided into a series of incremental iterations. Each function or features are being added upon the previous one to gradually deliver a full-fledged functional system. [1]



##### 

##### Figure 1.1: Incremental Delivery

Figure 1.1 explain a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations versions are implemented and delivered to the customer until the desired system is released.

## 1.6 Report Organization

The report on “**Book Recommendation System**” is based on six chapters. Each chapter follows the constructive building of this project. **Chapter 1** gives an overview idea of our project. It anticipates and combines the main points to be described later in the chapters followingly. Similarly, **Chapter 2** usually contains the theoretical literature review. It gives an insight to distinguish the possible hypothesis, strategies and shortfalls in the current research. **Chapter 3** studies the system such that information can be analyzed, modeled, and developed. It also gives enough information to replicate the study. It addresses the problems from chapter 1 and explains the objects of each experiment. **Chapter 4** contains an insight of the system design and algorithm being used while developing the system. **Chapter 5** contains system testing it discusses the execution of a program or system with the intent of finding errors. It also includes the examination of code as well as execution of that code in various environments and conditions. **Chapter 6** describes the significance of “Book Recommender System”, moreover, discusses the future recommendations applicable to enhance the project.

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## 2.1 Background Study

In recent years, the entertainment industry has been undergoing a significant shift towards digitalization, with an increasing number of consumers opting for online streaming services to read books. The vast selection of books available online can often lead to confusion and decision paralysis, with consumers struggling to choose books that aligns with their preferences. This problem can be addressed through the development of a Book Recommender system, which can provide personalized recommendations to consumers based on their viewing history, ratings and preferences.

Book Recommender systems have been gaining popularity in recent years due to their ability to provide tailored recommendations to users, thus improving the user experience. These systems utilize machine learning algorithms and data analysis techniques to analyze user behavior and preferences, and suggest books that are most likely to be of interest to the user. These algorithms consider various factors such as book genre, book authors, ratings, and user feedback to generate personalized recommendations.

The development of a Book Recommender system requires expertise in various areas such as machine learning, data analysis, and software development. The system will need to be able to collect and analyze user data in real-time and generate personalized recommendations efficiently. Additionally, the system will need to be scalable and capable of handling large amounts of data, as well as being user-friendly and easy to navigate.

Various criteria determine how the recommender systems work. The criteria are based on machine learning or deep learning algorithms that are used in matching the similarities before the suggestions are made. The algorithms achieve different levels of accuracy and require different computational times to retrieve the suggestions. Various computational algorithms have been proposed and used to increase the efficiency of recommender systems [2]

The benefits of a Book Recommender system are numerous, both for consumers and book streaming services. Consumers will be able to easily find books that align with their preferences, leading to a more satisfying reading experience.

In conclusion, the development of a Book Recommender system has the potential to revolutionize the entertainment industry by providing personalized recommendations to consumers, thus improving the overall user experience. With the increasing demand for book reading services, the development of such a system has become a necessity for book reading services to remain competitive in the market.

## 2.2 Literature Review

Okon et.al. proposed a model that generates recommendations to buyers, through an enhanced CF algorithm, a quick sort algorithm and Object-Oriented Analysis and Design Methodology (OOADM). Scalability was ensured through the implementation of Firebase SQL. This system performed well on the evaluation metrics [3].

Kurmashov et.al. used Pearson correlation coefficient-based CF to provide internet-based recommendations to book readers and evaluated the system through an online survey [4].

Parvatikar et.al. proposed item-based collaborative filtering and association rule mining to give recommendations. Similarity between different users was computed through the Adjusted Cosine Vector Similarity function. Better recommendations were obtained as through this method data sparsity problem was removed [5].

Feng et.al. proposed a RS for movies based on a similarity model constituted of factors S1 (similarity between users), S2 (ratio of co-rated items) and S3 (user’s rating choice weight). This RS was particularly useful for sparse datasets [6].

# CHAPTER 3: SYSTEM ANALYSIS

## 3.1 System Analysis

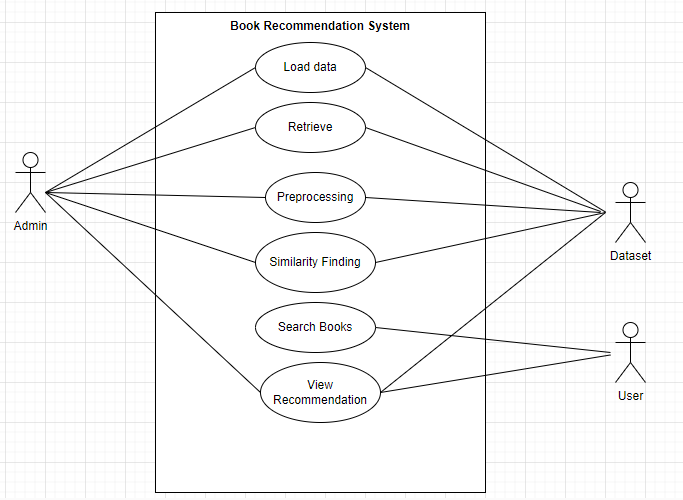
Systems analysis is a process of studying a system or organization in order to understand its components, how they interact and how they can be improved. It is a holistic approach that looks at the system as a whole and identifies the relationships between its parts. The goal of systems analysis is to identify problems and inefficiencies in the current system and to propose solutions for improvement.

### 3.1.1 Requirement Analysis

Requirement Analysis We’ve analyzed and validated the requirements, recorded and monitored the implementation throughout the project.

#### 3.1.1.1 Functional Requirement

1. Book dataset: The system should maintain a dataset of books along with their ratings and users with relevant metadata such as title, author, rating, and their respective images.
2. Book search: The system should allow users to search for books.
3. Personalized recommendations: The system should provide personalized recommendations to each user based on their preferences.



##### Figure 3.1 Use Case Diagram of Book Recommender

In Figure 3.1, whenever the user searches a certain book, the system retrieves the data of the book form the database and find the similarity with other books and recommend the books to the users.

#### 3.1.1.2 Non-Functional Requirement

The points below focus on the non-functional requirement of the system proposed.

* **User friendly**

User friendly generally means easy to read, use and communicate. The system is not complex and self-explanatory. Our system is well-organized, making it easy to locate different tools and options.

* **Reliability**

The system is reliable. The system takes data from various trusted sources and organization.

* **Scalability**

Scalability is crucial for handling growth in the user base and the amount of data. Our system will accommodate increasing numbers of users and growing datasets without compromising performance.

* **Easy access**

Our project is a web-based application. Considering this our platform can be accessed by anyone, anywhere where there is internet connection.

### 3.1.2 Feasibility Study

Feasibility studies aim to objectively and rationally uncover the strengths and weakness of an existing or proposed system, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for the success.

#### 3.1.2.1 Technical Feasibility

The technical feasibility of the project is high, as the required hardware and software resources are widely available and accessible. Requirements of our system can be categorized as:

**Hardware Requirements:**

* A computer with a minimum of 2 GB of RAM
* Processor with 1.7-2.4 GHz speed
* Minimum of 2GB Graphic card

**Software Requirements:**

* Operating System: Windows 10, Linux, or MacOS
* Text Editor (VS-code, PyCharm)
* Anaconda distribution package
* Python libraries
* **Anaconda distribution:** Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3
* **Python libraries:** For the computation and analysis certain python libraries which are used to perform analytics are required. Packages such as NumPy, pandas, Flask framework, etc. are needed.
* **NumPy:** NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.
* **Pandas:** Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.
* **Flask:** It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex application.

#### 3.1.2.3 Operational Feasibility

Operational feasibility measures how well a proposed system can solve the defined problem and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase. The system can be developed to be reliable, maintainable, usable, sustainable and affordable. So, this system is operationally feasible.

#### 3.1.2.3 Economic Feasibility

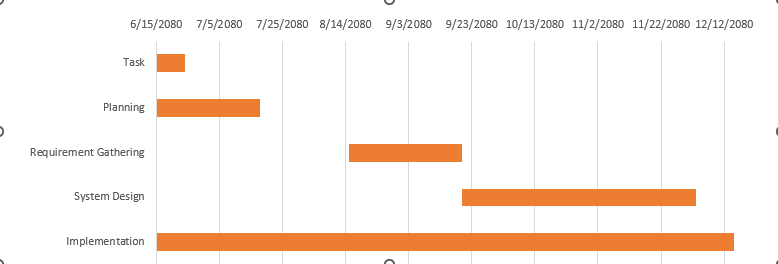
Economic feasibility analyses the project’s costs and revenue in an effort to determine whether it is possible to complete or not. There will not be any necessary equipment to be bought. However, the project will require domain, hosting and probably API which can be bought and configured with a suitable plan. Even if some features were to be added, it will be cost free as no extra equipment will be necessary. As the team already has everything needed, this system is economically feasible. Extensive databases can be maintained when the number of users of the app starts increasing.

#### 3.1.2.4 Schedule Feasibility

It is the most important for the completion of the project on time. The project is to be completed within time constraints. In figure 3.2, shows the time schedule of different activities of the process.

Table 3.1: Schedule Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Duration** |
| Planning | 6/15/2080 | 6/24/2080 | 9 |
| Requirement Gathering | 6/15/2080 | 7/18/2080 | 33 |
| System Design | 8/15/2080 | 9/20/2080 | 36 |
| Implementation | 9/20/2080 | 12/3/2080 | 74 |
| Documentation | 6/15/2080 | 12/15/2080 | 183 |



##### Figure 3.2: Gantt Chart

### 3.1.3 Analysis

#### 3.1.3.1 Class Diagram

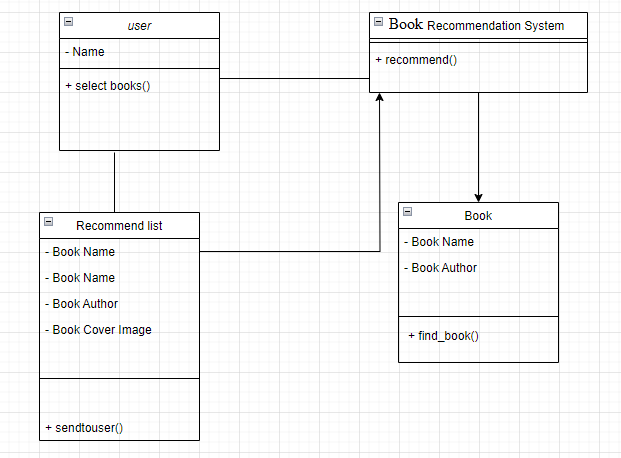
Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

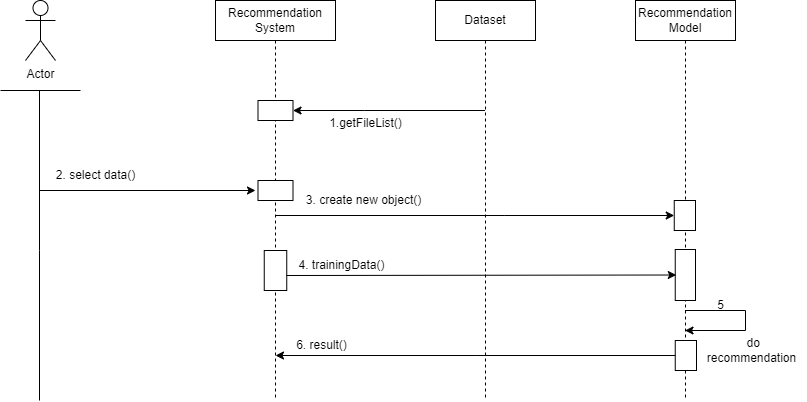
Figure 3.3: Class Diagram of Book Recommender System

In Figure 3.3 illustrate the blueprint of the system. It represents the types of objects residing in the system and the relationships between them.

#### 3.1.3.2 Sequence Diagram

Sequence diagrams can be useful reference diagrams for businesses and other organizations. Draw a sequence diagram to:

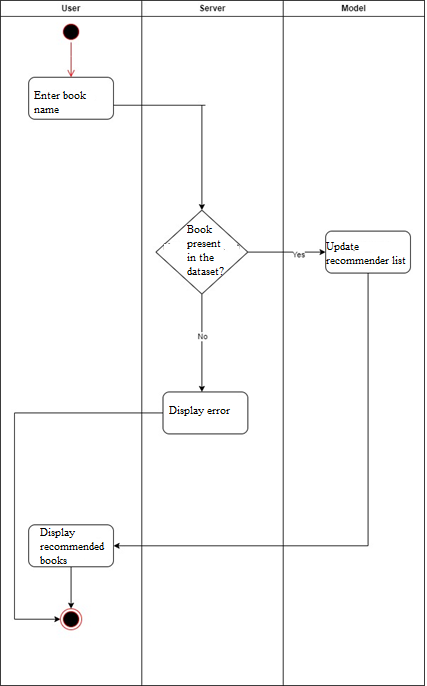
* Represent the details of a UML use case.
* Model the logic of a sophisticated procedure, function, or operation.
* See how tasks are moved between objects or components of a process.
* Plan and understand the detailed functionality of an existing or future scenario.



##### Figure 3.4: Sequence diagram of book recommender system

Figure 3.4, illustrate the interaction between user and the recommendation system. When the user selects the books, the system gets the recommended books from the database with the help of recommendation model. When the user selects a certain book in the system then the system creates a new object and transfer to the recommendation model. According to the selected book model recommend the book and provide result to the user. It shows the sequence diagram of the book recommendation system. The user enters the name of a book in the search bar and clicks the search button. The system validates the input and checks if the book is present in the dataset. If the book is found, the system retrieves the data of the book from the database and passes it to the recommendation model.

#### 3.1.3.3 Activity Diagram

******Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. The basic purposes of activity diagrams are to capture the dynamic behavior of the system.



##### 

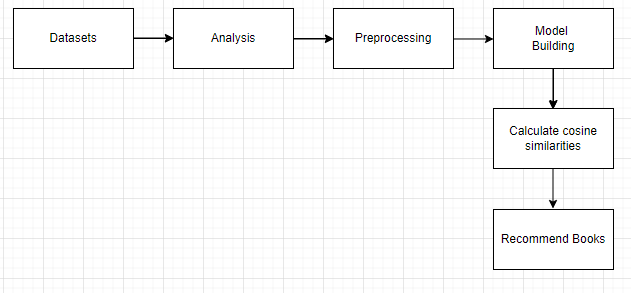
##### Figure 3.5: Activity Diagram of Book Recommender System

In Figure 3.5, the user searches a book in the search bar, and the server checks whether the book is present in the dataset or not. If the book is present in the dataset, then the server updates the recommendation list using the recommendation model and displays the recommended books to the user. However, if the book is not present in the dataset, then it will display an error.

# CHAPTER 4: SYSTEM DESIGN

## 4.1 Design

System design is the process of representing architecture, interfaces, components that are included in the system. i.e., system design can be seen as the application of system theory to product development.



##### Figure 4.1: System Design of Book Recommendation System

The system architecture is shown in the Figure 4.1, The procedures involved are as:

* **Datasets**: We start by collecting and curating various datasets like books, ratings and users containing information about various books, books users and rating of the book.
* **Analysis**: Next, we analyze the collected datasets to gain insights into the characteristics and trends within the data. This analysis helps us understand the distribution of books, identify popular authors, and uncover patterns.
* **Preprocessing**: Before building the recommendation model, we preprocess the data to clean and transform it into a suitable format. This involves handling missing values, standardizing text data, encoding categorical variables, and scaling numerical features.
* **Model** **Building**: With the preprocessed data, we proceed to build the recommendation model. Depending on the approach, we used collaborative filtering technique. The goal is to develop a model that can effectively suggest relevant books to users based on their preferences.
* **Calculate Cosine Similarity**: In collaborative filtering-based recommendation systems, we calculate the cosine similarity between books or users to measure their similarity. This similarity metric helps us identify books that are closely related to each other, allowing us to make personalized recommendations.
* **Recommend Books**: Finally, using the calculated cosine similarity scores, we generate recommendations for users. We suggest a list of books that are likely to interest them. This recommendation process forms the core functionality of our book recommender system, helping users discover new and relevant reading material.

## 4.2 Interface Design

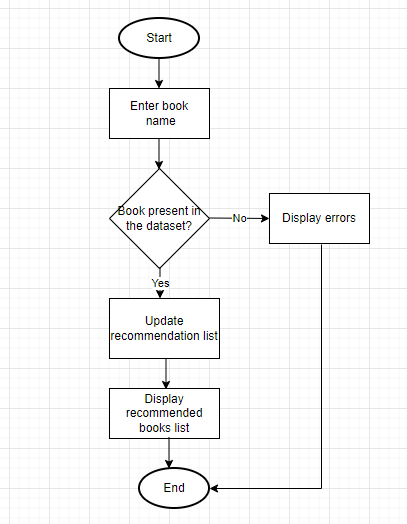
For the User Interface design of our book recommender system, we are using HTML, CSS, and Flask. HTML forms the backbone of our interface, defining the structure and layout of the pages to ensure clarity and organization.

To enhance the visual appeal and user experience, we use the power of CSS. By styling elements and adding design elements such as colors, fonts, and layouts, CSS transforms the interface into an attractive and user-friendly platform for discovering and exploring books.

Furthermore, Flask serves as the server-side framework, seamlessly integrating with our HTML and CSS to render dynamic content, handle user requests, and manage the overall functionality of the system. With Flask, we can create interactive features, process user input, and deliver personalized recommendations, enriching the user experience and making our book recommender system a valuable resource for book enthusiasts.

## 4.3 Process Design

The simple process of how our system works is shown through the flowchart given below:



##### Figure 4.2 Workflow Diagram

## 4.4 Dataset Description

The recommendation system utilizes three datasets entitled Books.csv, Ratings.csv, Users.csv. 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. Ratings.csv 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. Users.csv contains information about the users. The 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. [7]

## 4.5 Algorithm Details

**Collaborative Filtering**

Collaborative filtering is a recommendation technique that predicts user preferences by leveraging the behavior and preferences of similar users or items. It relies on user-item interaction data to recommend items to users based on the preferences of other users with similar tastes.

To address some of the limitations of content-based filtering, collaborative filtering uses similarities between users and items simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. Furthermore, the embeddings can be learned automatically, without relying on hand-engineering of features. [8]

### 4.5.1 Cosine Similarity

After the user is prompted to enter a book, the algorithm provides 10 other books like the one used as an input by the user.

In cosine similarity, vectors are taken as the data objects in data sets, when defined in a product space, the similarity is figured out. The smaller this distance, the higher the similarity, but the larger the distance, the lower the similarity. Cosine similarity is a measure that helps to find out how similar data objects are, regardless of size. Mathematically, it is the cosine of the angle between two vectors projected in a multi-dimensional space.

𝑎⃗. 𝑏⃗⃗ ∑𝑛1 𝑎𝑖𝑏𝑖

𝐶𝑜𝑠𝜃 = = ……………………(i)

√

∑

𝑎

𝑖

2

𝑛

1

√

∑

𝑏

𝑖

2

𝑛

1

‖𝑎⃗‖‖𝑏⃗⃗‖

Where*, 𝑎⃗. 𝑏⃗⃗ = ∑𝑛1 𝑎𝑖𝑏𝑖 = 𝑎𝑖𝑏𝑖 + 𝑎2𝑏2 + ⋯ + 𝑎𝑛𝑏𝑛* is the dot product of the two vectors.[9]

The angle between two vectors determines its direction and is measured in ‘θ’. This angle can be calculated by using above equation (i).

When θ = 0 °, the `*x*` and` *y*` vectors overlap and prove to be similar.

When θ = 90 °, the `*x*` and` *y*` vectors are therefore dissimilar.

### 4.5.2 K-Nearest Neighbor

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems [10]. KNN is a supervised learning algorithm used for classification and regression tasks. It classifies a data point based on the majority class of its 'k' nearest neighbors. In regression, it predicts the value of a data point based on the average value of its 'k' nearest neighbors. Euclidean distance is commonly used to measure the distance between data points, although other distance metrics like Manhattan distance or Minkowski distance can also be used. The formula to calculate Euclidean distance between two points are:

Euclidean distance between two points = 

By calculating the Euclidean distance, we got the nearest neighbors [10] .

The steps involved are:

* We merged user ratings and book information tables to create a matrix with books as rows and users as columns, containing user ratings for each book.
* We mainly focused on users who have rated over 200 books and books with over 50 ratings to ensure engagement and popularity.
* This calculation of cosine similarities between users and the items and also the calculation of Euclidean distance using K-NN ensures reliability by ensuring recommendations on preferences of dedicated readers and well-reviewed books.
* By filtering data, we aim to provide more robust and meaningful suggestions to users, enhancing the overall recommendation system.

# CHAPTER 5: IMPLEMENTATION AND TESTING

## 5.1 Implementation Overview

The implementation model is a method for developing software products that involves designing, implementing, and testing the final product one by one. When the requirements are known before the product is developed and can be implemented in stages, the incremental development technique is used. For our project, the incremental approach includes gradually updating the model with updated data over time to increase its accuracy. In order to increase the system accuracy, refinement of algorithms is done. Over time, the incremental approach can provide better sentiment analysis increasing the accuracy of recommendation system.

### 5.1.1 Programming Language Tools

For the Book Recommender: Collaborative Filtering Book Recommendation System, the primary language used for the backend is Python, and for the frontend, HTML, CSS, Bootstrap, Flask are used for the web interface.

#### 5.1.1.1 Front-end Tools

* **HTML**

HTML is the most basic building block of the web. It defines the meaning and structure of web content. It was used to develop the web page.

* **CSS**

CSS is the style sheet language for describing the presentation and design of web pages including colors, fonts and layouts. It was used to enable the distinction between presentation and content, including colors, layout and fonts.

#### 5.1.1.2 Back-end Tools

* **Python libraries:** For the computation and analysis the system need certain python libraries which are used to perform analytics. Packages such as SKlearn, NumPy, pandas, Matplotlib, Flask framework, etc. are needed.

1. **SKlearn:** It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
2. **NumPy:** NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.
3. **Pandas:** Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

* **Flask:** It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

#### 5.1.1.3 Drawing Tools

* **Draw.io:**

It is a popular online diagramming tool used to create various types of diagrams. We used this tool to create following diagram:

Class Diagram

Activity Diagram

Sequence Diagram

Use Case Diagram

### 5.1.2 Working Mechanism

Popularity-based recommendation system is a non-personalized recommendation system that uses data from a large number of users to determine the most popular or trending items. In our proposed system, using popularity-based recommendation system, we will display the top 50 books with highest average ratings but will consider only those books which have got a minimum of 250 ratings. We only look at books that have received at least 250 ratings. This ensures we're focusing on books with a fair number of opinions. For each of these selected books, we add up all the ratings and divide by the number of ratings. This gives us the average rating for each book.

Average Rating = Sum of Ratings / Number of Ratings

We then arrange these books from highest to lowest based on their average ratings. The books with the best average ratings come first. Finally, we show the top 50 books from this sorted list. These are the books that not only have high average ratings but also have received a good number of ratings from users.

In the process of developing our recommendation system, we use user ratings from the rating table (user id, book id, rating) and information about books from the books table (book name). By merging these datasets, we construct a new table where rows represent books, columns represent users, and the grid entries contain the ratings provided by users for each book.

To enhance the reliability of our recommendations, we narrow down our focus. We only include users who have actively rated over 200 books, indicating a maximum engagement with the platform. Similarly, we filter out books that haven't received much attention, considering only those with over 50 ratings from various users.

In collaborative filtering, we use cosine similarity as a key metric to quantify the similarity between books and users. By computing the cosine similarity scores between items, we ascertain their likeness, facilitating the identification of books with similar characteristics or user preferences. This similarity measurement is fundamental in generating personalized recommendations tailored to individual user tastes.

The below formula is the formula for calculating cosine similarities.

𝑎⃗. 𝑏⃗⃗ ∑𝑛1 𝑎𝑖𝑏𝑖

𝐶𝑜𝑠𝜃 = = ……………………(i)

√

∑

𝑎

𝑖

2

𝑛

1

√

∑

𝑏

𝑖

2

𝑛

1

‖𝑎⃗‖‖𝑏⃗⃗‖

Where*, 𝑎⃗. 𝑏⃗⃗ = ∑𝑛1 𝑎𝑖𝑏𝑖 = 𝑎𝑖𝑏𝑖 + 𝑎2𝑏2 + ⋯ + 𝑎𝑛𝑏𝑛* is the dot product of the two vectors.[9]

This selective approach ensures that our recommendation system is built on the preferences of dedicated readers and popular, well-reviewed books, thereby offering a more robust and meaningful set of suggestions to users.

## 5.2 **Testing**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:

* To affirm the quality of the project.
* To find and eliminate any residual errors from previous stages.
* To validate the software as a solution to the original problem.
* To provide operational reliability of the system.

**Testing Methodologies**

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

### 5.2.1 Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. Test Left is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

Table 5.1: Test Cases for Interface

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. N** | Test  Cases | Feature | Test  Description | Steps To  Execute | Expected  Results | Remarks |
| 1. | TC-IV-1 | User Interface | Check all  the textboxes and buttons | Check  Page | • UI should be perfect  • Text boxes and button should be  aligned | Pass |
| 2. | TC-IV-2 | RequiredFields | Check therequired fields by not filling any data. | Enter invalid book name | Error message should        be shown | Pass |

Table 5.2: Test Cases for Model Accuracy

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. N** | **Test**  **Cases** | **Feature** | **Test**  **Description** | **Steps To**  **Execute** | **Expected**  **Results** | **Remarks** |
| 1 | TC-MA-  2 | Recommendation  testing | Tested on  test data of our dataset | Check accuracy | Good  Accuracy | Pass |

Table 5. 3: Test Cases for Performance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. N** | **Test**  **Cases** | **Feature** | **Test**  **Description** | **Steps   To**  **Execute** | **Expected**  **Results** | **Remarks** |
| **1.** | TC-P-1 | Waiting period. | System performs efficiently under normal and peak loads | Check loading time. | Recommends books in  minimal time | Pass |
| **2.** | TC-P-2 | Waiting period. | The system provides efficient recommendations under normal and peak loads. | Check response time for generating book recommendations. | Performs book recommendation with minimal latency. | Pass. |

### 5.2.2 System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

Table 5. 4: Test Cases for Web Scraping

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. N** | **Test**  **Cases** | **Feature** | **Test**  **Description** | **Steps    To**  **Execute** | **Expected**  **Results** | **Remarks** |
| 1. | TC-WS-  1 | Information Retrieval | Retrieve the relevant information about books  from    the dataset. | Access the dataset | Successfully extract the relevant and needed information from    the dataset. | Pass |
| 2. | TC-WS-  2 | User Ratings Extraction | Extract the user ratings  from    the dataset for book recommendations. | Access the dataset | Successfully retrieve user ratings for recommendation analysis | Pass |

## 5.3 Result Analysis

The system was tested through unit testing and proved to be effective in executing its intended functions. The results showed that the project was able to meet its goals, but there is still room for improvement in terms of expanding the system's capabilities and increasing community involvement.

### 5.3.1 Evaluating Accuracy

In machine learning, accuracy is a common metric used to evaluate the performance of a classifier model. Accuracy measures the proportion of correctly classified instances among all instances in the dataset. To calculate accuracy, the first step is to divide the dataset into two parts: a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the model's performance.

In classifier model the most common measure to evaluate accuracy are:

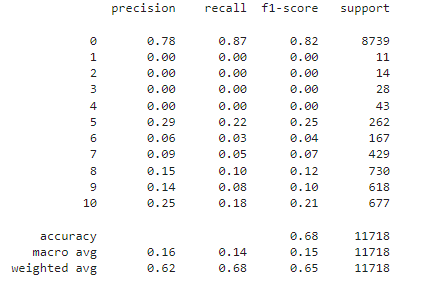
* Precision: Precision is the fraction of true positives among all the positive predictions made by the model. The equation (i) measure how accurate the model is, when predicting positive instances. The formula for precision is:

…………..(i)

* Recall: Recall is the fraction of true positives among all the actual positive instances in the dataset. The equation (ii) measure how well the model is able to identify positive instances. The formula for recall is:

*…………..…..(ii)*

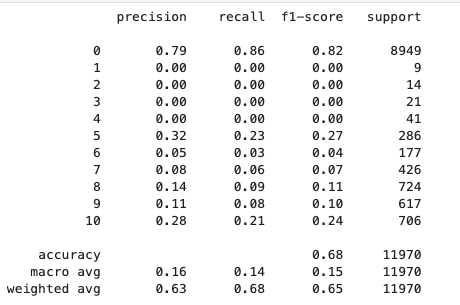
* F1 score: The F1 score is the harmonic mean of precision and recall. It provides a single score that balances the tradeoff between precision and recall. The F1 score ranges from 0 to 1, where a score of 1 represents perfect precision and recall, and 0 represents the worst performance.



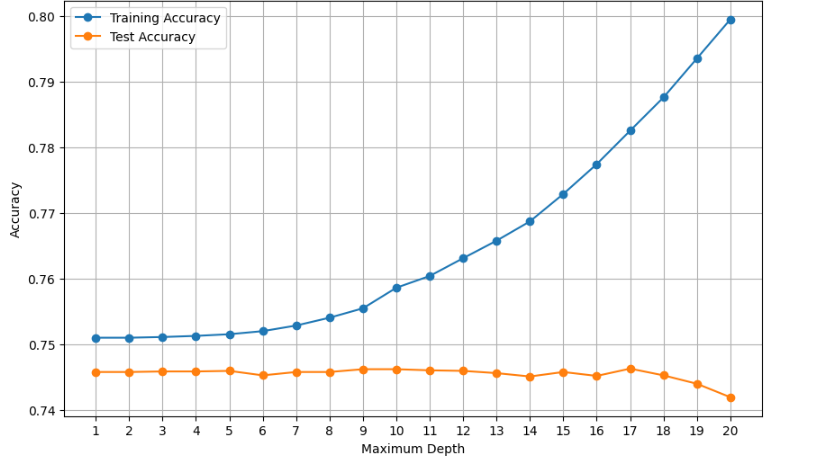
.

##### Figure 5.1 Accuracy of Recommendation System using Cosine Similarity

Figure 5.1 illustrates the accuracy of the system using cosine similarity. The accuracy of the system is 68%.

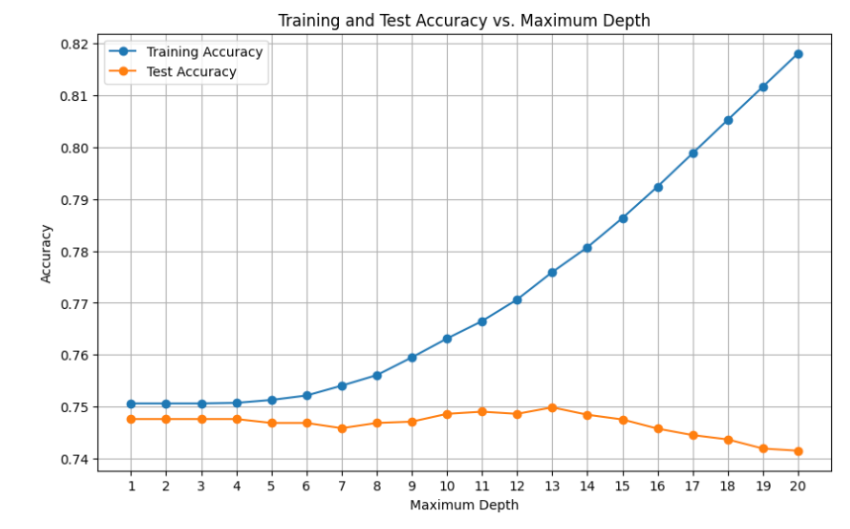


##### Figure 5.2 Accuracy of Recommendation System using K-Nearest Neighbor

Figure 5.1 illustrates the accuracy of the system using KNN. The accuracy of the system is 68%.

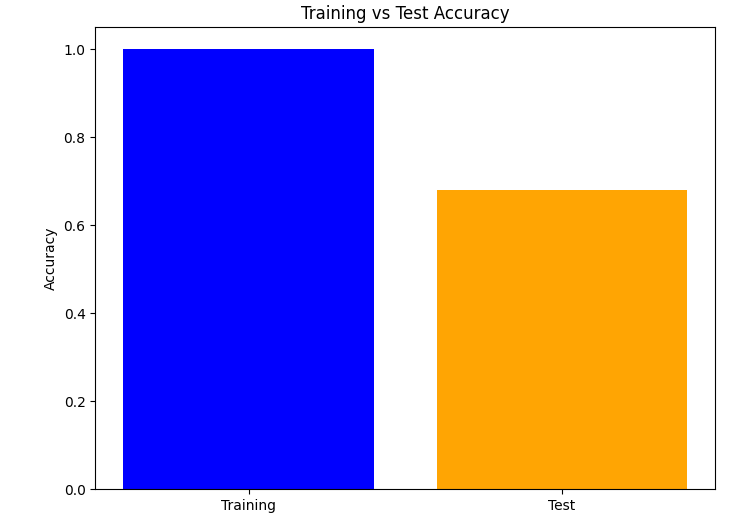
##### Figure 5.3 Train vs Test Graph for Cosine Similarity

Figure 5.2 illustrates the training accuracy and the testing accuracy graph. Here the lower line determines the performance of the testing accuracy and the upper line represents the training accuracy; while X-axis displays maximum depth while Y-axis displays the increasing accuracy.



##### Figure 5.4 Train vs Test Graph for KNN

Figure 5.4 illustrates the training accuracy and the testing accuracy graph. Here the lower line determines the performance of the testing accuracy and the upper line represents the training accuracy; while X-axis displays maximum depth while Y-axis displays the increasing accuracy.



##### Figure 5.5 Training vs Testing Accuracy

# CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATION

## 6.1 Conclusion

Thus, the recommender system was successfully implemented. Here we have used an approach to a book recommendation system using Cosine Similarity to recommend similar books based on the one chosen by the user. The analysis done on the recommendation generated shows 68% accuracy for both Cosine Similarity and K-Nearest Neighbor. Hence, we found that Collaborative Filtering with both cosine similarity and K-NN will the best in our situation. For working on large dataset, it was an approach in implementing the algorithms and making it a Recommender System.

It was a challenge to implement a recommender system on this scale of huge data. Recommender systems have become ubiquitous. People use them to find books, music, news, smart phones, vacation trips, and romantic partners. Nearly every product, service, or type of information has recommenders to help people select from among the various alternatives the few they would most appreciate. Sustaining these commercial applications is a vibrant research community, with creative interaction ideas, powerful new algorithms, and careful experiments.

## 6.2 Future Recommendations

In the future, we would like to improve the suggestion process by:

1. Add registration feature.
2. We can add user registration feature to store each users’ unique preferences and recommend them books accordingly.
3. Introduce user dislike book list. The user data is always useful in recommender systems. In the future we can collect more user data and add user dislike book list. We will input dislike book list into the recommender system as well and generate scores that will be added to previous result. By this way we can improve the result of recommender system.
4. Make the recommender system as an internal service. In the future, we can make it as an internal APIs for developers to invoke. Some book lists in the website will be sorted by recommendation.

# REFERENCES

[1] *Incremental Process Model - Software Engineering - Geeksforgeeks*, www.geeksforgeeks.org/software-engineering-incremental-process-model/. Accessed 18 Mar. 2024.

[2] H. H. M. H. Cami B.R., "A content-based movie recommender system based on temporal user preferences," Shahrood, Iran, 2017

[3] Okon, E.U., Eke, B.O. and Asagba, P.O. (2018). An improved online book recommender system using collaborative filtering algorithms. International Journal of Computer Applications (0975-8887) Volume 179-No.46, June 2018

[4] Kurmashov, N., Konstantin, L., Nussipbekov, A. (2015). Online book recommendation System. Proceedings of Twelve International Conference on Electronics Computer and Computation (ICECC)

[5] Arvind Nair, et al. “Enhancing book recommendations: A comprehensive exploration of cosine similarity in collaborative filtering for personalized reading experiences.” *International Journal For Multidisciplinary Research*, vol. 5, no. 6, 20 Dec. 2023, https://doi.org/10.36948/ijfmr.2023.v05i06.10825.

[6] Liu, Q., Chen, E., Xiong, H., Ding, C.H.Q. and Chen, J. (2012). Enhancing Collaborative Filtering by User Interest Expansion via Personalized Ranking. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 42, No.1, February 2012.

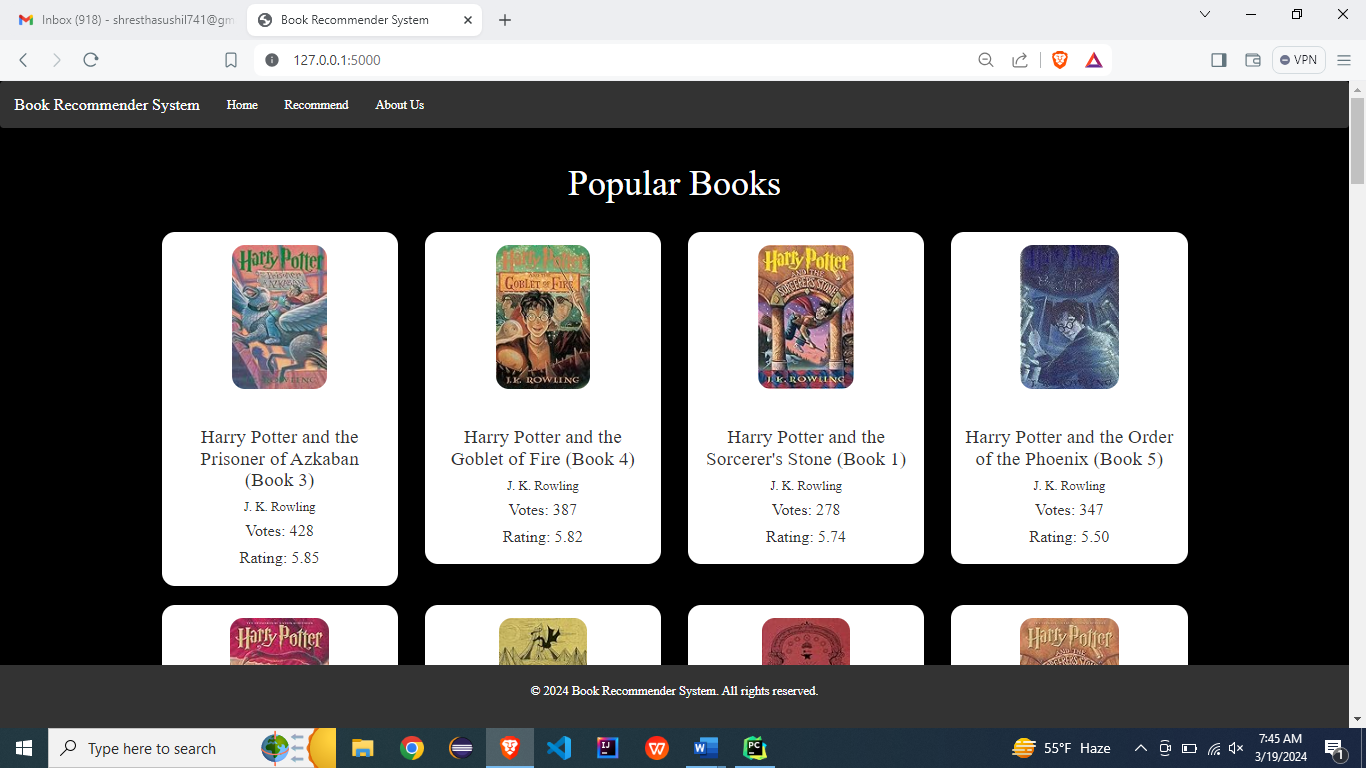
[7] M&ouml;bius. “Book Recommendation Dataset.” *Kaggle*, 9 Feb. 2024, www.kaggle.com/datasets/arashnic/book-recommendation-dataset?select=Users.csv.

[8] “Collaborative Filtering  |  Machine Learning  |  Google for Developers.” *Google*, Google, developers.google.com/machine-learning/recommendation/collaborative/basics. Accessed 19 Mar. 2024.

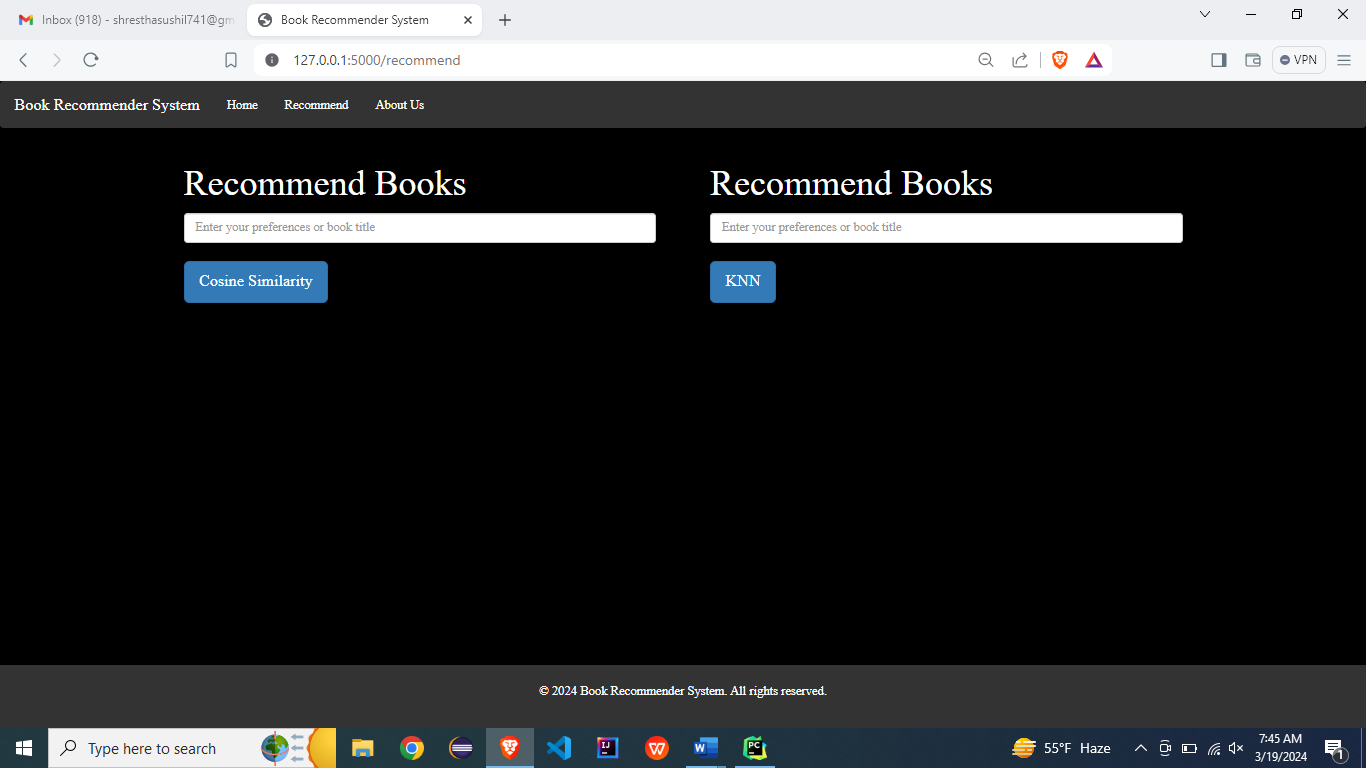
[9] “Cosine Similarity, Clearly Explained!!!” *YouTube*, YouTube, 30 Jan. 2023, www.youtube.com/watch?v=e9U0QAFbfLI&t=515s.

[10] “K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint.” *Www.Javatpoint.Com*, www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning. Accessed 19 Mar. 2024.

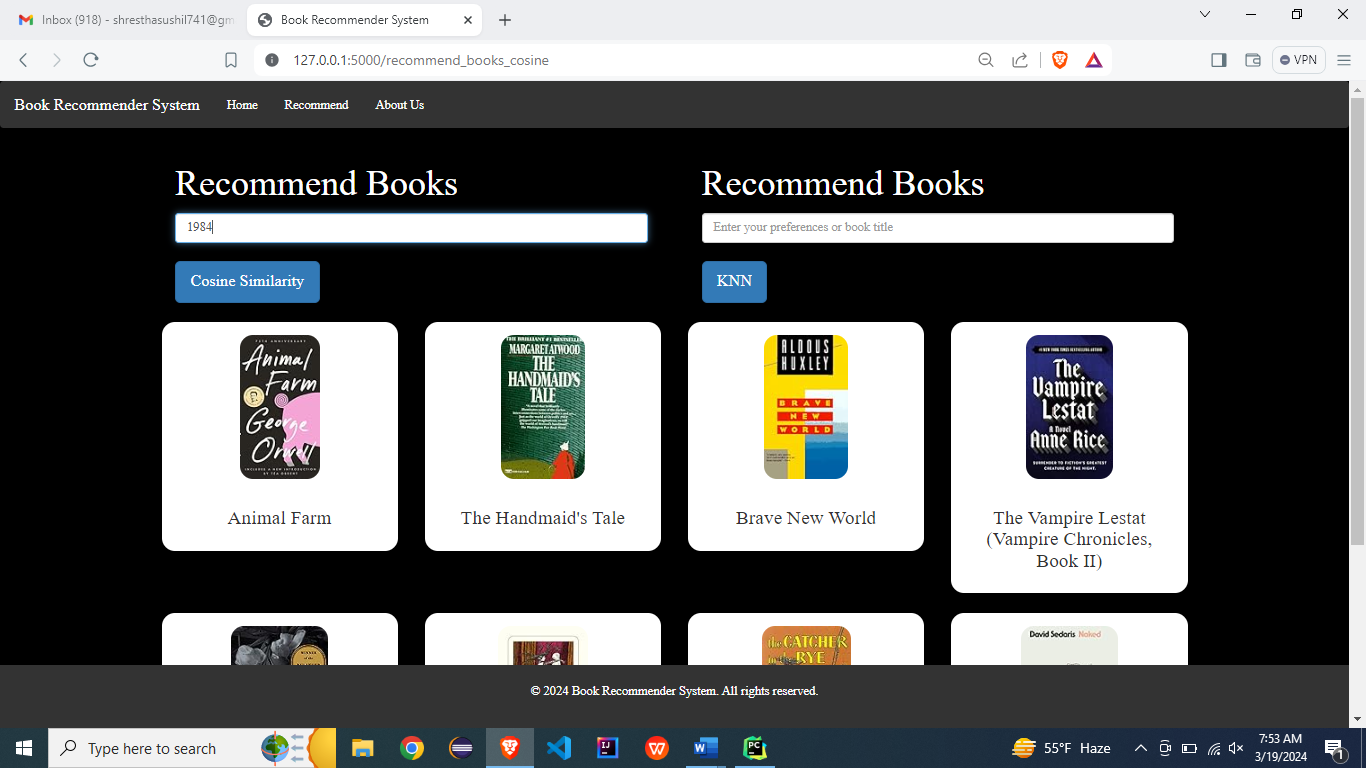
# APPENDICES

****

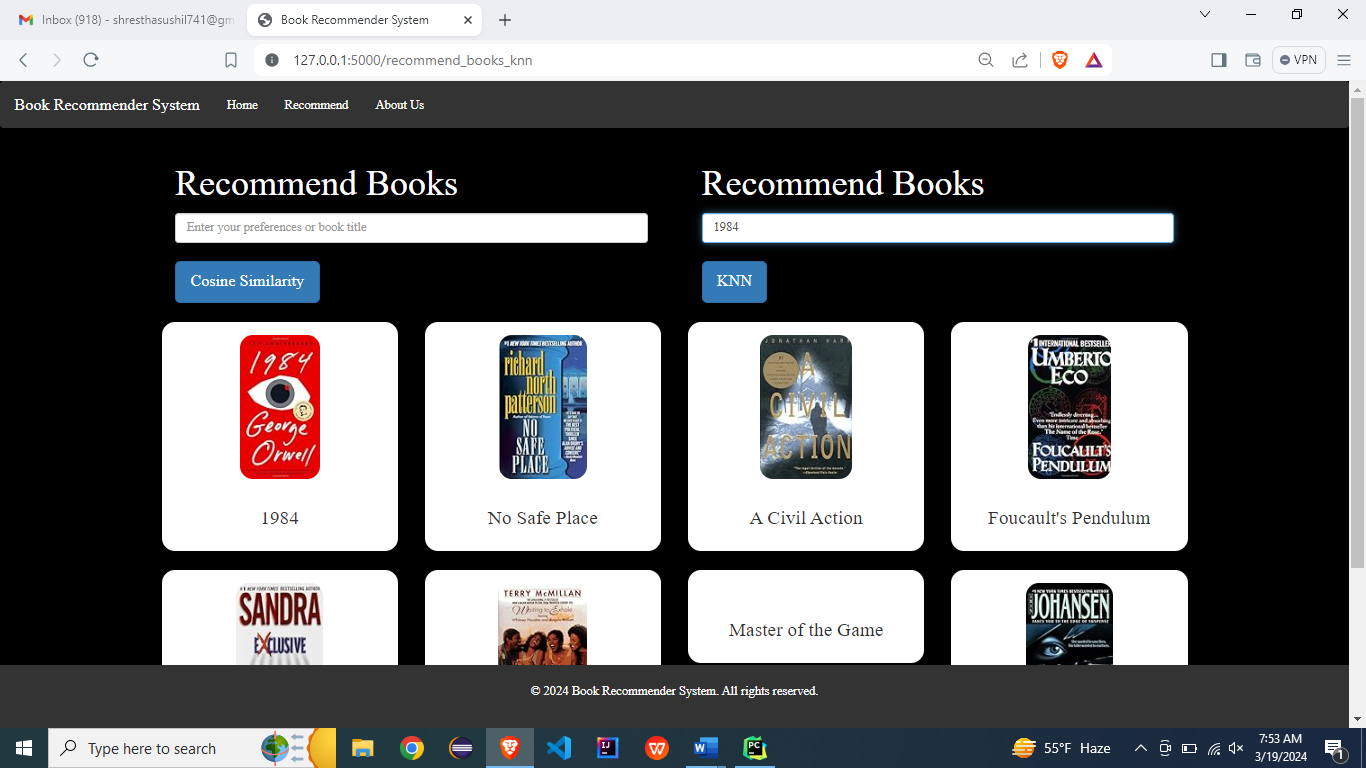
**Home Page**

****

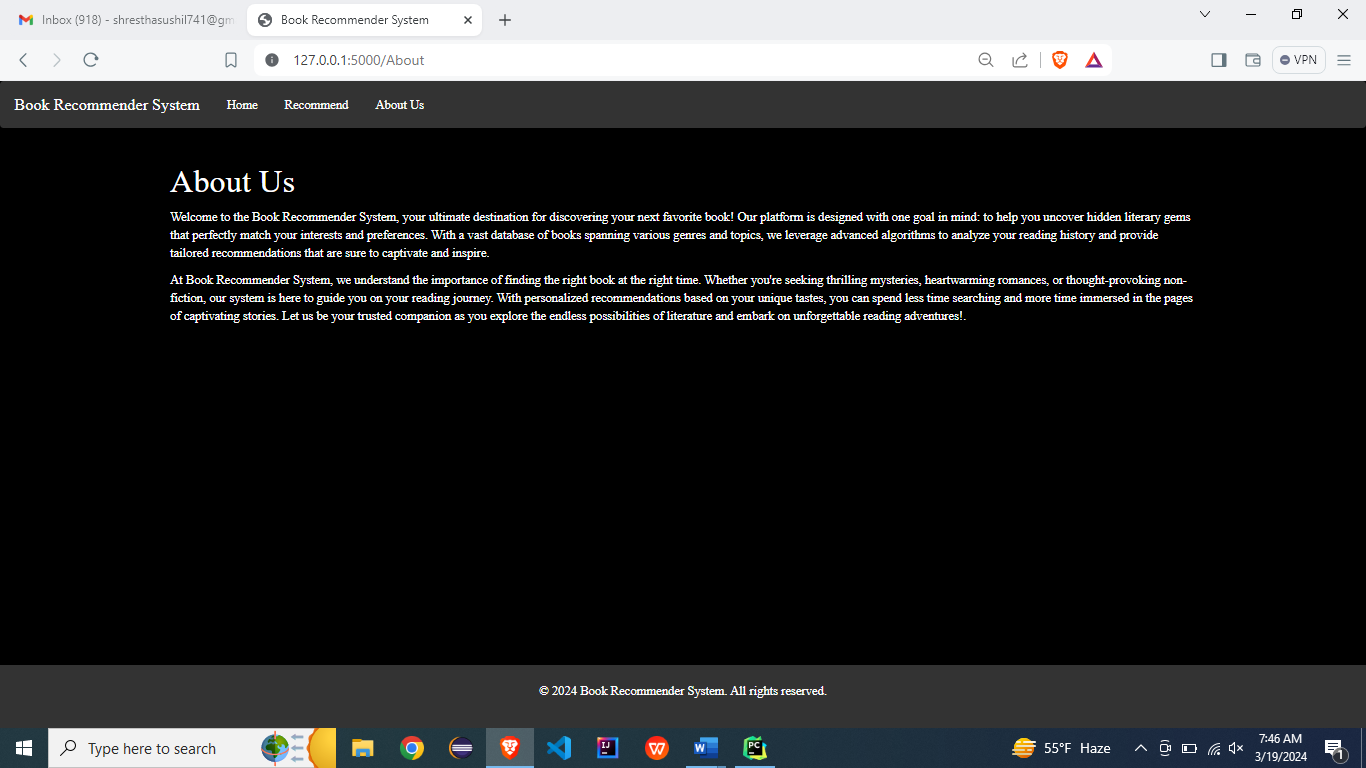
**Recommend Page**

****

**Recommendation using Cosine Similarity**



**Recommendation using K-NN**

****

**About Us Page**