SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES

INDORE

An INTERNSHIP REPORT

ON

Book Recommendation System using Collaborative Filtering

Company Name: TATA Consultancy Services (TCS)

Submitted to "Symbiosis University of Applied Sciences, Indore
As an internship report for the partial fulfillment of the award of degree of

BACHELOR OF TECHNOLOGY

IN

SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Submitted To:
Mr. Deepak Agrawal
Assistant Professor – CSIT

Submitted By:
Uttkarsh Bharadia– 2019BTCS085
Fourth Year – CSIT (2019-23)

SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES

INDORE

CERTIFICATE

This is to certify that the Internship report entitled, "Book Recommendation System using Collaborative Filtering," submitted by **Uttkarsh Bharadia (085)**, student in the fourth year towards partial fulfillment of the degree of Bachelor of Technology in the School of Computer Science and Information Technology in the year 2019-2023 Symbiosis University of Applied Sciences, Indore (M.P.) is in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology and is a bonafide record of the work carried by **Uttkarsh Bharadia (085)**, during the academic semester seventh.

Place: Indore

Date:

INTERNAL EXAMINER EXTERNAL EXAMINER

SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES

INDORE

RECOMMENDATION

The work entitled “Book Recommendation System using Collaborative Filtering,” submitted by **Uttkarsh Bharadia (085)** student of fourth year Computer Science and Information Technology, towards the partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Information Technology of Symbiosis University of Applied Sciences Indore (M.P.) is a satisfactory account of their Internship and is recommended for the award of the degree.

Endorsed By:

Dr. Neha Gupta

Director I/C, SCSIT

Student Undertaking

I hereby undertake that the project work entitled "Book Recommendation System using Collaborative Filtering" has been carried out by me from the period June 04, 2022 to December 09, 2022 and the report so prepared is a record of work done by me during my Internship. I further declare that I have completed the Internship in accordance with the Internship policy of the University. This Project report is submitted towards the fulfillment of my academic requirement and not for any other purpose.

I hereby undertake that the material of this Project is my original work and I have not copied anything from anywhere else. The material obtained from other sources has been duly acknowledged. I understand that if, at any stage, it is found that I have indulged in any malpractice or the Project and the project report has been copied or not completed by me, the University shall cancel my degree/withhold my result, and appropriate disciplinary action shall be initiated against me.

Student Name and Signature

Uttkarsh Bharadia

Mentor Name & Signature

Mr. Deepak Agrawal

Enrollment Number:

2019BTCS085

Head of School Name & Signature

Dr. Neha Gupta

Name of School: SCSIT

Date:

SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES

INDORE

ACKNOWLEDGEMENT

The successful completion of any work is generally not an individual effort. It is an outcome of the dedicated and cumulative efforts of a number of people, each having its own importance to the objective. This section is a value of thanks and gratitude towards all those persons who have implicitly or explicitly contributed in their own unique way towards the completion of the Project. For their invaluable comments and suggestions, I wish to thank them all.

Positive inspiration and the right guidance are a must in every aspect of life. Especially when we arrive at the academic stage, for instance. For the success of our Project, a number of obligations have been taken. We have performed the solemn duty of expressing a heartfelt thanks to Mr. Deepak Agrawal, SUAS – Assistant Professor, and TCS Industrial Mentor Mr. Akshay Pandit, who has endowed us with their precious perpetual guidance, suggestions, and information. Any kind of help, directly or indirectly, has proved important to us.

Table of Contents

Chapter 1: INTRODUCTION

1.1 Background of the Company

Tata Consultancy Services TCS, one of the leading multinational information technology businesses worldwide, was established in 1968 by Tata Sons. TCS is based in Mumbai. It was placed 66th on Forbes' list of the world's most innovative companies and has operations in numerous places across 46 countries.

With a broad network of innovation and delivery centers, Tata Consultancy Services (TCS) is a world leader in IT services, consulting, and business solutions. Through a strong dedication to our clients, in-depth industry knowledge, and a worldwide network of innovation and delivery centers, TCS ensures the greatest levels of assurance and satisfaction. TCS functions as a complete stakeholder in business, providing a consulting-led approach with an integrated portfolio of technology-led solutions that cover the whole Enterprise value chain. The customer-centric engagement model outlines how TCS interact with customers, providing customized services and solutions that are tailored to the specific requirements of your company.

TCS creates custom teams based on your industry and technological needs using their talent pool of over 488,649 international professionals, including 36.5% women from 154 different nationalities. TCS provides range of products are supported by our domain expertise, which was developed over decades of working in various sectors. The organizational structure of TCS is domain-driven and equipped to assist in offering Customers a single window to sector-specific solutions. Agile industry units have embedded features that enable quick reactions and give their customers a competitive edge. Additionally, there is a special Global Network Delivery Model (GNDM) (covering 40 global sites) that is now

1.2.1 Main activities/Business Organization

TCS provides a spectrum of business, technology, and engineering services and solutions that are led by consulting and powered by cognitive technology. This is provided using its distinctive Location Independent Agile™ delivery model, which is regarded as a gold standard for excellence in software development.

By using technology, TCS hopes to change the world:

- Internet of Things (IoT)
- Digital Engineering
- Sustainability Services
- Data Analytics

1.2.2 Organization Details

TCS's vision is to make working with us enjoyable for all parties involved while assisting clients in achieving their business goals through creative, best-in-class consulting, IT solutions, and services.

Natarajan Chandrasekaran serves as the Chairman of the Board of Tata Sons, a holding company that serves as the promoter of more than 100 Tata operating companies with combined annual revenues exceeding US $100 billion. In October 2016, he joined the Tata Sons board, and in January 2017, he was named chairman.

Chapter 2: THE PROJECT

2.1 Project definition

Problem Identification The stages of problem identification are accomplished in two ways: first, by conducting interviews related to phenomena that occur in the library environment, including both the flow of existing systems and the obstacles encountered by users when searching for books. The user, employees who work in the library, and employees who control the system are the subjects of the interview. Second, make observations on problems that arise to develop a hypothesis as information data to be analyzed.

Needs assessment

There are two stages of needs analysis: analysis of software requirements (software) for implementation needs and analysis of hardware requirements (hardware) for needs. This stage is a tool that will aid in the development of a system that will be designed in the future.

Design

The design phase consists of creating a workflow system or Entity Relational Diagram, database, and system interface design. design includes the creation of use cases and Activity Diagrams to aid in implementation. Then, using MySQL, create a database and a prototype display interface for each system function.

Implementation

Implementation stages are the application of the previous stages to create a complete system. At this stage, the Python programming language is used to implement designed systems

2.1.1 Objective

A library's Book Search System, in general, searches books by title, author, publisher, and book subject. However, with so many book search results displayed by the system, users (library members) may find it difficult to select books that match their profile. To accomplish this, we require a system that can provide recommendations in the book search process based on previous user searches. The goal of this research is to create a book search recommendation system in a desktop library using the Machine Learning algorithms. The recommendation system aims to reduce errors in obtaining the necessary reference books. The information obtained through the search process is pertinent to the user's requirements, allowing it to be useful right on target with user needs, allowing it to save time when searching for books The book search recommendation system employs a user-based collaborative filtering method based on the similarity of one member to another based on the lending pattern, which is organized into groups based about the book being sought. This system will rank the results of book search recommendations from highest to lowest and recommend book titles to users based on their profile.

2.1.2 Project scope

In order to be able to offer the best products to each user, the recommendation system must attempt to predict the choices that users will make about certain substances. Data sparsity, scalability, and grey sheep are some further issues with recommendation systems. When data is sparse, it is dispersed broadly and contains missing and null values. Scalability indicates that a large number of rating items make prediction challenging. Gray sheep denotes issues with time and memory requirements.

2.1.3 Literature Review

Recommendation systems filter information by anticipating consumer ratings or perceptions for goods they would be likely to use. It makes an effort to make product recommendations to customers based on their preferences and needs. The two main information filtering techniques used by RS are collaborative filtering and content-based filtering. When using content-based filtering, a consumer is recommended products that are similar in substance to those that they have already used. It creates a profile of the consumer in the beginning, including his or her preferences.

User resource, item resource, and the received a notification are the three main components of a complete RS. Similar to how the item model analyses the features of the items, the user model analyses the interests of the consumers. Then, the consumer's characteristics are compared with using the recommendation algorithm, determine which items to recommend based on item characteristics.

Okon et al. developed a model that uses an improved CF algorithm, a quick sort algorithm, and object-oriented analysis and design methodology to generate recommendations for buyers (OOADM). Because of the use of Firebase SQL, scalability was guaranteed. The evaluation criteria for this system showed good performance.

Kurmashov et al. (2015) evaluated the system using an online survey and used CF based on Pearson correlation coefficient to recommend books to readers online.

A system that saves information about the user's purchases of books was suggested by Mathew et al. in 2016. A hybrid algorithm that uses collaborative filtering, content-based filtering, and association rules to generate book recommendations from these book contents and ratings. They suggested using Equivalence class Clustering and bottom up Lattice Transversal (ECLAT) as opposed to Apriori because this algorithm is quicker because it only looks at the entire dataset once.

To make recommendations, Parvatikar et al. (2015) proposed item-based collaborative filtering and association rule mining. The Adjusted Cosine Vector Similarity function was used to calculate user similarity. Better recommendations were obtained as a result of the removal of the data sparsity issue through this method.

When using nearest neighbor based collaborative filtering, Ayub et al. (2018) proposed a similarity function that is like Jaccard Similarity to locate related items and users for the inquiring item and user. They suggested comparing the absolute value of ratings to the proportion of co-rated items used in the Jaccard Similarity formula. They also assessed how well their approach performed against other similarity metrics.

Aggarwal, C.C. claims that recommendation systems use the collaborative filtering method to create recommendations based on ratings submitted by other system users. The fundamental premise of CF is that if a customer rates one item similarly to another, it is likely that they would rate other goods similarly as well. Since computers are unable to evaluate qualitative elements like flavor or quality, recommendations based on the ratings of humans who can evaluate qualitative elements, i.e. teamwork, will produce superior results. In their work, Soanpet Sree Lakshmi and Dr. T. Adi Lakshmi have extensively discussed the issues with the CF recommendation system, including overspecialization, data sparsity, cold start, scalability, ranking of the recommendations, etc. Online item ratings are not something that buyers prefer to spend a lot of time on. As a result, rating information is typically scarce, which lowers the value of recommendations. Since new users haven't left any explicit or implicit assessments, it can be challenging to identify similarities and offer recommendations. This is referred to as a cold start issue. In this approach, the recommendation system will determine how comparable books in the Book dataset and books that the user has already rated are.

The following are the steps involved in developing the book recommender:

- Load the datasets and import the required libraries.
- Rename the relevant columns and delete the unneeded ones.
- Use the command books.head to determine the dataset's dependability()
- Utilize the CF matrix factorization.
- Make a matrix containing columns for users, indexes for books, and ratings for values.
- Take into account the users who gave at least 250 books a rating and provide each user 50 book ratings.
- Use ratings['user id'] for the exploratory data analysis.
- value counts().
- Users who have rated at least 250 books should be retrieved.
- Rank the books together with their ratings.
- Select only the books with at least a 50 rating.
- Assign users as columns, books as indices, and values as ratings in the pivot table definition.
- Convert the pivot entries into spare form before applying KNN training to build the model.
- Check the top 10 novels and apply the nearest neighbor.

A machine learning (ML) class called a recommendation engine is described as providing the user or business with the best ideas or recommendations. The recommendation system aids in building trust between the customer or the business and the services and goods. Collaborative filtering (CF), content-based filtering, and a combination of the two are the categories under which these systems fall. Based on the readers' tastes and interests, the book recommendation algorithm suggests books to them. The online sites make use of book recommenders that offer a variety of categories for electronic books [1, 2]. The development of a book recommender utilizing the CF technique is the main goal of this study.
Systems for making recommendations to consumers have developed as a result of the evolution of intelligent algorithms. They lessen the burden of selecting the finest option from the available options. Recommender systems can now be used in any industry, from network security to e-commerce, in the form of tailored services. By recommending products to consumers that cannot yet be requested, they benefit both the manufacturer and the consumer. Each recommender system consists of two components: a user and an item. Any consumer or buyer of a product who receives recommendations is considered a

user. A database of users and items can be used as input for the recommendation algorithm, and recommendations will undoubtedly be the output. A database of users and items can be used as input for the recommendation algorithm, and recommendations will undoubtedly be the output. Customers and book data are the system's inputs, while the output of the system is a list of recommended books. In this essay, a novel method for making book recommendations to customers is presented. To provide effective suggestions, this system combines association rule mining, collaborative filtering, and content filtering.

2.1.4 Current Conventional Model

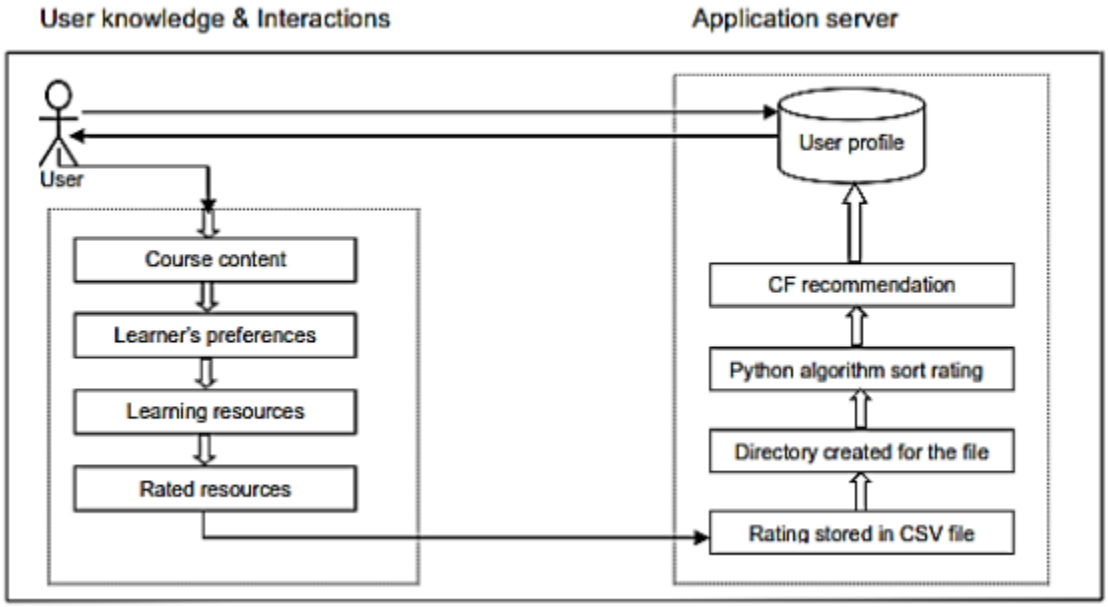

Figure 2.1: Recommendation System Conventional Approach

1. The User: The user is a global term that refers to any person or client who uses the web application's interface at any given time or location. A user is typically the one who accesses and explores the online courses in the massive open online course (MOOC) system. Users can rate books they've already taken. A user using the current system may be referred to as a user in this context.
2. Course Content: The course material is the aim of the current system, and users have access to a variety of course contents (resources). It is a collection of each online course's many contents.
3. Learner's preference: Referring to the user's preferred selection of online resources, we say that the learner's preference exists. Different online course options are available to users and learners, and a compilation of these options helps to effectively propose courses to other users.
4. Learning Resources: In this context, the term "learning resources" refers to the various types of educational materials that users of the selected course content have access to. These instructional resources can be found in a variety of formats, such as a slide presentation, audio tutorial, video tutorial, image tutorial, etc.
5. Rated Resources: When a learning resource is rated, it means that the user who is taking the online course has given it a rating of at least 5.

6. Rating stored in a (.csv) file: Comma separated value files are used to perform item recommendations on the backend and are made up of a variety of various rated resources and a user list (in form of columns).
7. Directory Created for the file: The path or location known as the directory establishes an absolute or relative link to the rating file.
8. Python algorithm for sorting rating: In this case, the use of the python programming language is used to implement the sorting of ratings step by step.
9. A Collaborative Filtering (CF) Recommendation: Using prior ratings from other students, this approach entails recommending educational resources to users. The user-based Collaborative Filtering (CF) algorithm is the kind of CF algorithm that is used by the current system.
10. User Profile: A user's personal information, implicit and explicit ratings, and a list of suggestions based on the user's similarity to other users are all described in the user profile.

Thus, the existing system uses a Scrum – leading agile software development methodology and applied a user-based collaborative filtering (CF) algorithm to compute and perform recommendations.

2.1.5 Proposed Model

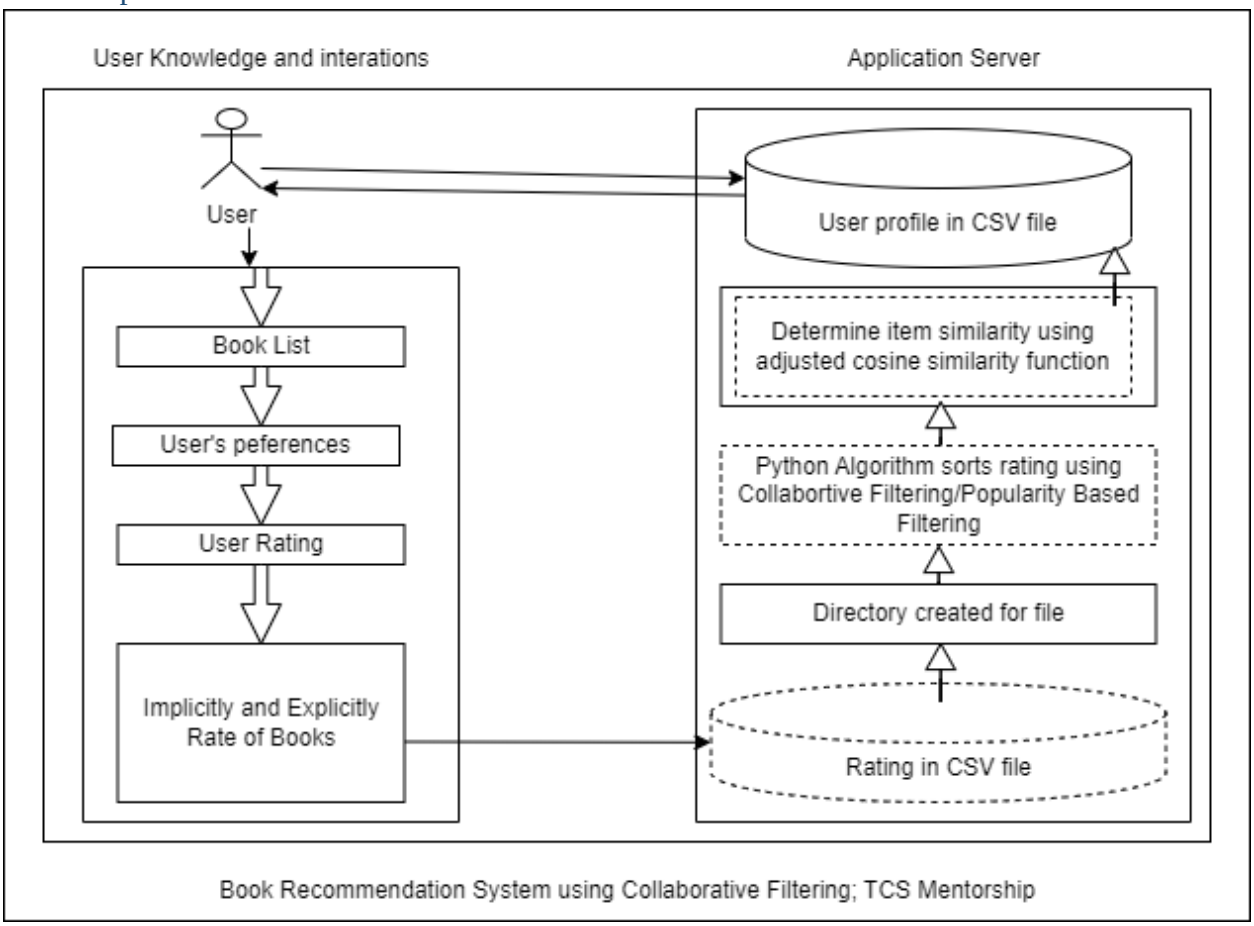

Figure 2.2: Proposed Methodology for Recommendation System

1. Book List: A collection of suggested or unrecommended books is provided to the user via the web interface as a book list in the proposed system.

2. User's Preferences: The user's preference is used to describe the user's preferred selection of suggested books. Users can choose from a variety of books. These user preferences are gathered together to help other users receive accurate book recommendations.
3. Users rating: The user rating is a term used to describe the act of users ranking books in an online bookstore.
4. Implicitly and Explicitly Rated Books: This is a collection of online book reviews that were gathered by expressly giving people the choice or interface to rate books. The implicit observation and recording of user interactions with the online bookshop is another method used to get data about rated books.
5. Rating stored in CSV: A CSV file containing a user list, a collection of variously rated books, is utilized to provide item recommendations.
6. Python Algorithm: The collaborative filtering process used by the Python program guarantees proper handling of the rating list.
7. Collaborative Filtering (CF) Recommender: Utilizing inputs, this entails creating a user-friendly book recommendation list. For a specific user, the CF Recommender engine uses an adjusted cosine similarity model to determine how closely related items are highly rated items in the CSV File shown below:

$$s(i,j) = \frac{\sum_{u \epsilon U}(R_{u,i} - R_u)(R_{u,j} - R_u)}{\sqrt{\sum_{u \epsilon U}(R_{u,i} - R_u)2} \quad \sqrt{\sum_{u \epsilon U}(R_{u,j} - R_u)2}} \quad \ldots (1)$$

   - The similarity score s in the rating system serves as a representation of a particular item i and another item (j) in the system (i,j). Every user u in a group of users u is represented by the string $u \epsilon U$.
   - The rating that user u provides for item (book) i is Ru,i
   - Additionally, Ru reflects the overall average rating that user u gave to all products user u reviewed.
   - Ru,j is the rating user u gives to next item (book) j.
   - Once the similarity index has been calculated and results have been acquired, the algorithm is used to sort the rated items in ascending order.
   - Finally, the results are sent to the user profile.

8. Directory Created for the file: To ensure that information from the book recommendation system homepage can be accessed, links are created for each entry in the CSV File.
9. Python Algorithm CF: In order to ensure proper handling of the rating list, the Python algorithm implements CF.
10. User Profile: The user profile defines a csv column that includes the user's personal information, implicit and explicit book ratings, and a list of suggestions based on the user's resemblance to other users.

Chapter 3: REQUIREMENTS ANALYSIS

3.1 Functional Requirements

1. External interfaces: These activities relate to how systems external to the primary system interface with it.

2. Algorithms: Any necessary formulas or data element operations can be captured by algorithms.

3. Historical data: If your database is dynamic, you will experience a growth in the amount of data it can hold, so you must specify the amount of storage you need.

The Following Functional Requirements through User Stories:

- User Story 1: as an user, I must be able to explore the book recommendation website
  1. Functional requirements:
     - Goes to the Book Recommendation System
     - Searches a Book
     - A feature to filter and organize the books
     - The Book Recommendation System returns with the book and similarities as

- User Story 2: as an user, I must be able to search top 50 Books
  2. Functional requirements:
     - Going to the Book Recommendation System
     - A feature that enables users to explore different genres, etc.
     - Going through the top 50 Books
     - Displaying the result (chosen book)

3.2 Non-Functional Requirements

1. Security Requirement: Keeping user's network safe from hackers or avoiding malicious packets being transferred to the user's device by installing in-built security such as Firewalls.
2. Data Integrity Requirement: Keeping user's data safe and secure from malicious attacks to the user's communication devices such as laptops, PCs or even avoid targeting servers.
3. Usability: Users can easily navigate an interface when an application is usable. For instance, if a user can successfully browse a streaming service's user interface (UI), they may comprehend how the application organizes its material and know how to get to sites like the settings page.
4. Compatibility: When running different programs on a device, highly compatible systems usually work effectively. Additionally, compatibility enables the use of the same applications by users of various operating systems (Windows 10/11; MacOS, Linux, etc.).

3.3 Use-Case Specification

UML, which stands for Unified Modified language, is a standardized modeling language that helps developers understand the architecture, design, and implementation of complex software systems. It is a visual representation used to keep track of the relationships and hierarchies for a specific software system. It is popular because it is platform-independent, and any programmer can use it to implement the code in its language for platforms and requirements.

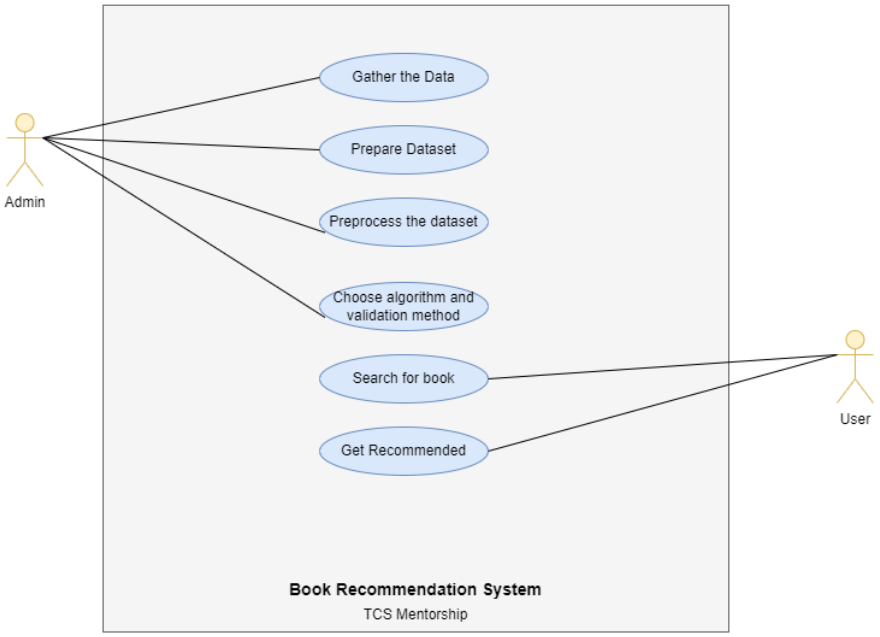

Figure 3.1: Use-Case Diagram of BRS

In our Project, we have deployed a book recommendation system whose use case diagram shows a total of six levels that define the basic working of our Project.

We collected our dataset from Kaggle and other sources for this Project. The dataset is provided to our Project that further prepare it by cleaning and refining purposes. Then this refined dataset is offered to preprocessors to sort ratings into ascending order for respective books based on their popularity among readers. This processed data is provided to our algorithm that filters our data using collaborative and popularity filtering.

Further defining filtering popularity filtering deals with the popularity of authors and their content among the reader community using their universal statistics and rating on various trusted platforms. Then collaborative filtering is also one of the most used filtering by famous media like Good-Reads, and kindle; that states that if the users are multiple for a specific account, then their interests based on frequency are refined for respective users like a mutual suggestion.

After that, our results are stored in the csv file, and when the user searches for its respective content, the Project will automatically analyze and provide a recommendation to the user that fulfills our motive.

3.3.1 Find Actors

- User: Using the book recommendation system via the internet.
- Admin: Having control over the book recommendation system, where new books are being added into the system and running the collaborative filtering algorithm to recommend books to the users.

Chapter 4: DESIGN

4.1 Class diagrams

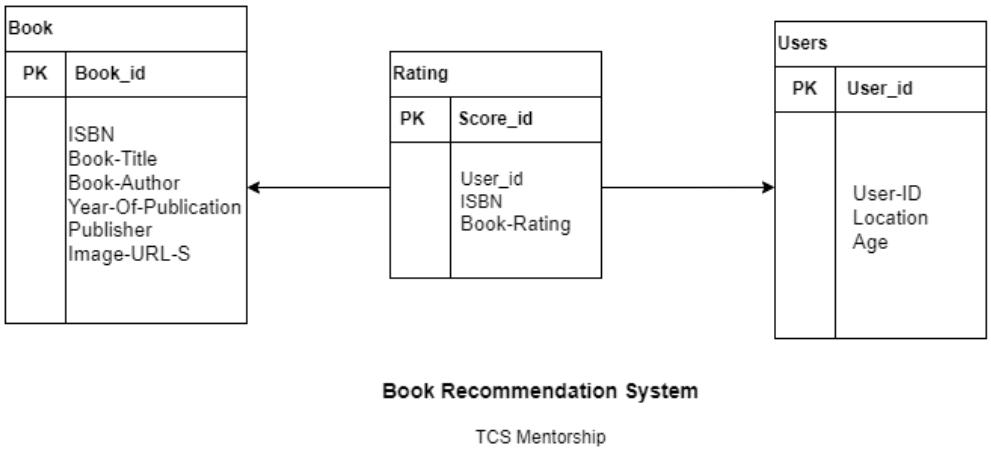

Figure 4.1: Class Diagram of BRS

4.2 Activity diagrams

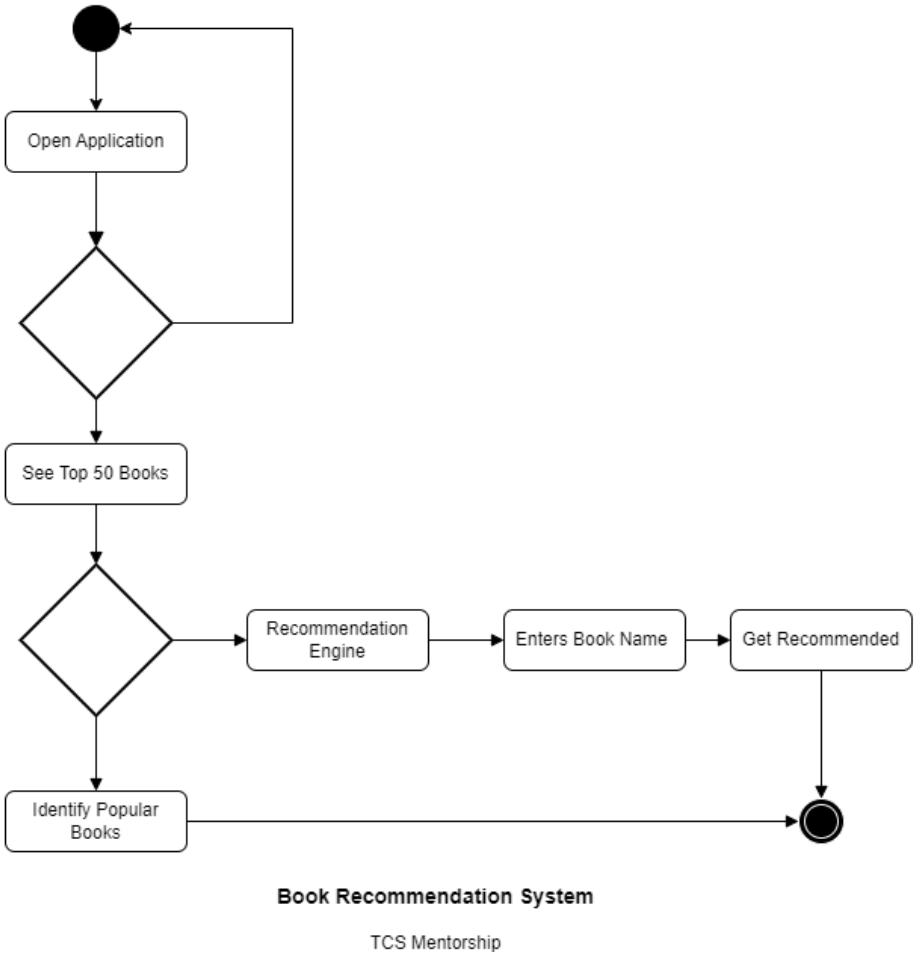

Figure 4.2: Activity Diagram of BRS

Chapter 5: EXPERIMENTAL RESULT AND TESTING

5.1 Test cases developed

The test cases developed in the Book Recommendation System with the following outcomes.

1. Homepage of the Book Recommendation System (BRS)

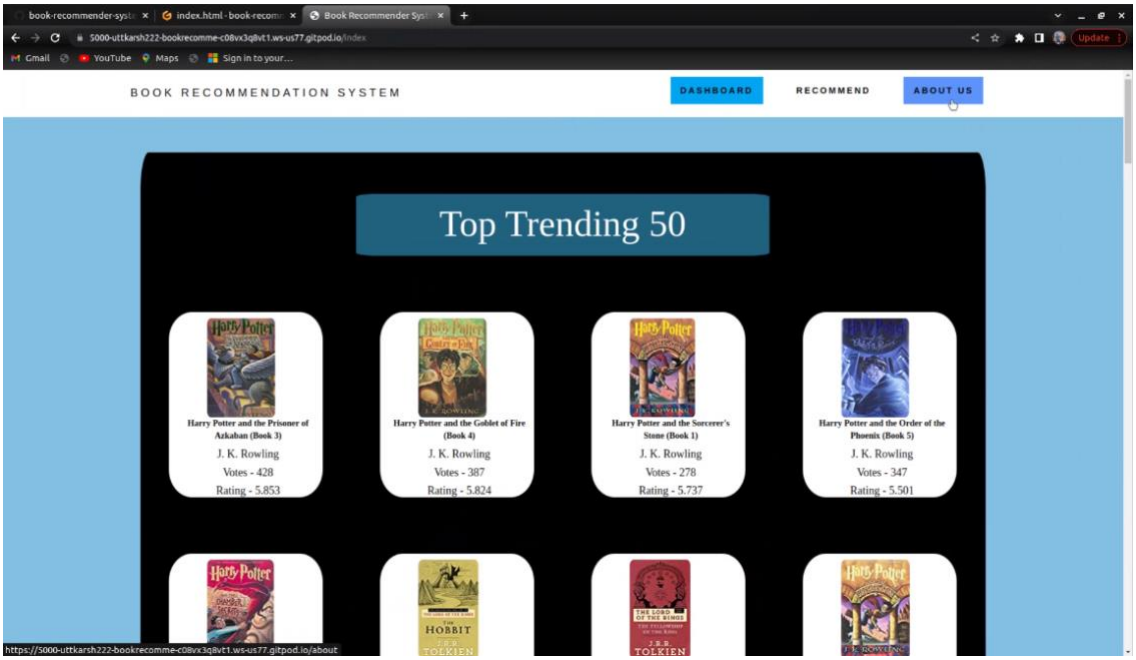

Figure 5.1: Homepage of BRS

2. Testing the BRS by going to Recommend Tab

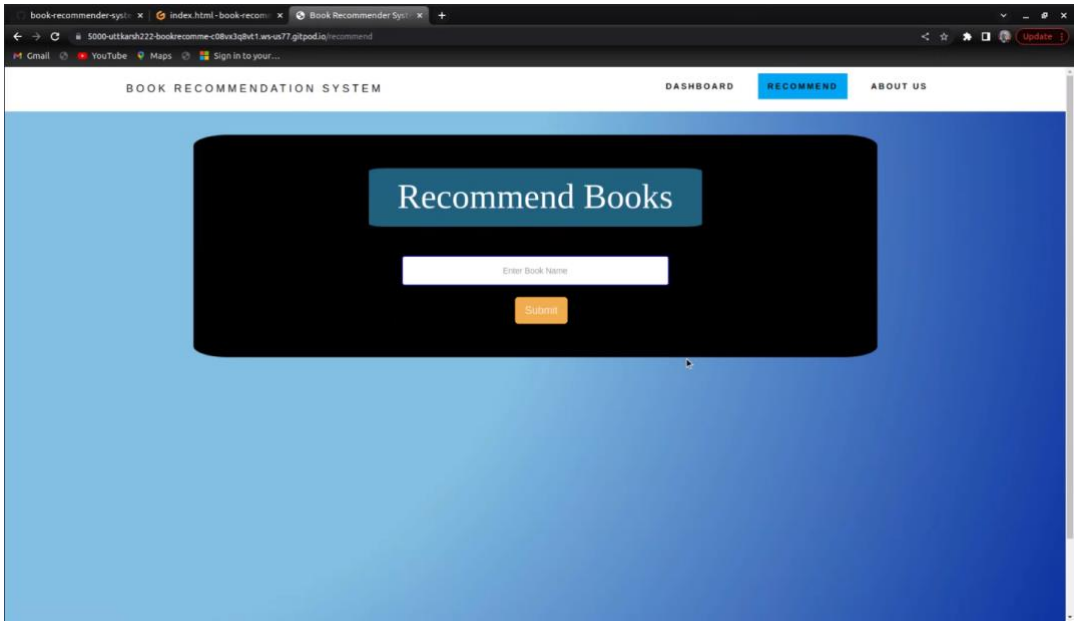

Figure 5.2: Recommend Tab Interface of BRS

3. Testing the BRS by going to Recommend Tab; Searching, “To Kill a Mockingbird”

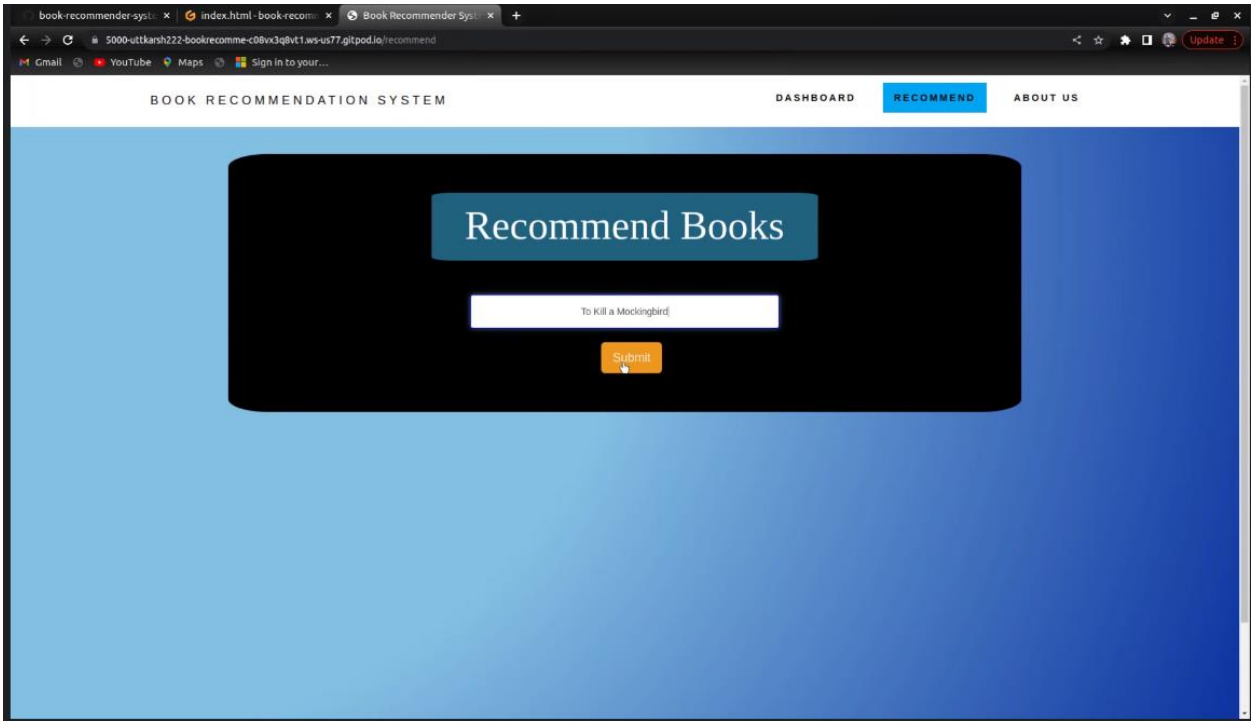

Figure 5.3: Testing of Recommend Tab (in BRS)

4. Results upon searching To Kill the Mockingbird in BRS

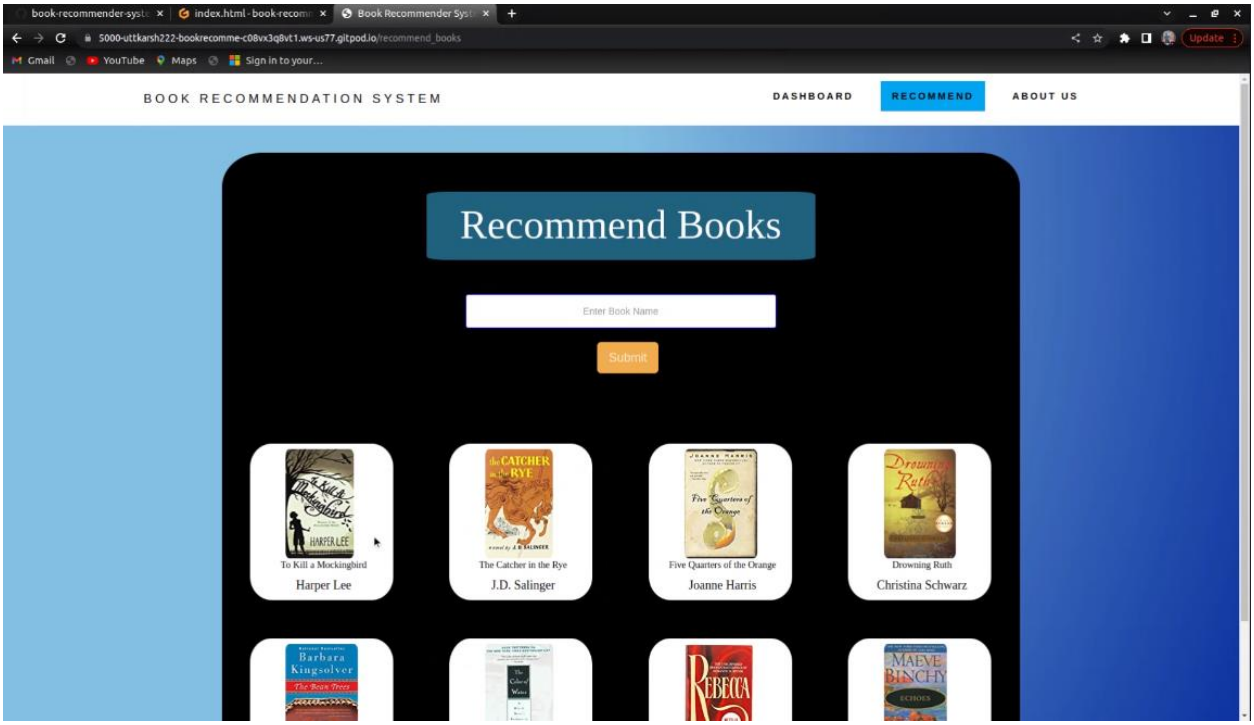

Figure 5.4: Results of Testing – Input 1

5. Testing the BRS by going to Recommend Tab; Searching, “114” (random search)

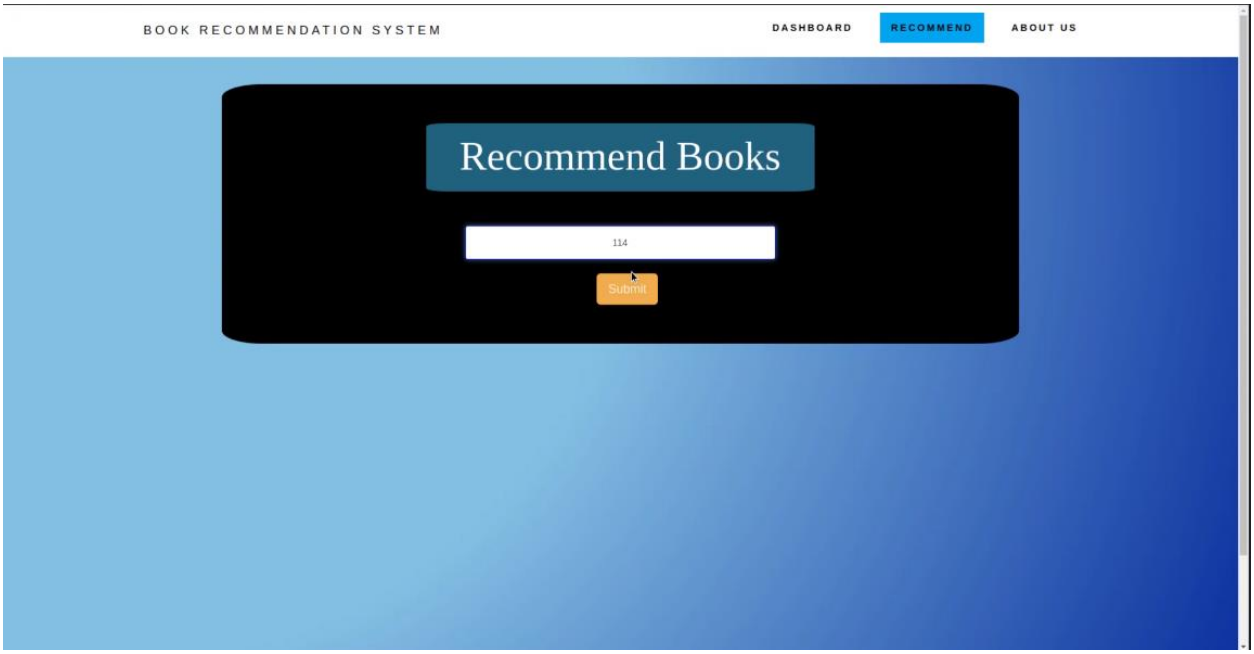

Figure 5.5 Testing of Recommend Tab (in BRS) with alternate Input – 2

6. Results obtained after searching, “114”

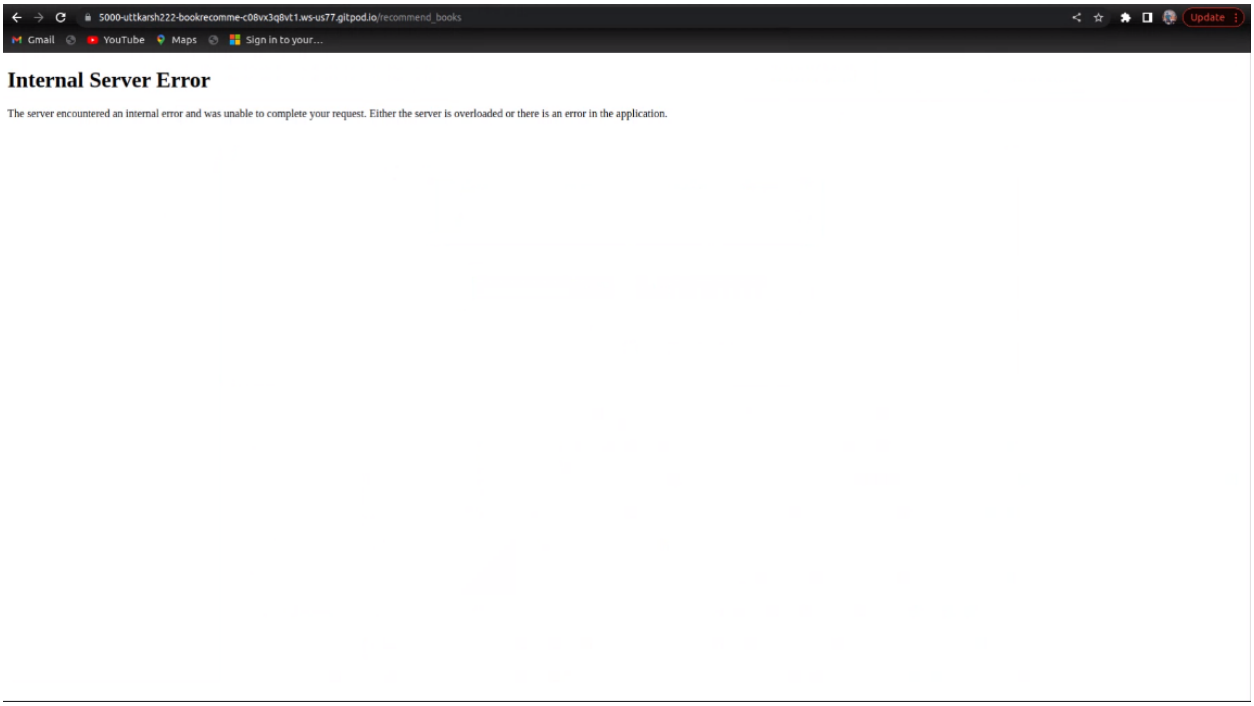

Figure 5.7: Results of Testing – Input 2

7. Testing the BRS by going to, “About Us” Tab

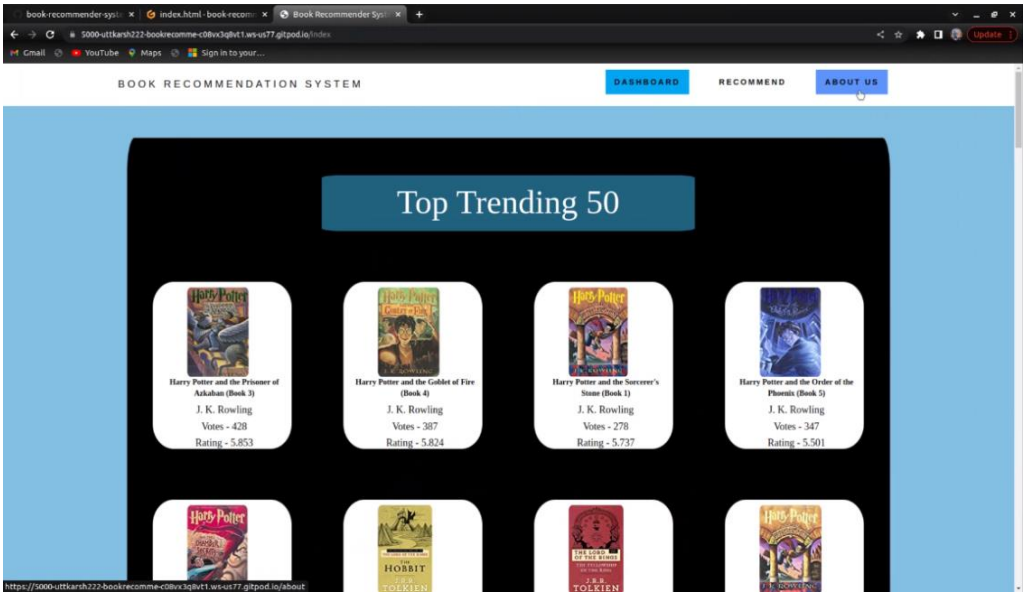

Figure 5.8: Testing of About Us Tab (in BRS)

8. Results upon going to About Us Page

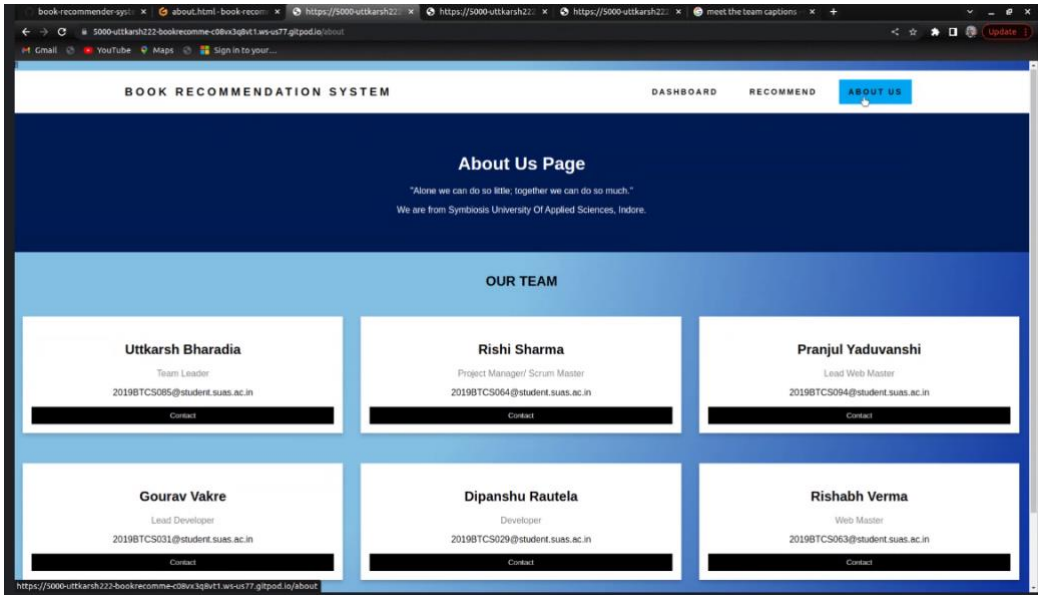

Figure 5.7: Results of Testing About Us Tab

5.2 Testing used in our Project

The Book Recommendation System project links can be accessed here:

- Project Code: https://github.com/iampranjul/book-recommender-system
- Project Website: https://book-recommender-yag2.onrender.com/

Chapter 6: DAILY DAIRY

6.1 Monthly Diary

Table 6.1: Daily Month 1

MONTH - 1	JUNE – JULY		
Department/Division	SCSIT	Name of finished Product	Book Recommendation System using Collaborative Filtering
Name of the Faculty Mentor	Er. Deepak Agrawal deepak.agrawal@suas.ac.in		
Main Points of the Month	1. Choose a topic from the TCS Mentorship 2. Research about the chosen topic 3. Gather materials to understand the concepts 4. Understand Jira SCRUM BOARD and start implementing it		

Table 6.2: Daily Month 2

MONTH - 2	JULY – AUGUST		
Department/Division	SCSIT	Name of finished Product	Book Recommendation System using Collaborative Filtering
Name of the Faculty Mentor	Er. Deepak Agrawal deepak.agrawal@suas.ac.in		
Main Points of the Month	1. Start gathering resources on Book Recommendation System (BRS) 2. Understand the theoretical concepts of needed in BRS 3. Get familiarized with Machine Learning 4. Check SCRUM Board for any pending tasks remaining		

Table 6.3: Daily Month 3

MONTH - 3	AUGUST – SEPTEMBER		
Department/Division	SCSIT	Name of finished Product	Book Recommendation System using Collaborative Filtering
Name of the Faculty Mentor	Er. Deepak Agrawal deepak.agrawal@suas.ac.in		
Main Points of the Month	1. Start designing a basic plan of BRS 2. Get familiarized with the main requirements 3. Understand Collaborative Filtering 4. Check SCRUM Board for any pending tasks remaining		

Table 6.4: Daily Month 4

MONTH - 4	SEPTEMBER – OCTOBER		
Department/Division	SCSIT	Name of finished Product	Book Recommendation System using Collaborative Filtering
Name of the Faculty Mentor	Er. Deepak Agrawal deepak.agrawal@suas.ac.in		
Main Points of the Month	1. Get a hands-on experience with Google Collaboratory 2. Start working on the Algorithm 3. Begin creating the documentation; Software Requirement Specification 4. Check SCRUM Board for any pending tasks remaining		

Table 6.5: Daily Month 5

MONTH - 5	OCTOBER – NOVEMBER		
Department/Division	SCSIT	Name of finished Product	Book Recommendation System using Collaborative Filtering
Name of the Faculty Mentor	Er. Deepak Agrawal deepak.agrawal@suas.ac.in		
Main Points of the Month	1. Start testing the Algorithm 2. Start and finish working on the Web Application of BRS 3. Start working on the Main Report 4. Check SCRUM Board for any pending tasks remaining		

Table 6.6: Daily Month 6

MONTH - 6	NOVEMBER – DECEMBER		
Department/Division	SCSIT	Name of finished Product	Book Recommendation System using Collaborative Filtering
Name of the Faculty Mentor	Er. Deepak Agrawal deepak.agrawal@suas.ac.in		
Main Points of the Month	1. Start and finish the testing of Web Application 2. Finish up the documentation 3. Start and Finish working on the BRS PowerPoint Presentation 4. Check SCRUM Board for any pending tasks remaining		

6.2 Book Recommendation System Timeline:

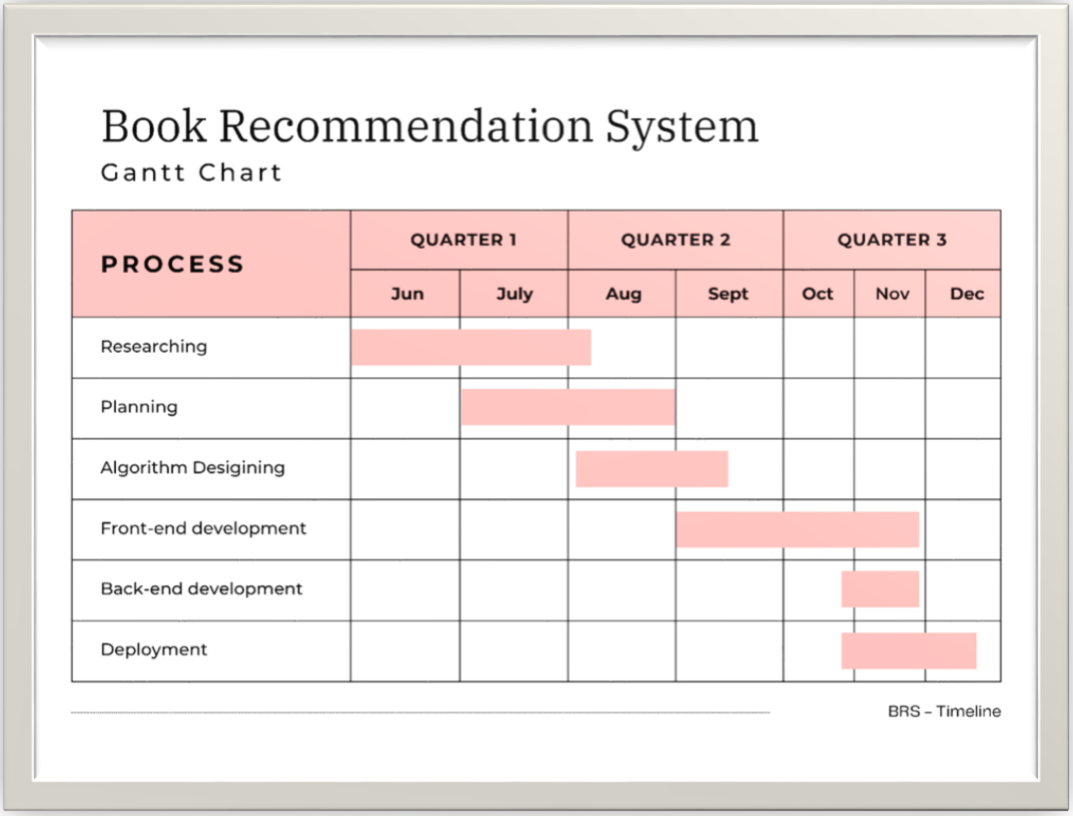

Figure 6.1: Timeline of BRS (Gantt Chart)

6.3 Jira Dashboard (SCRUM Board)

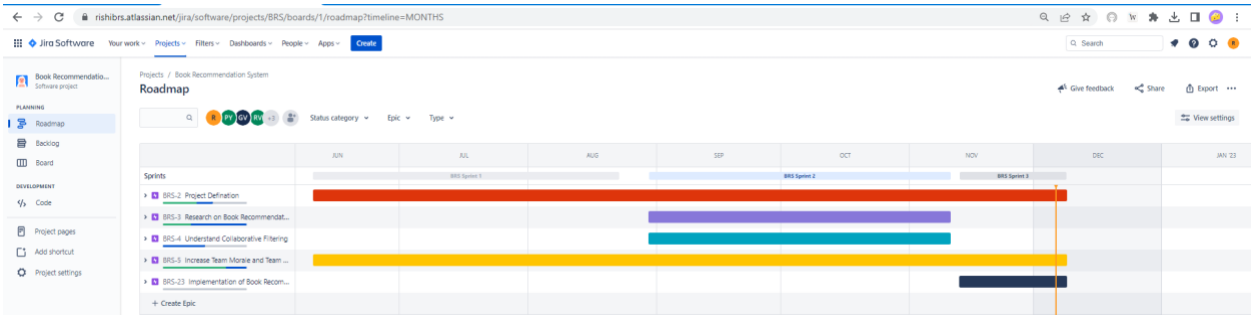

Figure 6.2: SCRUM Dashboard; Roadmap

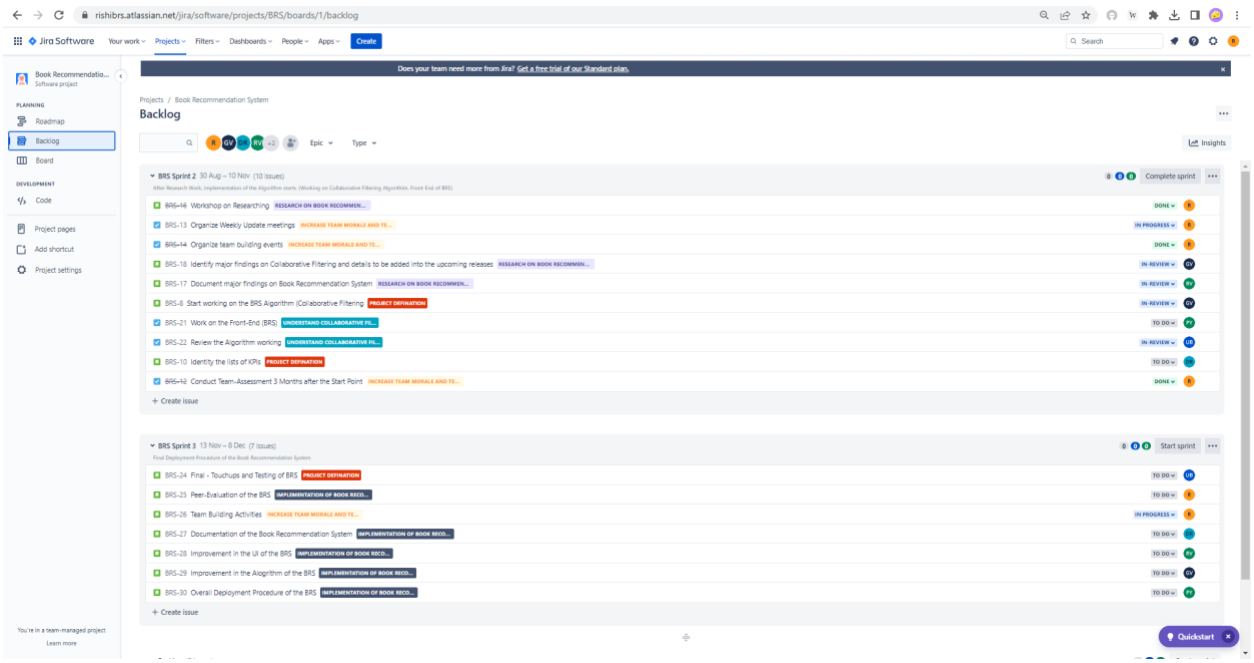

Figure 6.3: SCRUM Dashboard; Backlog

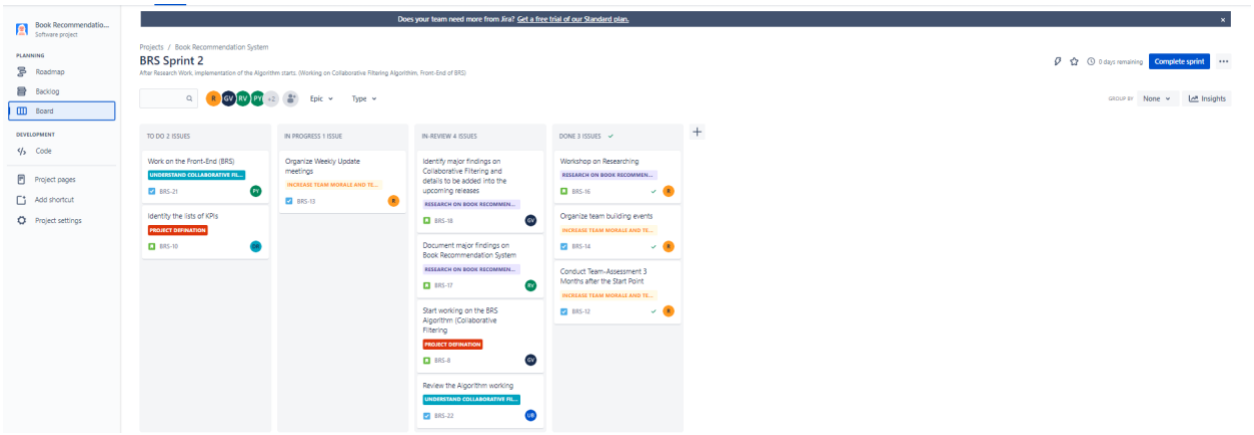

Figure 6.4: SCRUM Dashboard; Active Sprints 2 & 3

Chapter 7: CONCLUSION

7.1 Problems and Issues in currents system

This paper was able to present a comprehensive review on researches previously targeted on improving recommender systems. It also introduced to cosine similarity and collaborative filtering algorithm to improve on recommender systems. From the results and visualizations, we can deduce that the accuracy of rating followed a normal distribution which suggests consistency and efficiency

As an alternative, you might choose an off-the-shelf solution provided by a third party company, but how would you know which of the many solutions on the market would be best for your organization? It can take a long time to analyze alternative solutions because you have to look at their case studies, technology, integration with your present business model, and other factors. For instance, is the best suggestion method for your website collaborative filtering? A better option might be content-based screening.

The more information the algorithm has about the user, the more precise the recommendations it makes. However, considering multiple high-profile instances of consumer data leaks in recent years, many customers are reluctant to provide personal information. But the recommendation engine can't work well without this client information. Therefore, it's crucial to develop customer and business trust.

7.2 Future extension

The System has adequate scope for modification in future if it is necessary. Development and launching of Mobile app and refining existing services and adding more service, System security, data security and reliability are the main features which can be done in future. The API for the shopping and payment gateway can be added so that we can also buy a book at the moment. In the existing system there are only some selected categories, so as an extension to the site we can add more categories as compared to existing site. Also, we can add admin side with some functionalities like books management, User management etc. Hence, future works should target on securing recommender system data against attacks and improving on the algorithms used.

Appendix

REFERENCE

1. Okon, E.U., Asagba, P.O., and Eke, B.O. (2018). a more effective algorithm for collaborative filtering that recommends books online. Volume 179, No. 46, June 2018, International Journal of Computer Applications (0975- 8887)
2. Nussipbekov, A., Kurmashov, N., and Konstantin (2015). System for recommending books online.12th International Conference on Electronics, Computer, and Computation Proceedings (ICECC)
3. Kuriakose, B., Mathew, P., and Hegde, V. (2016). System for recommending books using a collaborative, content-based filtering approach. International Conference on Data Mining and Advanced Computing Proceedings (SAPIENCE)
4. S. Parvitikar and B. Dr. Joshi (2015). Utilizing association mining and collaborative filtering, an online book recommendation system. International Conference on Computational Intelligence and Computing Research Proceedings, IEEE (ICCIC)
5. The authors are Ayub, M., Ghazanfar, M.A., Maqsood, and A. Saleem (2018). CF-based recommendation system performance is improved by a Jaccard-based similarity metric. International Conference on Information Networking Proceedings (ICOIN)
6. Yousri, N.A., Sabek, I., Samir, M., Badawy, M., and Elgohary, A. (2010). Wiki-rec is a semantic recommendation engine that makes use of Wikipedia as its ontology. during the 10th International Conference on Intelligent Systems Design and Applications (ISDA).
7. Oku, K., Uemura, S., Nakajima, S., and Miyazaki (2006). Context-sensitive SVM for information recommendation in various contexts. International Conference on Mobile Data Management's MDM'06 Proceedings.
8. R. Ghani and A. Fano (2002). constructing recommender systems utilising a product semantics database. In the proceedings of the second international conference on adaptive web-based systems and adaptive hypermedia.
9. Miyahara, K., and M.J. Pazzani (2000) Cooperative filtering using a straightforward Bayesian classifier. International Conference on Artificial Intelligence for the Pacific Rim Proceedings.
10. Algorithms and Methods in Recommender Systems, D. Asanov, 2011. Germany's Berlin Institute of Technology
11. Lakshmi, T.A. and S.S. Lakshmi (2014). Problems and difficulties with recommendation systems.
12. International Journal of Computer Science and Information Technologies (IJCSIT), Volume 5 (4), 2014, 5771–5772
13. Okon, E.U., Asagba, P.O., and Eke, B.O. (2018). a more effective algorithm for collaborative filtering that recommends books online. Volume 179, No. 46, June 2018, International Journal of Computer Applications (0975- 8887)
14. Nussipbekov, A., Kurmashov, N., and Konstantin (2015). System for recommending books online.
15. 12th International Conference on Electronics, Computer, and Computation Proceedings (ICECC)
16. Kuriakose, B., Mathew, P., and Hegde, V. (2016). System for recommending books using a crowdsourced, content-based filtering approach. International Conference on Data Mining and Advanced Computing Proceedings (SAPIENCE)
17. The authors are Ayub, M., Ghazanfar, M.A., Maqsood, and A. Saleem (2018). CF-based recommendation system performance is improved by a Jaccard-based similarity metric. International Conference on Information Networking Proceedings (ICOIN)
18. Majumdar, A., and Gogna, A. (2015). An All-Inclusive Recommender System Model: Increasing Accuracy for Users of Cold and Warm Starts. 2015's IEEE Access Vol. 3, pp. 2803-2813.

19. Thus, H., Chatti, M.A., Dakova, S., and Schroeder (2013). Recommendation for Tag-Based Collaborative Filtering in Personal Learning Environments IEEE Transactions on Learning Technologies, October-December 2013, Vol. 6, No. 4
20. "Recommendation System in E-Commerce Websites: A Graph Based Approached," 2017 IEEE 7th International Advance Computing Conference (IACC), pp. 931–934, doi: 10.1109/IACC.2017.0189. S. Shaikh, S. Rathi, and P. Janrao.
21. Journal of Zhejiang University of Technology, vol. 47, no. 4, pp. 425-429, 2019. P. Li and Z. Hong, "Book selection algorithm based on the interest and type factor for university."
22. Computer Applications and Software, vol. 29, no. 12, pp. 21–23, 2012. P. Wang, "Research and practise of book recommendation system based on SNS."
23. Computer Applications and Software, vol. 38, no. 5, pp. 288-293, 2021. C. Liu, "Collaborative filtering hybrid recommendation algorithm based on enhanced biased and cluster user closest neighbour."
24. Jeong, Y.S., Jeong, M.K., and Choi, S.H. (2010). A Hybrid Recommendation Method for Large-Scale Applications with Reduced Data. Applications and Reviews, Volume 40, Number 5, September 2010, IEEE Transactions on Systems, Man, and Cybernetics-Part C.
25. Computer Applications and Software, vol. 38, no. 5, pp. 288-293, 2021. C. Liu, "Collaborative filtering hybrid recommendation algorithm based on enhanced biassed and cluster user closest neighbour."
26. Journal of Beijing Information Science and Technology University (Natural Science Edition), vol. 32, no. 4, pp. 90-94, T. Huang, "Application research of collaborative filtering algorithm based on neighbour model," 2017.
27. Solving Graph Coloring Problem Using Divide and Conquer-Based Turbulent Particle Swarm Optimization. Marappan, R., Sethumadhavan, G. Arab J Sci Eng (2021). https://doi.org/10.1007/s13369-021-06323-x
28. Complexity Analysis and Stochastic Convergence of Some Well-Known Evolutionary Operators for the Solution of the Graph Coloring Problem. Marappan, R.; Sethumadhavan, G. 8, 303 in mathematics in 2020. https://doi.org/10.3390/math8030303
29. Solution to Graph Coloring Using Genetic and Tabu Search Procedures. Marappan, R., Sethumadhavan, G. 43, 525-542 Arab J Sci Eng (2018). https://doi.org/10.1007/s13369-017-2686-9
30. A New Genetic Algorithm for Graph Coloring, Fifth International Conference on Computational Intelligence, Modeling and Simulation, 2013, pp. 49–54, doi: 10.1109/CIMSim.2013.17, R. Marappan and G. Sethumadhavan.