



*Mini Project Report On*

## **LitLink :- A Book Recommendation System**

*Submitted in partial fulfillment of the requirements for the  
award of the degree of*

**Bachelor of Technology**  
*in*  
**Computer Science & Engineering**

By

**JK Yaswanth (U2103105)**

**Eldho Markose (U2103084)**

**Jeevan James Mathew (U2103107)**

**Eshaan Kolloth (U2103086)**

**Under the guidance of**

**Mrs.Mehbooba P Shareef**

**Department of Computer Science & Engineering  
Rajagiri School of Engineering & Technology (Autonomous)  
(Affiliated to APJ Abdul Kalam Technological University)  
Rajagiri Valley, Kakkanad, Kochi, 682039  
May 2024**

# CERTIFICATE

*This is to certify that the mini project report entitled "**Litlink :- A Book Recommendation System**" is a bonafide record of the work done by **JK Yaswanth (U2103105), Eldho Markose (U2103084), Jeevan James Mathew (U2103107), Eshaan Kolloth (U2103086)**, submitted to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Ms. Mehbooba P Shareef  
Assistant Professor  
Dept. of CSE  
RSET

Dr. Saritha S  
Professor  
Dept. of CSE  
RSET

Dr Preetha K.G  
Professor and Head of Department  
Dept. of CSE  
RSET

## **ACKNOWLEDGEMENTS**

We wish to express my sincere gratitude towards Dr P. S. Sreejith, Principal of RSET, and Dr. Preetha K.G., Professor and Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our mini project, "Litlink :- A Book Recommendation System".

we are highly indebted to my project coordinators, **Dr. Saritha S**, Professor, Department of Computer Science and Engineering for her valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Mrs Mehbooba P Shareef** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, we would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

**JK Yaswanth (U2103105)**

**Eldho Markose (U2103084)**

**Jeevan James Mathew (U2103107)**

**Eshaan Kolloth (U2103086)**

## Abstract

In contemporary society, where social media's short-form content prevails, the habit of leisure reading is diminishing, posing a challenge to traditional book consumption and fostering shorter attention spans. To address this issue, we propose a novel solution: a machine learning-based book recommendation system integrated into a user-friendly website. This system caters to both casual readers and book enthusiasts, leveraging machine learning algorithms to provide tailored recommendations based on users' interests and past reading habits.

Our project encompasses expertise in web design, machine learning modeling, databases, and website backend systems. By deploying a comprehensive dataset of books, our platform offers a diverse range of recommendations, ensuring relevance and engagement for users. The website interface facilitates seamless interaction, allowing users to create accounts, personalize preferences, and receive curated suggestions effortlessly.

Compared to existing works, our project stands out through its integration of advanced machine learning techniques, such as reinforcement learning, to enhance user experience and decision support. Moreover, the incorporation of a diverse book dataset ensures a rich and varied selection of recommendations, appealing to a wide audience. Through this innovative approach, we aim to revitalize the culture of reading in the digital age, bridging the gap between leisure and non-leisure readers.

**Keywords:** *Book Recommendation System, Machine Learning, Collaborative Learning, User Experience, Reading Habits, User Data Safety.*

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Definition . . . . .	2
1.3 Scope and Motivation . . . . .	2
1.4 Objectives . . . . .	2
1.5 Challenges . . . . .	3
1.6 Assumptions . . . . .	3
1.7 Societal / Industrial Relevance . . . . .	4
1.8 Organization of the Report . . . . .	4
<b>2 Software Requirements Specification</b>	<b>6</b>
2.1 Introduction . . . . .	6
2.1.1 Purpose . . . . .	6
2.1.2 Product Scope . . . . .	6
2.2 Overall Description . . . . .	6
2.2.1 Product Perspective . . . . .	7
2.2.2 Product Functions . . . . .	7
2.2.3 Operating Environment . . . . .	8
2.2.4 Design and Implementation Constraints . . . . .	8
2.2.5 Assumptions and Dependencies . . . . .	8
2.3 External Interface Requirements . . . . .	8

2.3.1	User Interfaces . . . . .	9
2.3.2	Hardware Interfaces . . . . .	9
2.3.3	Software Interfaces . . . . .	9
2.3.4	Communications Interfaces . . . . .	10
2.4	System Features . . . . .	10
2.4.1	Personalised Reinforcement Recommendation . . . . .	10
2.4.2	User Profile Management . . . . .	11
2.4.3	Bookshelf Management . . . . .	12
2.5	Other Nonfunctional Requirements . . . . .	13
2.5.1	Performance Requirements . . . . .	13
2.5.2	Safety Requirements . . . . .	14
2.5.3	Security Requirements . . . . .	14
2.5.4	Software Quality Attributes . . . . .	15
<b>3</b>	<b>System Architecture and Design</b>	<b>17</b>
3.1	System Overview . . . . .	17
3.2	Architectural Design . . . . .	19
3.3	Dataset identified . . . . .	19
3.4	Proposed Methodology/Algorithms . . . . .	20
3.5	User Interface Design . . . . .	21
3.6	Database Design . . . . .	23
3.7	Description of Implementation Strategies . . . . .	23
3.8	Module Division . . . . .	24
3.8.1	K-Nearest Neighbors (KNN) Model . . . . .	24
3.8.2	Front End . . . . .	25
3.8.3	Back-end . . . . .	25
3.9	Work Schedule - Gantt Chart . . . . .	26
<b>4</b>	<b>Results and Discussions</b>	<b>28</b>
4.1	Overview . . . . .	28
4.2	Testing . . . . .	28
4.3	Graphical Analysis . . . . .	30
4.4	Discussion . . . . .	30

<b>5 Conclusion</b>	<b>33</b>
5.1 Conclusion . . . . .	33
5.2 Future Scope . . . . .	34
<b>Appendix A: Presentation</b>	<b>36</b>
<b>Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes</b>	<b>70</b>
<b>Vision, Mission, POs, PSOs and COs</b>	<b>2</b>
<b>Appendix C: CO-PO-PSO Mapping</b>	<b>6</b>

## List of Figures

2.1	Workflow	7
3.1	System Architecture	17
3.2	System Architecture	20
3.3	Login Page	22
3.4	Sign Up Page	22
3.5	Home Page	23
3.6	Use-Case Diagram	24
3.7	Gantt Chart	27
4.1	Login Page Testing using Postman	28
4.2	Registration Page Testing using Postman	29
4.3	Resulting MongoDB outcome	29
4.4	Consecutive VS Code Operations	30
4.5	Output for Input Bridget Jones's Diary	31
4.6	Output for Harry Potter and the Chamber of secrets	31
4.7	Ender's Game	31
4.8	The Witching Hour	31
4.9	Output VS Presence graph of books and recommendation for 4 random users	31
5.1	Demo Example[1]	33
5.2	Demo Example[2]	34

## List of Tables

3.1	Book details . . . . .	20
3.2	Book ratings by users . . . . .	20
3.3	User locations and ages . . . . .	21

# Chapter 1

## Introduction

### 1.1 Background

The project aims to develop a book recommendation system tailored to individual preferences, enhancing the reading experience for users. This section provides insights into current scenarios within the literary world, highlighting the importance of personalized book recommendations.

Current Scenarios:

1. **Information Overload:** With an abundance of books available across various genres and topics, readers often struggle to discover new titles that align with their interests and preferences.
2. **Generic Recommendations:** Traditional bookstores and online platforms often provide generic recommendations based on popular trends or bestseller lists, neglecting the diverse tastes and preferences of individual readers.
3. **User Engagement Challenges:** Without personalized recommendations, users may experience decision fatigue and reduced engagement with book platforms, impacting user satisfaction and loyalty.
4. **Competitive Landscape:** The rise of digital platforms and e-commerce giants has intensified competition in the book industry, necessitating innovative approaches to attract and retain readers.

Importance of the Project:

1. **Personalized User Experience:** By leveraging algorithms and user data, the book recommendation system will deliver tailored recommendations, helping users discover books that resonate with their unique preferences and interests.
2. **Enhanced Engagement and Satisfaction:** Personalized recommendations will enhance user engagement with book platforms, leading to increased satisfaction, longer

browsing sessions, and higher conversion rates.

## 1.2 Problem Definition

Design and implement a book recommendation system that suggests personalized reading choices based on user preferences and past reading history, utilizing collaborative filtering and content-based filtering techniques to enhance user satisfaction and engagement with the platform. The system should also incorporate features for users to provide feedback on recommendations and continuously improve the recommendation algorithms through machine learning and data analysis.

## 1.3 Scope and Motivation

**Scope :** This project aims to develop a personalized book recommendation system utilizing machine learning techniques. It will include a front-end interface for user input, a back-end server for processing, and a machine-learning model on Google Colab for recommendation generation. The system will consider factors like genre preferences, author preferences, and user feedback for accurate recommendations, while also focusing on scalability and performance.

**Motivation:** The rising demand for personalized recommendation systems, especially in literature, drives this project. By providing tailored book suggestions, we aim to enhance user reading experiences and promote exploration of new authors and genres. Ultimately, our goal is to create a user-friendly platform that fosters a deeper engagement with reading and literature in the digital age.

## 1.4 Objectives

1. **Developing a Functional System:** Create a working prototype of the book recommendation system with basic functionality, including user input, recommendation generation, and feedback mechanisms.
2. **Implementing Machine Learning Techniques:** Utilize machine learning algorithms such as collaborative filtering, content-based filtering, or hybrid methods to

generate personalized book recommendations based on user preferences and reading history.

3. **User Interface Design:** Design an intuitive and visually appealing front-end interface for users to input their preferences, view recommended books, and provide feedback.
4. **Back-end Development:** Set up a back-end server to handle user requests, communicate with the machine learning model, and manage the flow of information between the front end and the ML model.
5. **Model Training and Deployment:** Train a machine learning model on Google Colab using relevant datasets and deploy it to generate real-time recommendations.
6. **Testing and Validation:** Conduct thorough testing of the system to ensure functionality, reliability, and accuracy of recommendations. Validate the system's performance under different scenarios and user inputs.

## 1.5 Challenges

The challenges we faced included finding reliable datasets with relevant data such as ratings, genre. Cleaning the datasets also posed some significant challenges as the vast number of datasets had different columns and rating systems(ie 0-5,0-10).

## 1.6 Assumptions

1. **Users provide accurate and representative feedback:** The effectiveness of the recommendation system relies on users providing honest feedback on recommended books, such as ratings or reviews, to improve the accuracy of future recommendations.
2. **User preferences remain consistent over time:** The system assumes that users' reading preferences remain relatively stable over the duration of their interaction with the platform, allowing for effective long-term recommendation strategies.
3. **Sufficient data availability:** The success of machine learning algorithms, including collaborative filtering and reinforcement learning, relies on the availability of a

substantial amount of user-book interaction data to train and optimize recommendation models.

## 1.7 Societal / Industrial Relevance

1. **Promoting Reading Culture:** Encourages diverse reading habits and intellectual enrichment within society.
2. **Enhancing Engagement in Publishing:** Boosts user engagement, satisfaction, and sales for publishers and online bookstores.
3. **Facilitating Education:** Supports academic learning and research by providing relevant reading materials to students and educators.
4. **Enriching Content Discovery:** Improves browsing experience on content discovery platforms by offering personalized book recommendations.
5. **Driving Sales in Retail:** Increases cross-selling opportunities and revenue generation for e-commerce platforms and bookstores.
6. **Fostering Personal Development:** Facilitates lifelong learning and personal growth by suggesting books aligned with individuals' interests and goals.

## 1.8 Organization of the Report

**Chapter 1** :- This chapter delves into the foundational aspects of the project, including the context, issue clarification, extent and inspiration, aims, hurdles, and other relevant considerations that need careful examination prior to commencement.

**Chapter 2** :- This chapter outlines the specifications for the system requirements necessary for the mini project.

**Chapter 3** :- This chapter elaborates on the system's design and architecture, encompassing database structures, machine learning model intricacies, and other relevant details integral to the project's implementation.

**Chapter 4** :- This chapter focuses on the result which were presented during the completion of the project, also the testing and analysis is also briefed here. The accuracy and the testing limitations of each proposed model is also briefed.

**Chapter 5 :-** This chapter is the conclusion of the project and the future enhancements to the project proposed.

# **Chapter 2**

## **Software Requirements Specification**

### **2.1 Introduction**

#### **2.1.1 Purpose**

This document aims to specify the software requirements for LitLink, a Machine Learning based book recommendation aimed at providing personalised reading suggestions to users based on their preferences and requests. This document encompasses the entire LitLink system including a web application, back-end infrastructure and recommendation engine, ensuring a comprehensive understanding of the project's software requirements and architecture.

#### **2.1.2 Product Scope**

LitLink is a ML-based book recommendation system designed to personalise the reading experience for both leisure and non leisure readers. As a large number of younger individuals turn to short form media for entertainment, we aim to encourage longer forms of entertainment such as books and novels as alternatives to the dopamine crazed entertainment systems of the present in an intuitive web application and through use of reinforcement learning algorithms. The books and novels are recommended through inputs provided by the users in the form of previous reading history, genre tags and feedback for the recommendations using a rating system.

### **2.2 Overall Description**

Litlink aims to revolutionize the way readers discover and engage with books by providing a personalized and intuitive platform. By leveraging advanced machine learning techniques and a robust web infrastructure, Litlink seeks to offer tailored recom-

dations that cater to individual tastes and preferences, enhancing the overall reading experience.

### 2.2.1 Product Perspective

Litlink is a new, self contained ML based solution to the growing need for a comprehensive platform dedicated to book discovery and reading personalisation. It operates as a self-contained product with all components including user interfaces, recommendation algorithms, database interactions and backend infrastructure hosted within the web environment. It leverages external databases and APIs for accessing book datasets, the primary functionality and user interactions are confined to the web platform. It is a standalone component in the wider digital landscape focused on delivering personalised recommendations.

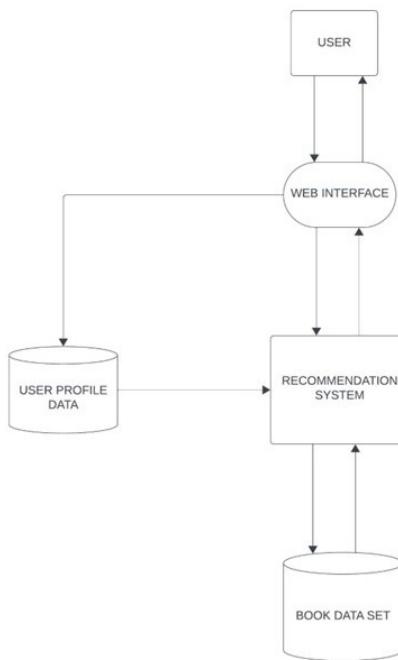


Figure 2.1: Workflow

### 2.2.2 Product Functions

- Recommendation Engine: Analyses user preferences to suggest personalised books.
- User Profile Management: Allows users to customise preferences and track reading history.

- Search and Filtering: Enables precise book discovery based on various criteria.
- Bookshelf Management: Organises and tracks users' book collections
- Review and Rating: Allows users to rate and review books for community feedback

#### **2.2.3 Operating Environment**

The primary concern is to increase the accuracy of the ML algorithm in the programming terminal. The Software operates on a web-based platform which will be powered by a Node and Express JS back-end system and MongoDB database on which the ML based model will be working. There are no operating system constraints, since it is a web-based application.

#### **2.2.4 Design and Implementation Constraints**

The major constraints for the web application is the consistency of the dataset used in the training of the ML model i.e., consistency of the tags associated with the labels of the books, updation of the dataset with time, proper algorithm used to train the model which will lead to efficient results. If the available dataset used is not up to the mark then the output produced by the model would be also inefficient. Proper input and storage of user preferences must be maintained for the normal execution of the proposed algorithm.

#### **2.2.5 Assumptions and Dependencies**

The availability of the proposed data set is a huge dependency on which the whole ML model would be trained, if there is any inconsistency in that part then the whole project would be vulnerable. The input given by the user is also a primary dependency on which the results depend upon, hence the user must be able to properly convey their needs via the available interface. Hence we are assuming that the user is distinct about their goal.

### **2.3 External Interface Requirements**

Litlink's external interface requirements necessitate seamless communication between its various components to ensure a smooth user experience. This entails robust software interfaces that facilitate data flow between the recommendation algorithm, web application,

and external databases. Additionally, secure communication protocols are paramount to safeguard user data during transmission, particularly when interacting with the MongoDB database. These interfaces lay the foundation for efficient user interactions, enabling users to discover personalized book recommendations effortlessly.

### **2.3.1 User Interfaces**

Since it's major focus is to improve algorithm output and accuracy hence the user interface is not a primary concern. But generally in a book recommendation system, users engage with a streamlined interface featuring a homepage that showcases featured books, personalised picks, and popular genres, complemented by a search bar offering autocomplete suggestions. Book listings display essential details like title, author, cover image, and rating, while a detailed book page provides in-depth information, including synopsis and reviews. Users can curate their experience through a Reading List, engage with personalised recommendation widgets, and adjust preferences in their User Profile. The system incorporates feedback mechanisms for continuous improvement and a navigation bar for easy access across sections. Notifications keep users informed of updates, and a dedicated Help section offers support, ensuring a comprehensive and user-friendly environment for exploring and discovering recommended books.

### **2.3.2 Hardware Interfaces**

In our book recommendation system there might not be a direct hardware interface for the software itself. However there could be potential hardware interfaces to be considered as it may use cloud based services for back end interface.

### **2.3.3 Software Interfaces**

The project prioritises developing an efficient recommendation algorithm before extending it to the web application. Initially, we focused on building the algorithm using TensorFlow, leveraging user behaviour and book details to generate personalised recommendations from datasets like Kaggle. Communication occurs via RESTful APIs, facilitating seamless data flow. Once proven effective, the algorithm integrates into a web application using Flask and front-end technologies. This phased approach ensures a refined recommendation

system before web deployment, enhancing overall user experience. Constraints like global data areas may optimise performance.

#### **2.3.4 Communications Interfaces**

The web application relies on standard HTTP/HTTPS protocols for user interactions via web browsers. Network server protocols facilitate data exchange among the web application, the machine learning model, and external components. User input, collected through electronic forms, requires proper message formatting, typically JSON or other standard formats. Security measures, including encryption, safeguard sensitive user data during communication, particularly when accessing the MongoDB database.

### **2.4 System Features**

Introducing Litlink's core functionalities, the system features are meticulously designed to elevate users' reading experiences. These customizations not only enhance user engagement but also foster a deeper connection between readers and the platform, ultimately enriching their overall reading journey.

#### **2.4.1 Personalised Reinforcement Recommendation**

##### **Description and Priority**

The "Personalised Reinforcement Recommendation" feature aims to provide individualised suggestions for reinforcing user engagement. Given its significance, this feature is designated as a High priority.

##### **Stimulus/Response Sequences**

**Stimulus 1:** User engages with the system by accessing the recommendation interface.

**Response 1:** System analyses user behaviour and preferences.

**Response 2:** System generates personalised recommendations based on the analysis.

**Stimulus 2:** User interacts with the recommended content.

**Response 3:** System observes user engagement with the suggested content.

**Response 4:** System refines future recommendations based on observed user behaviour.

## **Functional Requirements**

**REQ-1:** The system must employ machine learning algorithms to analyse user behaviour and preferences.

**REQ-2:** A recommendation engine must be implemented to generate personalised content suggestions.

**REQ-3:** User profiles must be created and updated to store individual preferences.

**REQ-4:** The system should dynamically adjust recommendations based on real-time user interactions.

**REQ-5:** Error handling mechanisms must be in place to manage unexpected scenarios gracefully.

**REQ-6:** The system must prioritise content recommendations by considering the benefit, penalty, cost, and risk factors defined in the feature's priority assessment.

**REQ-7:** Regular updates to the recommendation algorithms should be scheduled for continuous improvement.

### **2.4.2 User Profile Management**

#### **Description and Priority**

User Profile Management enables users to customise preferences and track reading history, fostering personalised literary experiences while prioritising data privacy and security. It empowers users to tailor their reading journey and receive relevant book recommendations based on their interests and habits.

#### **Stimulus/Response Sequences**

**Stimulus 1:** User accesses the profile settings.

**Response 1:** System displays options for customization.

**Response 2:** User customises preferences (e.g., genres, favourite authors).

**Stimulus 2:** User interacts with books (e.g., reads, adds to favourites).

**Response 3:** System updates reading history and preferences.

## **Functional Requirements**

- REQ-1:** Allow users to create and update profiles.
- REQ-2:** Provide options for customising preferences.
- REQ-3:** Track reading history and user interactions with books.
- REQ-4:** Ensure user data privacy and security.
- REQ-5:** Implement user authentication mechanisms.

### **2.4.3 Bookshelf Management**

#### **Description and Priority**

Bookshelf Management organises and tracks users' book collections, providing a medium-priority function with benefits in user organisation and tracking, balanced against moderate penalties, costs, and risks.

#### **Stimulus/Response Sequences**

- Stimulus 1:** User accesses the bookshelf feature.
- Response 1:** System displays the user's book collection.
- Response 2:** User adds, removes, or categorises books.
- Stimulus 2:** User interacts with the bookshelf.
- Response 3:** System updates the bookshelf based on user actions.

## **Functional Requirements**

- REQ-1 :** Provide options for adding, removing, and categorising books.
- REQ-2:** Allow users to create multiple bookshelves or categories.
- REQ-3:** Sync bookshelf across devices if applicable.
- REQ-4:** Implement sorting options (e.g., by title, author, genre).
- REQ-5:** Enable exporting or sharing bookshelf data.

## **2.5 Other Nonfunctional Requirements**

### **2.5.1 Performance Requirements**

#### **Website Responsiveness :-**

**Page Load Time :-** Individual web pages should load within a few seconds on a stable internet connection. This ensures a smooth browsing experience and minimizes user frustration.

**Search Response Time :-** User search queries should return relevant recommendations within a few seconds. This responsiveness keeps users engaged and encourages them to explore further recommendations.

**Recommendation Generation Time :-** Recommendations should be generated and displayed to the user within a few seconds of initiating a search or entering preferences. This ensures users don't experience delays while waiting for suggestions.

#### **System Scalability :-**

The website should be able to handle a concurrent user load of  $X$  users (specify a target number based on user projections) without significant performance degradation. This ensures the website can handle peak traffic periods without compromising user experience.

#### **Rationale :-**

These performance requirements prioritize a fast and responsive user experience. Fast page load times keep users engaged and prevent them from abandoning the website due to frustration. Quick search response times maintain user interest and encourage them to explore the recommendations offered by the platform. Finally, timely generation of recommendations allows users to discover new books efficiently without waiting for suggestions. These factors all contribute to a more enjoyable and productive experience for the website's users.

### **2.5.2 Safety Requirements**

#### **User Data Privacy :-**

Ensuring secure storage of user information with encryption and access controls, granting users control over data collection, and offering clear opt-in/out mechanisms for recommendations are crucial. Accompanied by a transparent privacy policy articulating data usage practices, including what data is collected and shared, these measures demonstrate commitment to user privacy. Regular updates to security and policy revisions maintain alignment with regulations, fostering trust and ethical user relationships.

#### **Content Moderation:-**

Filtering out inappropriate content such as hate speech and violence is imperative, alongside age-gating for mature content with conspicuous warnings. Providing mechanisms for users to report inappropriate recommendations bolsters user safety and community standards. By implementing these measures, platforms can cultivate safer and more inclusive environments, prioritizing user well-being and upholding ethical standards of content dissemination.

#### **Security :-**

Ensuring secure user authentication and authorization processes is fundamental, bolstering protection against unauthorized access to both user data and website functionality. Regular security audits and updates play a crucial role in identifying and addressing vulnerabilities, maintaining robust defenses against potential breaches. By prioritizing these measures, platforms can instill confidence in users regarding the safety and integrity of their data, fostering trust and loyalty within the user community.

### **2.5.3 Security Requirements**

#### **Data Security**

**User Data Encryption :-** All user data (including passwords, reading history, preferences) must be encrypted at rest and in transit using industry-standard protocols.

**Data Access Controls :-** Implement access controls to restrict access to user data based on user roles and permissions. Only authorised personnel should be able to access sensitive user information.

## User Authentication

**Strong Password Hashing :-** Implement strong password hashing algorithms to protect user credentials.

## Security Incident Response

Developing a documented security incident response plan is essential to define clear procedures for handling security breaches and data leaks effectively. These requirements are critical to ensuring a secure platform for both user data and website functionality. It's crucial to consult with relevant security professionals to tailor these measures to your specific needs and regulatory landscape, ensuring comprehensive protection against potential threats. By proactively addressing security concerns and implementing robust protocols, platforms can mitigate risks and safeguard user trust and confidence in their services.

### 2.5.4 Software Quality Attributes

#### Usability

**Ease of Use :-** The website should be easy to navigate and use for all users.

**Task Completion Time :-** Users will be able to complete core tasks (finding a book recommendation, adding a book to their wishlist) with a least amount of time.

**Learning Curve :-** The website interface will be learnable within a single use session. This can be assessed through user feedback surveys after their initial interaction with the platform.

#### Reliability

**Uptime :-** The website will be available to users at least 99.5% of the time during peak hours (e.g., evenings and weekends). This ensures consistent access to the platform for users, promoting reliability and trust in the service.

**Mean Time Between Failures (MTBF) :-** The website will strive for an MTBF of at least 24 hours to minimise service disruptions.

**Data Integrity :-** User data (reading history, preferences) will be accurate and consistent across sessions.

## Performance

**Page Load Time :-** Individual web pages will load within a few seconds on a stable internet connection.

**Search Response Time :-** Search queries will return results within a few seconds.

**Recommendation Generation Time :-** Recommendations will be generated and displayed to the user within a few seconds of initiating a search or entering preferences.

## Maintainability

**Code Readability :-** The codebase will adhere to a consistent coding style and utilise clear naming conventions to facilitate future maintenance.

**Modular Design :-** The system will be designed with well-defined modules that can be easily modified or extended without impacting other functionalities.

**Unit Test Coverage :-** At least 80% of the core functionalities be covered by unit tests to ensure code stability during maintenance.

## Security

**Data Encryption :-** User data (passwords, reading history) will be encrypted at rest and in transit using industry-standard encryption protocols.

**Secure Login :-** User authentication will utilise a strong password and offer other added security.

# Chapter 3

## System Architecture and Design

### 3.1 System Overview

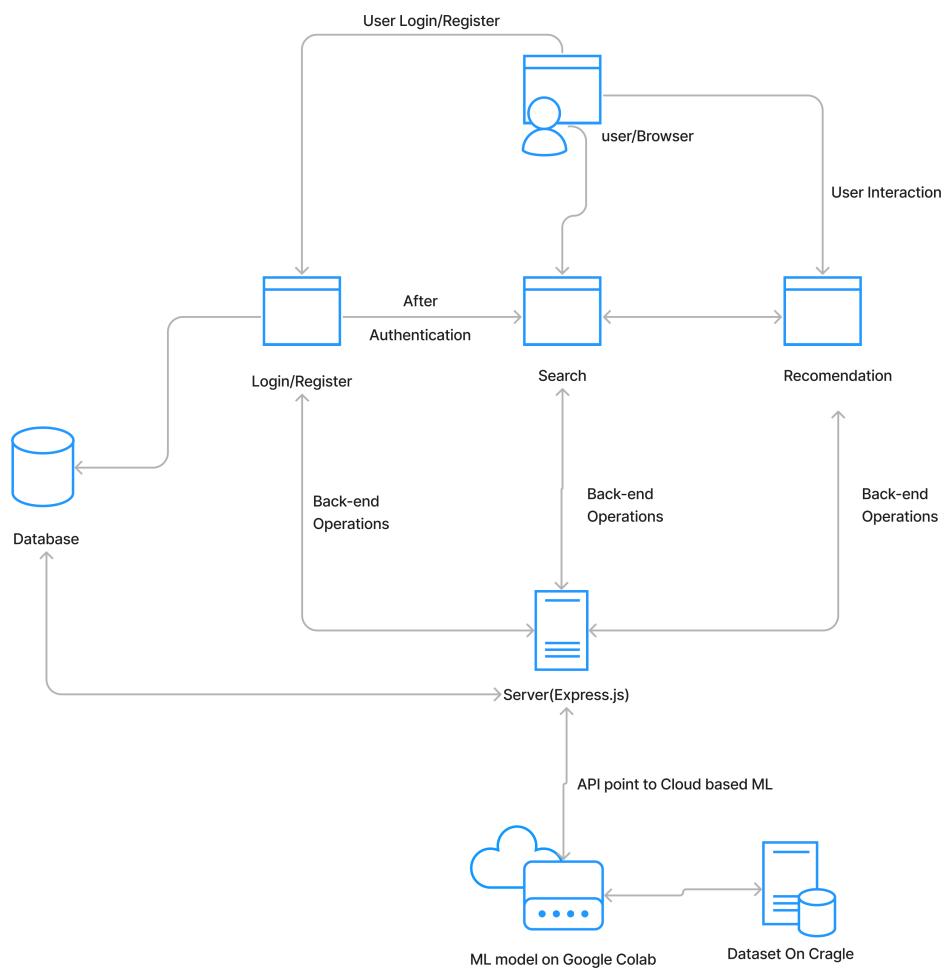


Figure 3.1: System Architecture

This section provides a detailed overview of the system architecture, highlighting the interaction between various components and their roles. The architecture integrates user interactions, back-end operations, and a machine learning model hosted on Google Colab to deliver personalized recommendations.

## 1. User Interaction Layer

- **User/Browser:** Represents the front-end interface where users interact with the system through a web browser, performing actions like logging in, registering, searching for content, and receiving recommendations.

## 2. Authentication Process

- **User Login/Register:** Users log in or register by providing their credentials.
- **Authentication Module:** The server validates the credentials against the database, granting access to the system's features upon successful authentication.

## 3. Database

- **Database:** Stores all necessary data, including user information, search history, and other relevant data. It interacts with back-end operations for data retrieval and storage.

## 4. Back-end Operations

- **Login/Register Operations:** Handle the validation and processing of user credentials.
- **Search Operations:** Manage search queries by retrieving relevant data from the database, processing it, and returning appropriate results.
- **Recommendation Operations:** Generate personalized recommendations based on user interactions and data analysis, interfacing with the machine learning model.

## 5. Server Layer

- **Server (Express.js):** The core of the back-end, built using Express.js, handles HTTP requests from the user/browser, processes these requests, and connects the application with the machine learning model via API endpoints.

## **Workflow Description**

**(a) User Registration/Login:**

- Users access the application via a web browser, register or log in, and the server authenticates their credentials by interacting with the database.

**(b) User Interaction:**

- Authenticated users search for content or interact with the system to receive recommendations, handled by specific back-end operations.

**(c) Search Operation:**

- Users input search queries via the browser. The server processes these requests, retrieves relevant data from the database, and sends back results.

**(d) Recommendation Operation:**

- Based on user activity and historical data, the server invokes recommendation operations, gathers necessary data, and sends it to the ML model via the API.

**(e) Machine Learning Processing:**

- The ML model on Google Colab processes the data received through the API, generates recommendations, and sends them back to the server.

**(f) Displaying Results:**

- The server processes the recommendations from the ML model and sends them to the user's browser, displaying personalized suggestions.

### **3.2 Architectural Design**

### **3.3 Dataset identified**

The dataset used for the model is in the .csv format and contains important data such as Author name, title, ISBN, ratings. The dataset is a combination of

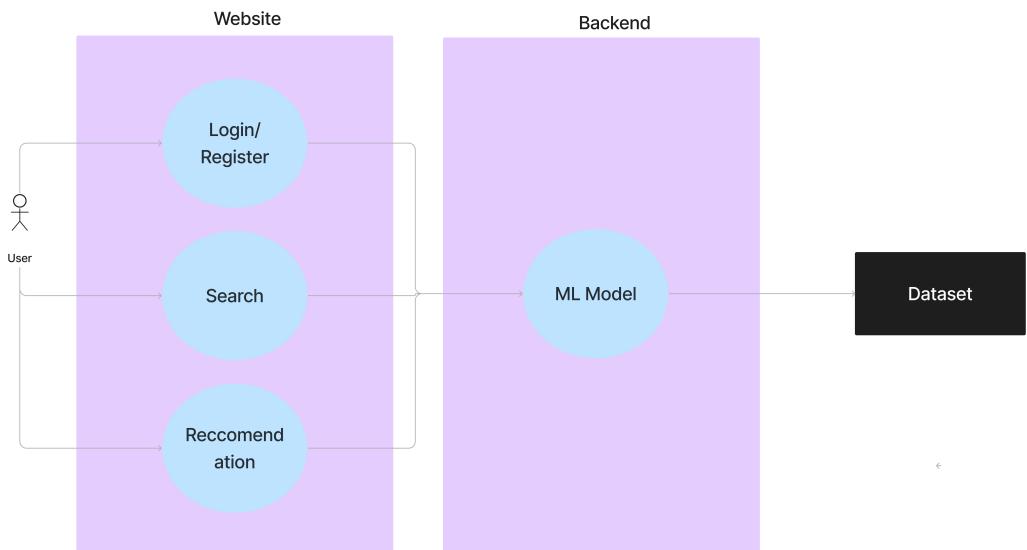


Figure 3.2: System Architecture

various datasets found on kaggle.com for the best combination of new books and large number of ratings.

ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada

Table 3.1: Book details

User-ID	ISBN	Book-Rating
276725	034545104X	0
276726	0155061224	5
276727	0446520802	0

Table 3.2: Book ratings by users

### 3.4 Proposed Methodology/Algorithms

The K-Nearest Neighbors (KNN) algorithm stores all data points during training, calculates distances to find the 'k' nearest neighbors, and assigns the most common

User-ID	Location	Age
1	nyc, new york, usa	NULL
2	stockton, california, usa	18
3	moscow, yukon territory, russia	NULL
4	porto, v.n.gaia, portugal	17

Table 3.3: User locations and ages

label for classification or the average value for regression, but its performance depends on choosing the right 'k' value and it can be computationally expensive with large datasets due to the need to compute distances for every data point, while also suffering from the curse of dimensionality.

### KNN Algorithm

**1.Training Phase:** Store all training data points and their corresponding labels.

**2.Prediction Phase:**

- For each new data point, calculate distances to all other data points in the training set.
- Select the 'k' nearest neighbors based on these distances.
- For classification, take a majority vote among the labels of the 'k' nearest neighbors; for regression, calculate the average of their target values.
- Assign the most common label or average value to the new data point.

**3.Choosing 'k':** Select a suitable 'k' value based on dataset characteristics, often using cross-validation to optimize performance.

### 3.5 User Interface Design

The user interface design (wireframe designs) can be highlighted in this section. The figures titles should be in chronological order and self-explanatory.

## Front End

The Front End was created by using HTML and CSS. It is mainly consist of three pages Login Page, Sign Up Page, and Home Page.

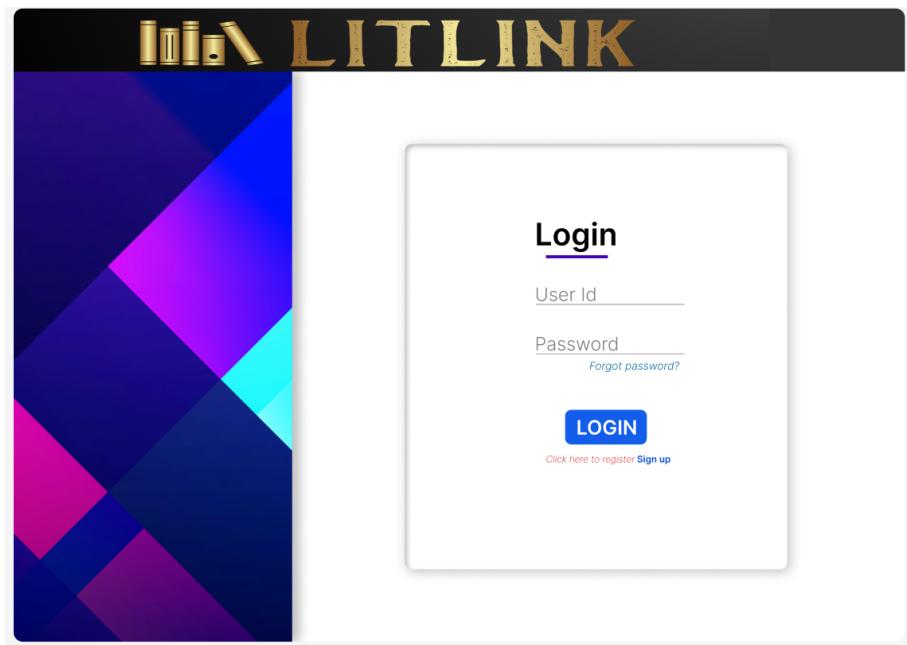


Figure 3.3: Login Page

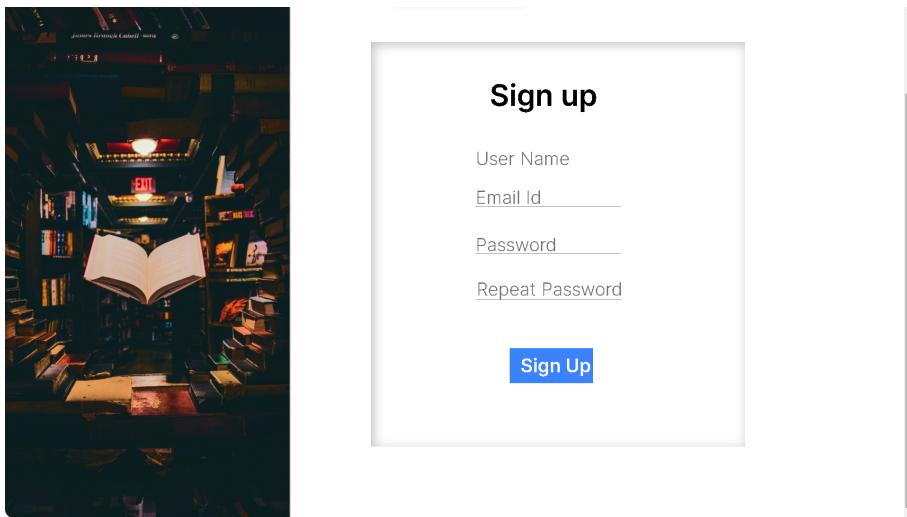


Figure 3.4: Sign Up Page

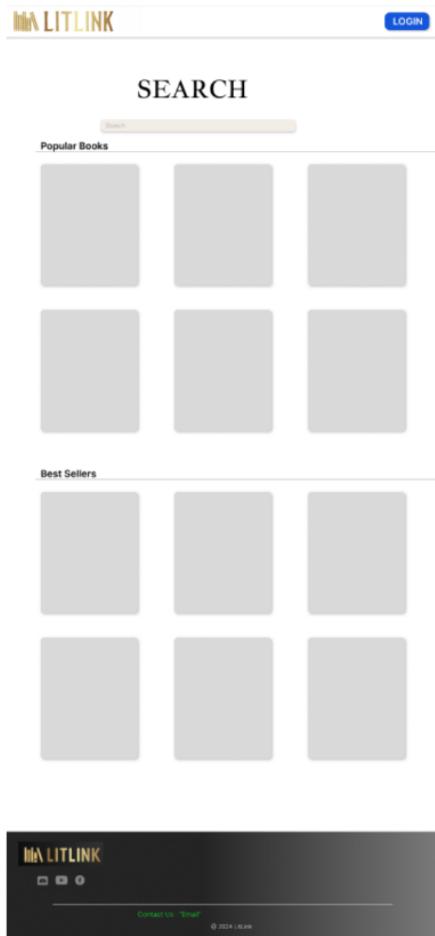


Figure 3.5: Home Page

### 3.6 Database Design

The database being used in the mini project is MongoDB, possibly the Atlas version. This type of NoSQL will provide much needed flexibility and will ease the flow of data retrieval and presentation to the end user.

Also the unexpected crashes could also be avoided hence improving the performance of the complete application.

### 3.7 Description of Implementation Strategies

**Python Libraries:** We utilize popular machine learning libraries such as sci-kit-learn and TensorFlow for implementing recommendation algorithms.

**Methodology:** We employ collaborative filtering techniques, such as matrix fac-

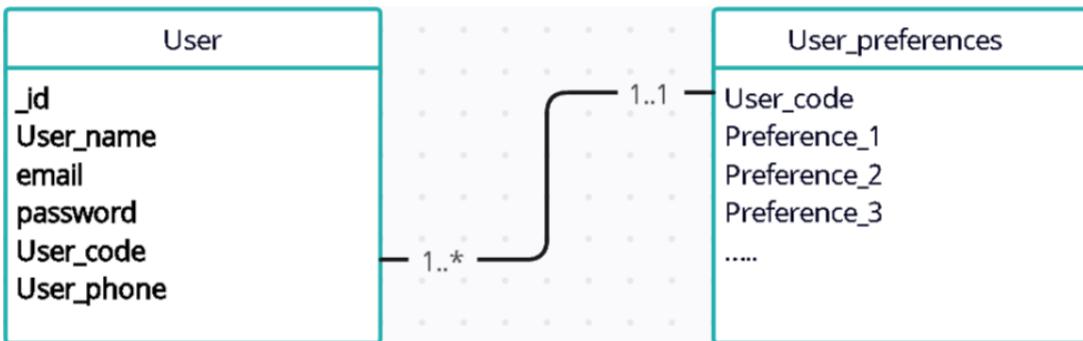


Figure 3.6: Use-Case Diagram

torization or nearest neighbor methods, to generate personalized book recommendations based on user behavior and preferences. These libraries are also used to evaluate the effectiveness of the recommendation engine.

### 3.8 Module Division

This section describes the different modules involved in this project and a small description of the same is expected. This section ends with the information of which module is assigned to each project member.

#### 3.8.1 K-Nearest Neighbors (KNN) Model

In this project, we implemented a K-Nearest Neighbors (KNN) algorithm to create a machine learning model for book recommendations. The KNN algorithm classifies a data point based on the majority class of its  $k$  nearest neighbors, determined using distance metrics such as Euclidean distance. Key steps in our approach included data preprocessing, training, and hyperparameter tuning to find the optimal number of neighbors ( $k$ ). The model was evaluated using metrics like accuracy for classification or mean squared error for regression.

Our KNN model achieved 0.65 accuracy on the test set, demonstrating its effectiveness at recommending books. The choice of  $k$  was critical, with an optimal  $k$  value of 6. This project highlights the KNN algorithm's simplicity and interpretability, proving its utility in real-world machine learning applications.

### **3.8.2 Front End**

The user is first shown a login page that features a clean and straightforward design, centered around a white box that contains all the essential elements for user authentication. Users can enter their credentials into two fields: one labeled "User ID" for the email or user ID, and another labeled "Password" for the password. Below the password field, there is a "Forgot password?" link for users who need to reset their credentials. The login process is facilitated by a prominent blue "LOGIN" button. Additionally, beneath the login button, there is a prompt for new users to create an account, with a clickable "Sign up" link for registration. After entering the credentials it checks whether it is in the users Data Base. if true the user is redirected to the home page else shows an error in the login page. if the user clicks on the Sign Up button it redirects the user to the sign page.

The Sign up Page contains a "User Name" input field where the user can enter their desired username. An "Email ID" input field is provided for the user's email address, which is used for account identification and recovery. There is also a "Password" input field where the user can enter their desired password, with the characters usually obscured for privacy. A "Repeat Password" input field is included where the user needs to re-enter their password to confirm it, ensuring the password is entered correctly. Finally, there is a "Sign Up" button that submits the form data to create a new account. When clicked, this button sends the information to the user Data Base and stores it there.

The Home Page contains a "search" input field where the user can enter the Book that they want to search and it redirects it to the search page that shows the results of the search. On the Home Page, it shows a list of books in categories like Best Sellers, Popular Books, etc. The Home Page also contains a Login button to Log out or Log in to another account.

### **3.8.3 Back-end**

The back-end of the application is structured using the Model-View-Controller (MVC) pattern. With the server connection successfully established, controllers

and routes have been created to handle the application's logic and API endpoints. Postman is utilized for testing the API, allowing for the posting and retrieving of values. The database setup includes a MongoDB server connected to both Atlas and Compass, ensuring seamless database management and connectivity. Sensitive information is safeguarded through the use of a .env file, keeping credentials and other private data secure.

In the models directory, a schema was created to define the structure of the data. Sample registration data was inserted using Postman to ensure the API functions correctly. The application checks for email existence to prevent duplicate registrations. Password encryption is handled using bcrypt, which hashes passwords before storing them in the database. For user authentication and authorization, JSON Web Tokens (JWT) are used to generate secure tokens. The login verification process includes password comparison to authenticate users. Additionally, registration inputs are validated using ZOD, providing robust error handling to ensure data integrity.

The integration of the saved pickle file from Google Colab into the backend involved several key steps. Initially, a machine learning model was trained in Google Colab and serialized into a pickle file ('model.pkl'). This file was then transferred to the backend server, where it was loaded using Python's 'pickle' library during server startup or the first relevant request. New API endpoints were created to utilize the model, allowing clients to send input data, which the backend processed and passed to the model for predictions. The endpoints, tested with Postman, ensured correct functionality and performance. Comprehensive error handling and optimizations were implemented to manage invalid inputs and maintain efficient server response times. This integration enabled the backend to leverage advanced machine learning capabilities, providing real-time predictions and data analysis through API calls.

### 3.9 Work Schedule - Gantt Chart

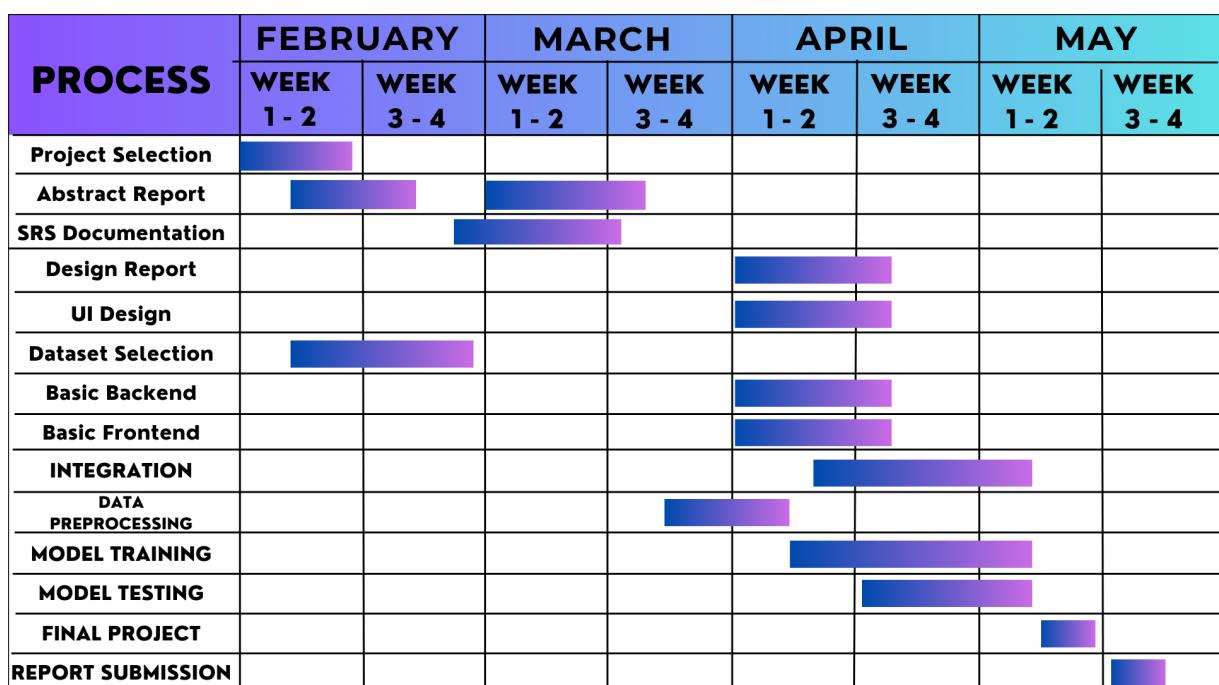


Figure 3.7: Gantt Chart

# Chapter 4

## Results and Discussions

### 4.1 Overview

The Collaborative filtering method was successfully applied to the project and also the user safety and privacy features were showcased.

### 4.2 Testing

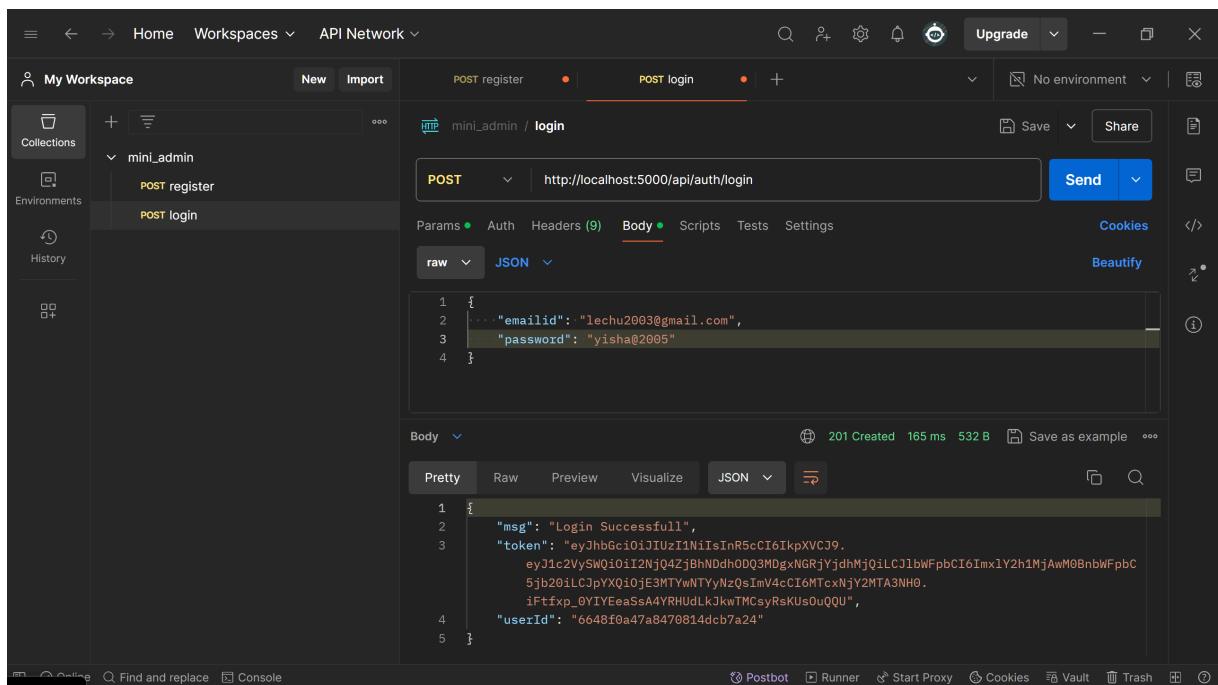


Figure 4.1: Login Page Testing using Postman

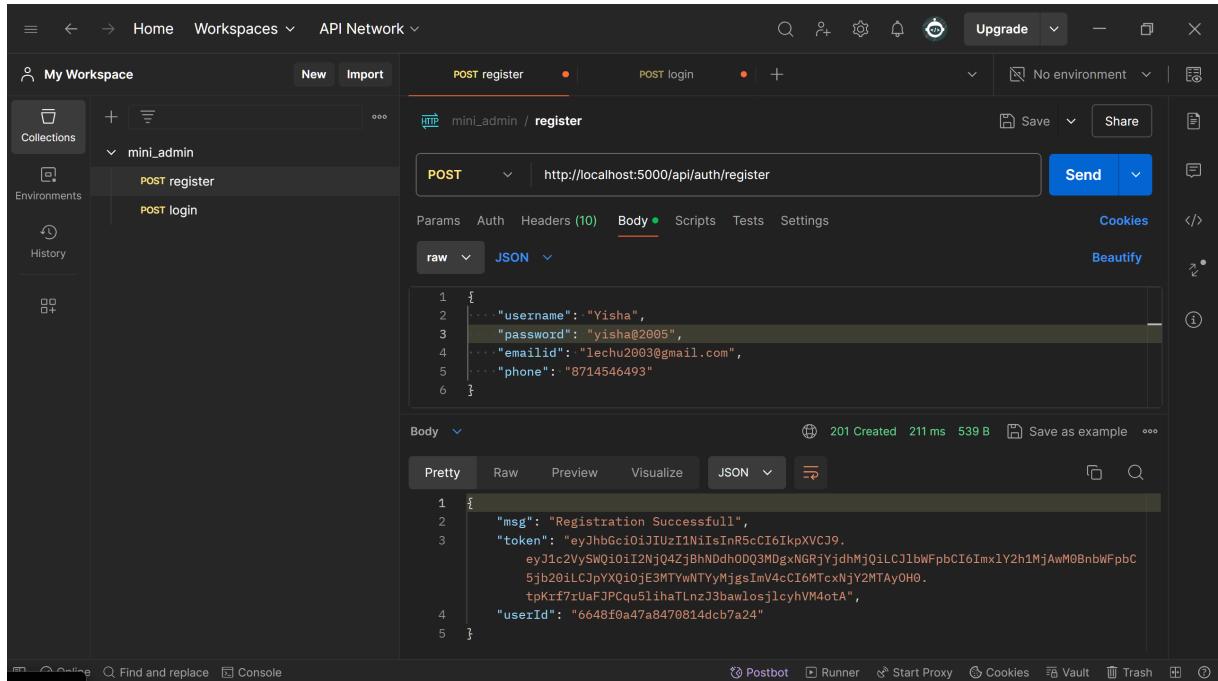


Figure 4.2: Registration Page Testing using Postman

The screenshot shows the MongoDB Compass interface. The left sidebar lists databases ('admin', 'config', 'litlink\_db', 'local', 'test') and collections ('users'). The main area shows the 'users' collection with three documents:

- Yisha:** \_id: ObjectId('66216424fb6c3c25e96b4bf'), username: "Yisha", password: "\$2b\$10\$xc4E0k3ynG72f6TCP1/y.GbsbrBgYLuiptTxwsm.q.4BI02jLR5K", emailId: "lechu2003@gmail.com", phone: "8714546493", \_\_v: 0
- Ghanu:** \_id: ObjectId('66216424fb6c3c25e96b4bf'), username: "Ghanu", password: "\$2b\$10\$xc4E0k3ynG72f6TCP1/y.GbsbrBgYLuiptTxwsm.q.4BI02jLR5K", emailId: "12345@gmail.com", phone: "12345", \_\_v: 0
- Chayan:** \_id: ObjectId('6648f0a47a8470814dcdb7a24'), username: "Chayan", password: "\$2b\$10\$FGR3Z.A8ovAfya.7B0XveUCDyQY9CPQg9yKPLwQotYqvdsSeRhsa", emailId: "lechu2003@gmail.com", phone: "8714546493", \_\_v: 0

Figure 4.3: Resulting MongoDB outcome

The screenshot shows the Visual Studio Code interface. The terminal window at the top right displays the output of running `node server.js`, which includes messages about nodemon starting the server, a successful database connection, and a warning about a phone number being too short. The Explorer sidebar on the left shows a file tree with a folder named `SERVER` containing subfolders like `controllers`, `middlewares`, and `models`, along with files such as `controller.js`, `user_model.js`, and `server.js`. The `server.js` file is currently selected. The bottom status bar shows details like line 10, column 19, and file type JavaScript.

Figure 4.4: Consecutive VS Code Operations

### 4.3 Graphical Analysis

The graphs below illustrate various users requesting books they have previously read. From these graphs, several insights can be derived. Firstly, the accuracy for recommending books in a series is higher compared to non-series books. Secondly, the number of reviews a book has significantly influences recommendations, as niche books with fewer reviews are less frequently recommended. Lastly, the recommendation system provides different outcomes even when the number of books in a series exceeds the number of neighbors considered by the recommendation algorithm.

### 4.4 Discussion

The project results included a demonstration of book recommendations based on the user's past interactions with the model. Only collaborative filtering was applied;

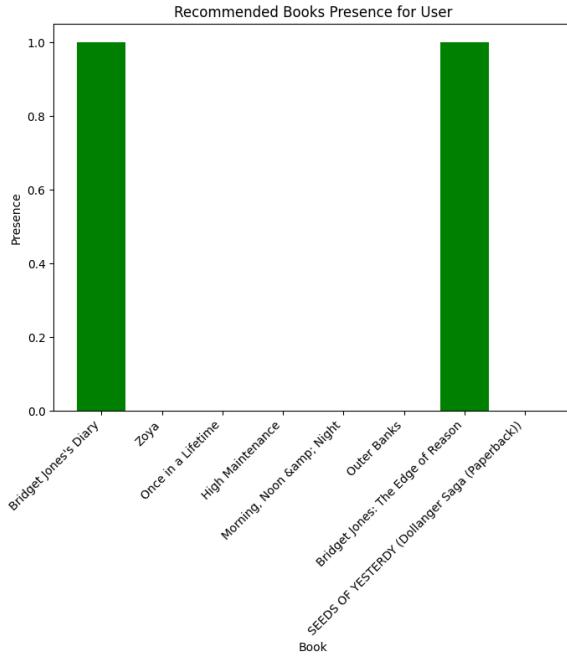


Figure 4.5: Output for Input Bridget Jones's Diary

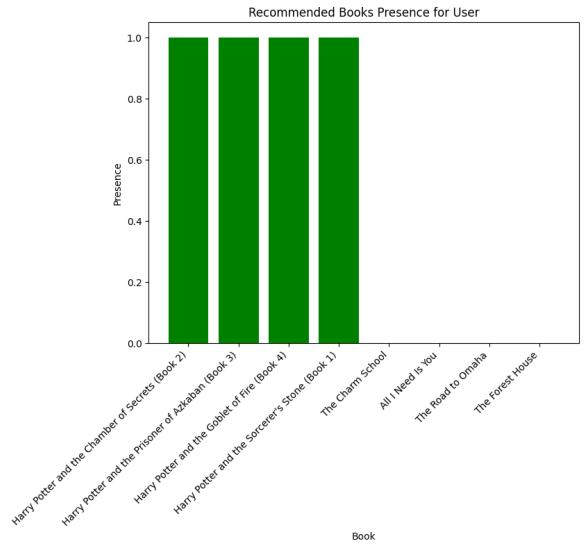


Figure 4.6: Output for Harry Potter and the Chamber of secrets

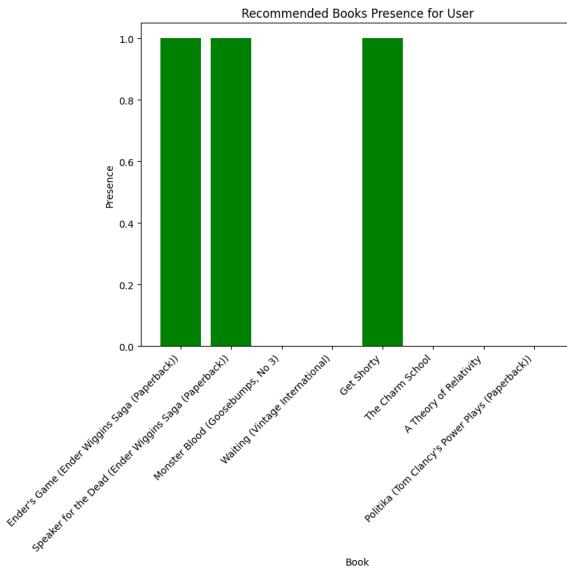


Figure 4.7: Ender's Game

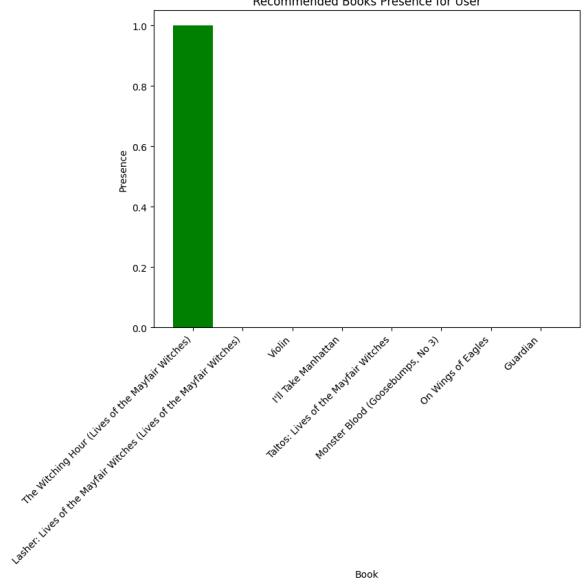


Figure 4.8: The Witching Hour

Figure 4.9: Output VS Presence graph of books and recommendation for 4 random users

although we proposed to implement content-based and hybrid filtering algorithms, these were not presented due to limited resources.

The deviation from expected results was primarily due to the limited size and ac-

cessibility of the available dataset, which could lead to unexpected outcomes.

# Chapter 5

## Conclusion

### 5.1 Conclusion

The project was successfully completed and all it's features including the recommendation model, front-end design and the back-end user security measures(Hashing, Tokenization, ZOD) were successfully demonstrated.

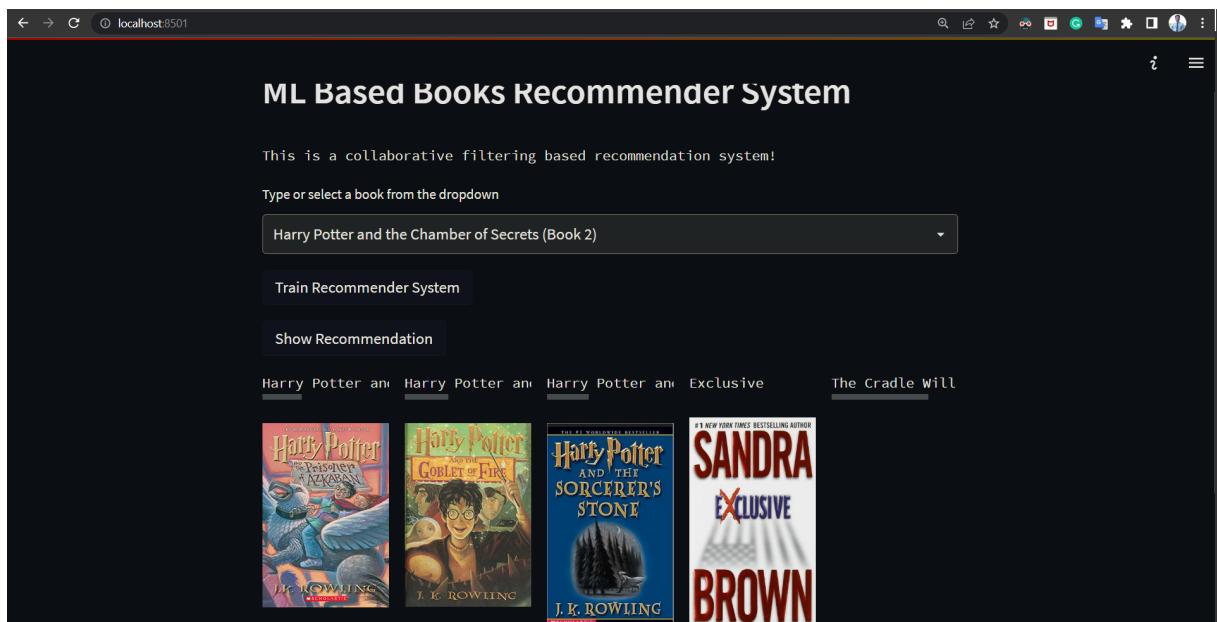


Figure 5.1: Demo Example[1]

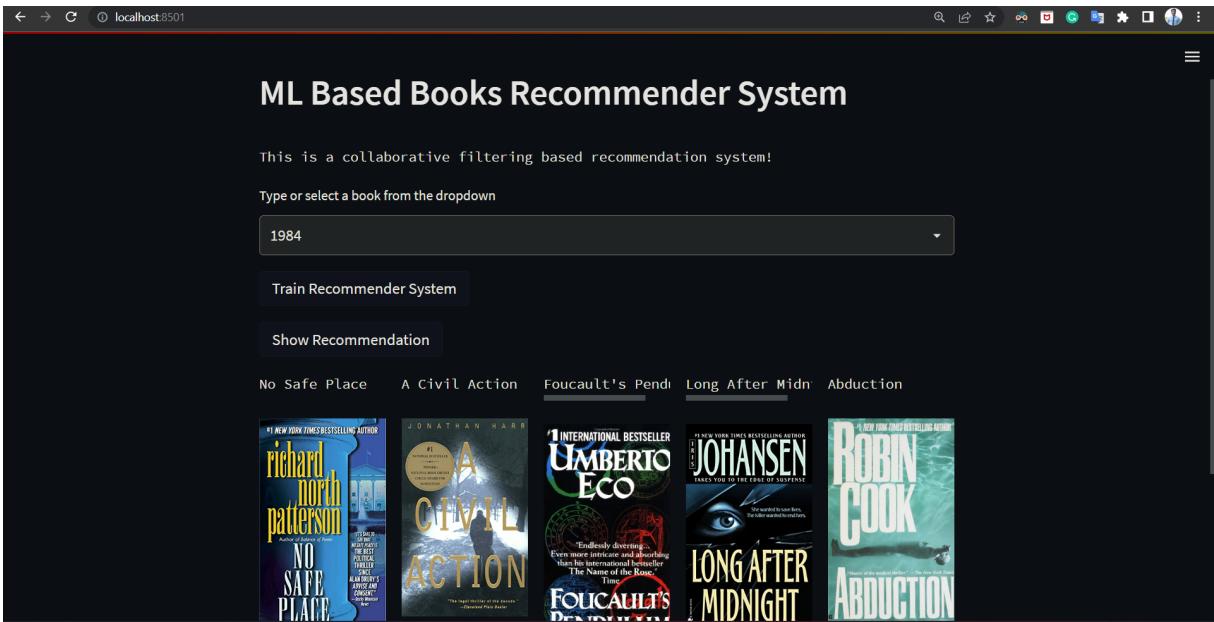


Figure 5.2: Demo Example[2]

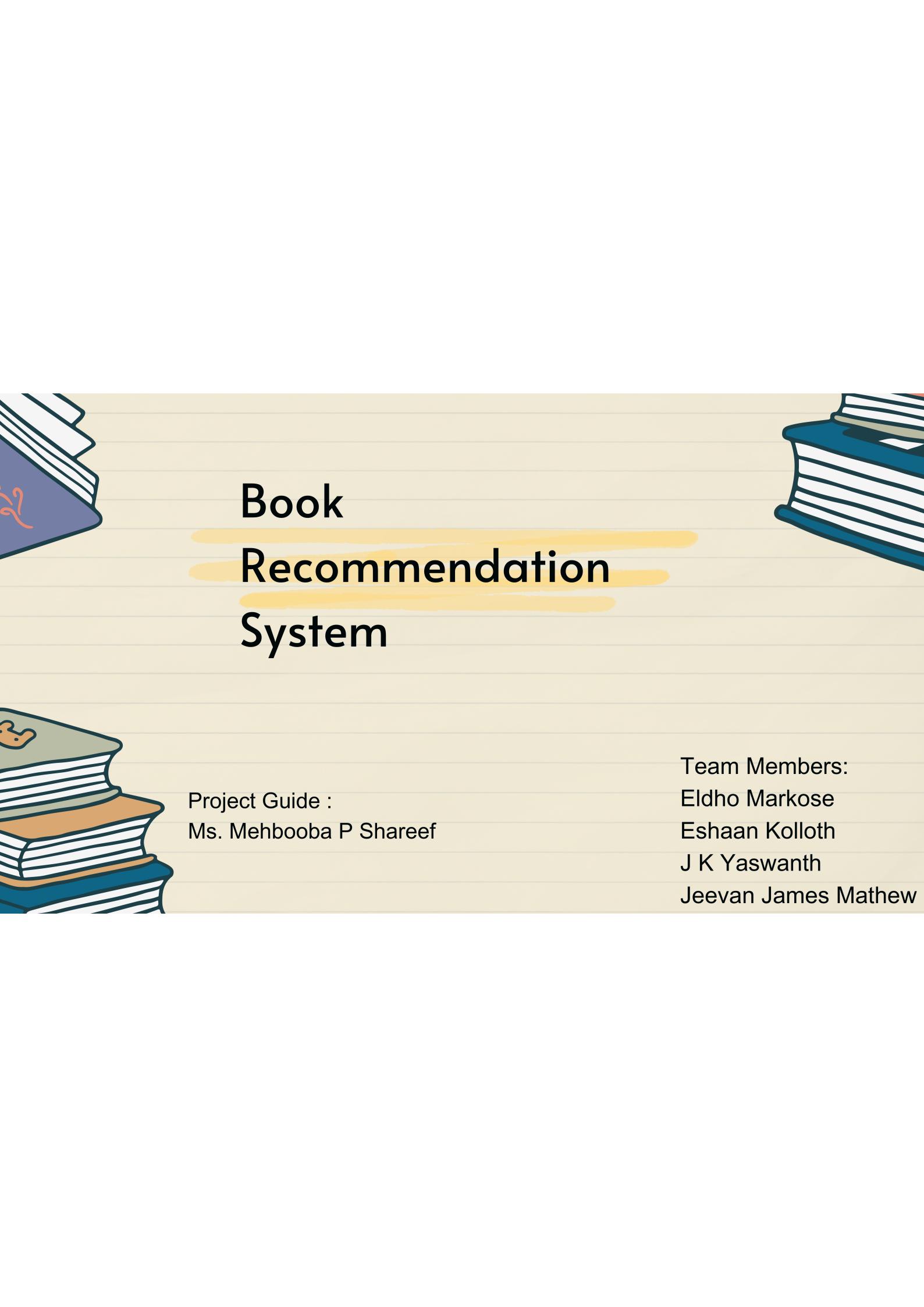
## 5.2 Future Scope

This project contains numerous features that could be expanded upon, making it highly appealing to bibliophiles. Some of the potential enhancements include incorporating more precise hybrid-based machine learning algorithms to improve the accuracy and personalization of book recommendations. By combining different algorithmic approaches, the system can better understand and cater to individual reading preferences. Additionally, introducing an agent-environment system would allow users to simulate various book preference environments. Through reinforcement learning, users could observe and compare different reading scenarios, gaining insights into where their reading journey might lead. This interactive approach could enhance user engagement and provide a more immersive reading experience.

## Bibliography

- [1] scikit-learn python library documentation - <https://scikit-learn.org/0.21/documentation.html>
- [2] Tensorflow documentation - <https://www.tensorflow.org>
- [3] MongoDB documentation - <https://www.mongodb.com/docs/>
- [4] Expressjs documentation - <https://expressjs.com/en/5x/api.html>
- [5] React.js documentation - <https://react.dev/learn>
- [6] HTML Development Material :- <https://www.w3schools.com/html/default.asp>
- [7] npm Documentation :- <https://docs.npmjs.com/>

## **Appendix A: Presentation**



# **Book Recommendation System**

Project Guide :  
Ms. Mehbooba P Shareef

Team Members:  
Eldho Markose  
Eshaan Kolloth  
J K Yaswanth  
Jeevan James Mathew

# Contents

- Introduction
- Problem Definition
- Objectives
- Scope and Relevance
- System Design
- Dataset
- Work Division
- Software Requirements
- Hardware Requirements
- Result
- Conclusion
- Future Enhancement
- References

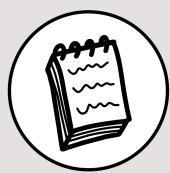
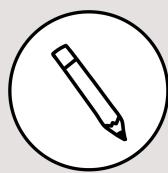


# Introduction

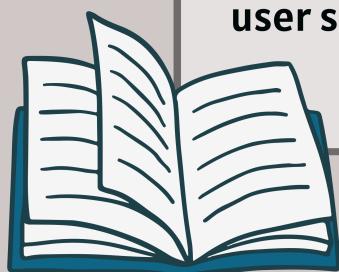
- **Proposal:** Addressing the decline in leisure reading in temporary society dominated by short-form social media content
- **Solution:** Machine learning-based book recommendation system integrated into a user-friendly website
- **Target Audience:** Catering to both casual readers and book enthusiasts



## Problem Definition



**Design and implement a book recommendation system that suggests personalized reading choices based on user preferences and past reading history, utilizing collaborative filtering and content-based filtering techniques to enhance user satisfaction and engagement with the platform.**



# Objectives

## Objective 1

Provides user with a book recommendation system with machine learning techniques to personalize the book recommendation according to user preference and history



## Objective 2

Provide a easy way to navigate user interface that is intuitive that clearly defines the features available to the user



# Scope and Relevance



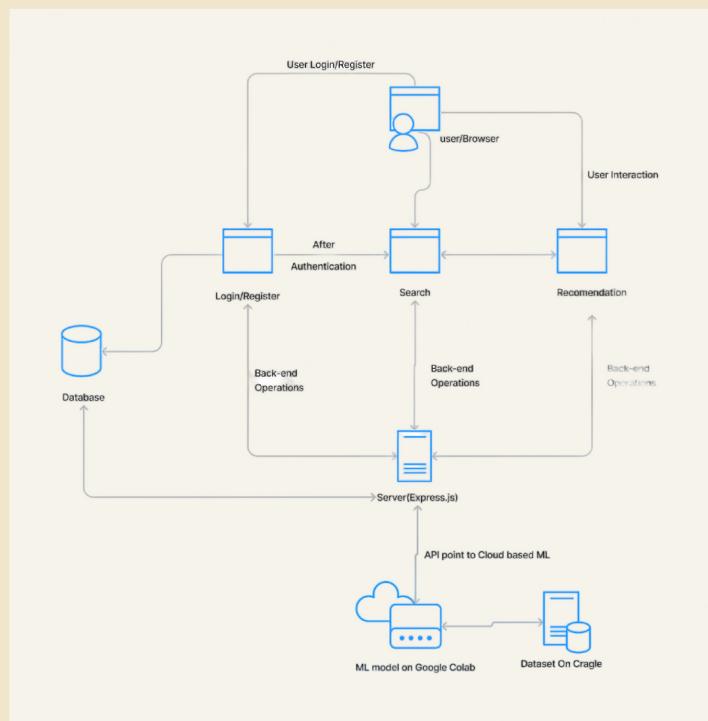
- LitLink is a ML-based book recommendation system designed to personalise the reading experience for readers.
- As a large number of younger individuals turn to short form media for entertainment
- we aim to encourage longer forms of entertainment .
- The books and novels are recommended through inputs provided

# System Design



- **User Authentication:** New users' data is stored in MongoDB Atlas, while existing users' information is verified.
- **Database:** MongoDB Atlas stores user data and improved recommendations.
- **Backend:** ExpressJS facilitates communication between the ML model, database, and users.
- **ML Model:** Trained on Kaggle using Google Colab, it provides personalized book recommendations.
- **User Interaction:** Users can search for new books recommendations based on preferences.

# ARCHITECTURE DESIGN



# DATASETS

The dataset used for the model is in the .csv format and contains important data such as Author name, title, ISBN, ratings. The dataset is a combination of various datasets found on kaggle.com for the best combination of new books and large number of ratings.

ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL	Image-URL	Image-URL
0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	<a href="http://images.amazon.com/images/P/0195153448.01.THUMBZZZ.jpg">http://images.amazon.com/images/P/0195153448.01.THUMBZZZ.jpg</a>	<a href="http://images.amazon.com/images/P/0195153448.01.MZZZZZZZ.jpg">http://images.amazon.com/images/P/0195153448.01.MZZZZZZZ.jpg</a>	<a href="http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg</a>
0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	<a href="http://images.amazon.com/images/P/0002005018.01.THUMBZZZ.jpg">http://images.amazon.com/images/P/0002005018.01.THUMBZZZ.jpg</a>	<a href="http://images.amazon.com/images/P/0002005018.01.MZZZZZZZ.jpg">http://images.amazon.com/images/P/0002005018.01.MZZZZZZZ.jpg</a>	<a href="http://images.amazon.com/images/P/0002005018.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0002005018.01.LZZZZZZZ.jpg</a>

# DATASETS

User-ID	ISBN	Book-Rating
276725	034545104X	0
276726	0155061224	5

# DATASETS

User-ID	Location	Age
1	nyc, new york, usa	NULL
2	stockton, california, usa	18
3	moscow, yukon territory, russia	NULL

# DATA PREPROCESSING

```
books = books[['ISBN', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher', 'Image-URL-L']]  
books.head()
```

index	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-L
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	<a href="http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg</a>
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	<a href="http://images.amazon.com/images/P/0002005018.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0002005018.01.LZZZZZZZ.jpg</a>
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	<a href="http://images.amazon.com/images/P/0060973129.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0060973129.01.LZZZZZZZ.jpg</a>
3	0374157065	Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	Gina Bari Kolata	1999	Farrar Straus Giroux	<a href="http://images.amazon.com/images/P/0374157065.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0374157065.01.LZZZZZZZ.jpg</a>
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	<a href="http://images.amazon.com/images/P/0393045218.01.LZZZZZZZ.jpg">http://images.amazon.com/images/P/0393045218.01.LZZZZZZZ.jpg</a>

THE FIRST STEP IN MAKING A MODEL IS DATA PREPROCESSING. DATA PREPROCESSING IS IMPORTANT BECAUSE IT ENSURES THAT THE DATA IS CLEAN, CONSISTENT, AND PROPERLY FORMATTED TO ENHANCE THE PERFORMANCE AND ACCURACY OF MACHINE LEARNING MODELS. HERE I HAVE REMOVED ALL THE COLUMNS THAT ARE NOT NEEDED FOR THE TRAINING OF THE MODEL

# DATA PREPROCESSING

```
x = ratings['user_id'].value_counts() > 50
```

```
y=x[x].index
```

IN THIS STEP, WE REMOVE ALL THE RATINGS WHOSE USERS HAVE GIVEN LESS THAN FIFTY RATINGS TO ENSURE THAT DATA IS CONSISTENT AND NOT NOISE

---

---

# DATA PREPROCESSING

```
ratings = ratings[ratings['user_id'].isin(y)]  
ratings.head()
```

	user_id	ISBN	rating	
173	276847	0446364193	0	
174	276847	3257200552	5	
175	276847	3379015180	0	
176	276847	3404145909	8	
177	276847	3404148576	8	

HERE, THE RATINGS ARE TRIMMED DOWN USING THE DATAFRAME RATINGS TO INCLUDE ONLY ROWS WHERE THE 'USER\_ID' COLUMN MATCHES VALUES PRESENT IN THE VARIABLE .

---

---

# DATA PREPROCESSING

```
rwb = ratings.merge(books,on='ISBN')
rwb.head()

 user_id      ISBN  rating          title    author  year  publisher           image_url
0   276847  0446364193      0  Along Came a Spider (Alex Cross Novels)  James Patterson  1993  Warner Books  http://images.amazon.com/images/P/0446364193.0...
1   278418  0446364193      0  Along Came a Spider (Alex Cross Novels)  James Patterson  1993  Warner Books  http://images.amazon.com/images/P/0446364193.0...
2    5483  0446364193      0  Along Came a Spider (Alex Cross Novels)  James Patterson  1993  Warner Books  http://images.amazon.com/images/P/0446364193.0...
3    7346  0446364193      0  Along Came a Spider (Alex Cross Novels)  James Patterson  1993  Warner Books  http://images.amazon.com/images/P/0446364193.0...
4    8362  0446364193      0  Along Came a Spider (Alex Cross Novels)  James Patterson  1993  Warner Books  http://images.amazon.com/images/P/0446364193.0...

final_rating = rwb.merge(number_rating, on='title')
```

THE DATASETS BOOKS AND RATINGS ARE MERGED BASED ON ISBN INTO A DATAFRAME CALLED RWB. THE RWB AND NUMBER\_RATING DATAFRAMES ARE MERGED ON TO CREATE THE DATAFRAME ON WHICH THE MODEL IS TRAINED ON

# DATA PREPROCESSING

```
final_rating = final_rating[final_rating['num_of_rating'] >= 50]
```

```
final_rating.head()
```

	user_id	ISBN	rating	title	author	year	publisher	image_url	num_of_rating
0	276847	0446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	http://images.amazon.com/images/P/0446364193.0...	198
1	278418	0446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	http://images.amazon.com/images/P/0446364193.0...	195
2	5483	0446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	http://images.amazon.com/images/P/0446364193.0...	195
3	7346	0446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	http://images.amazon.com/images/P/0446364193.0...	195
4	8362	0446364193	0	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	http://images.amazon.com/images/P/0446364193.0...	195

HERE, WE TRIM THE DATASET AGAIN BY REMOVING ALL BOOKS WHO HAVE LESS THAN 50 RATINGS TO MAKE SURE THAT THE BOOKS RECOMMENDED ARE NOT NICHE BOOKS

# DATA PREPROCESSING

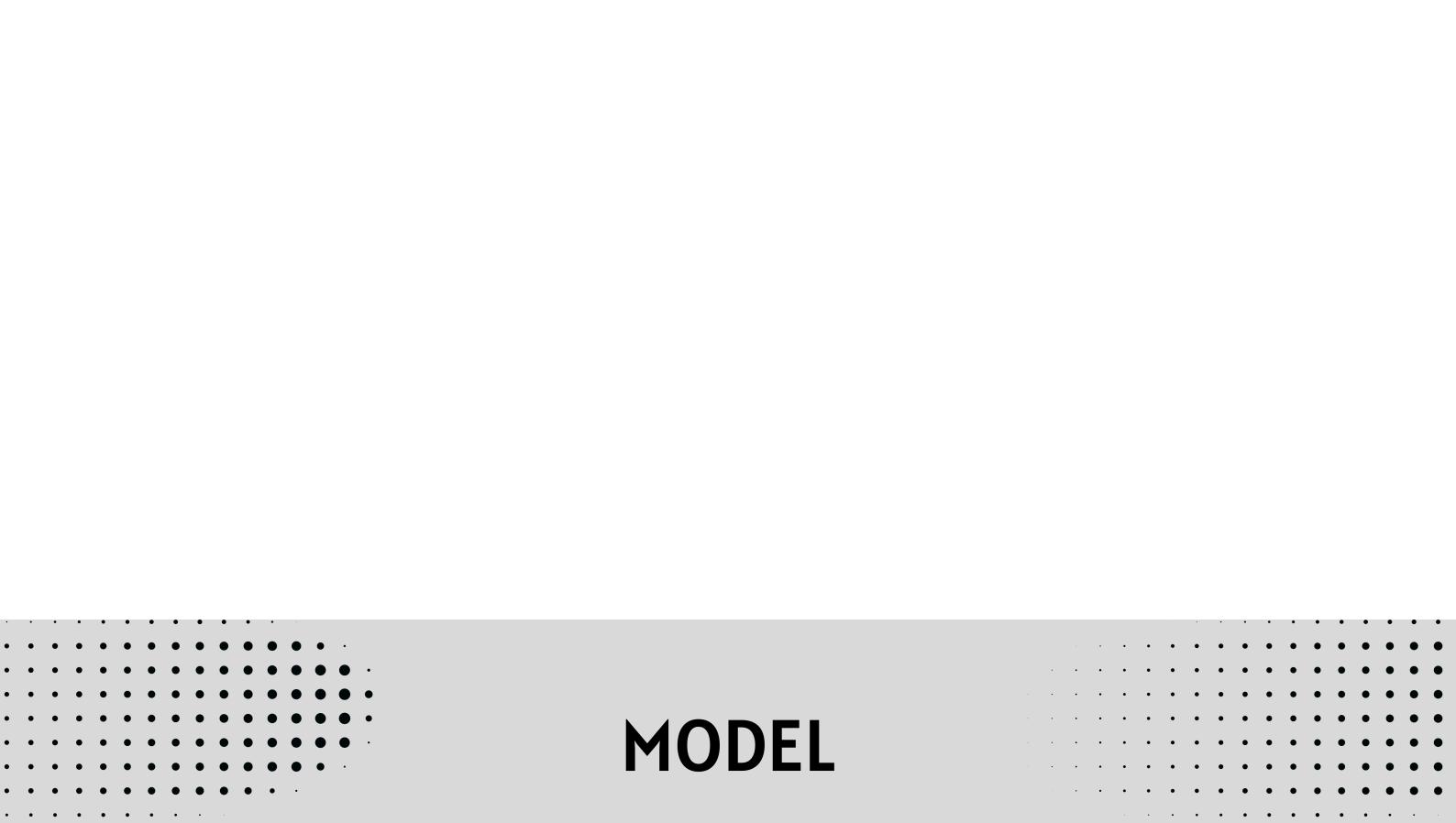
book_pivot = final_rating.pivot_table(columns='user_id', index='title', values= 'rating')																						
	user_id	183	243	254	507	626	638	643	741	882	929	...	277928	277965	278026	278137	278144	278188	278418	278582	278633	278843
	title																					
1984		NaN	NaN	9.0	NaN	...	NaN															
1st to Die: A Novel		NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN									
2010: Odyssey Two		NaN	...	NaN																		
204 Rosewood Lane		NaN	...	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN									
24 Hours		NaN	...	NaN																		
...		...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Year of Wonders		NaN	...	NaN																		
You Belong To Me		NaN	...	NaN																		
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values		NaN	...	NaN																		
Zoya		NaN	...	NaN																		
\"O!\" Is for Outlaw"		NaN	...	NaN																		
1373 rows × 3221 columns																						

THE DATAFRAME FINAL\_RATING IS TURNED INTO A PIVOT TABLE FOR TO BETTER HANDLE THE 2 DIMENSIONAL DATA



# ALGORITHM

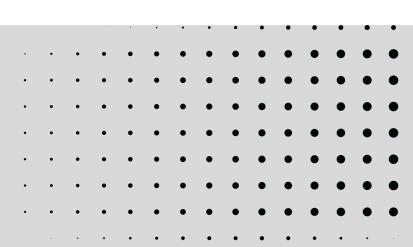
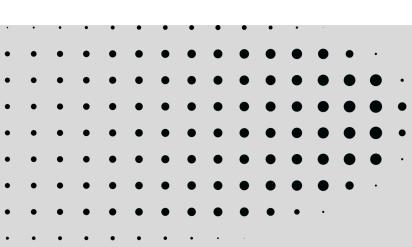
- THE ALGORITHM LEVERAGES THE CONCEPT OF NEAREST NEIGHBORS TO RECOMMEND BOOKS SIMILAR TO A GIVEN INPUT BOOK.
  - IT CALCULATES THE DISTANCE BETWEEN THE INPUT BOOK AND OTHER BOOKS IN THE DATASET USING A DISTANCE METRIC, SUCH AS EUCLIDEAN DISTANCE.
  - IT EMPLOYS THE KNN ALGORITHM TO FIND THE TOP K NEAREST NEIGHBORS OF THE INPUT BOOK.
  - THE PARAMETER K DETERMINES THE NUMBER OF SIMILAR BOOKS TO RECOMMEND. A SMALLER K VALUE SUGGESTS MORE LOCAL RECOMMENDATIONS, WHILE A LARGER K VALUE INCLUDES MORE DIVERSE RECOMMENDATIONS.
-



# MODEL

```
FROM SCIPY.SPARSE IMPORT CSR_MATRIX  
BOOK_SPARSE = CSR_MATRIX(BOOK_PIVOT)  
TYPE(BOOK_SPARSE)  
FROM SKLEARN.NEIGHBORS IMPORT NEARESTNEIGHBORS  
MODEL = NEARESTNEIGHBORS(ALGORITHM= 'BRUTE')  
MODEL.FIT(BOOK_SPARSE)  
DISTANCE, SUGGESTION  
MODEL.KNEIGHBORS(BOOK_PIVOT.ILOC[652,:].VALUES.reshape(1,-1),  
N_NEIGHBORS=6 )
```

---



# MODEL

```
DEF RECOMMEND_BOOK(BOOK_NAME):
    BOOK_ID = NP.WHERE(BOOK_PIVOT.INDEX == BOOK_NAME)[0][0]
    DISTANCE, SUGGESTION = MODEL.KNEIGHBORS(BOOK_PIVOT.ILOC[BOOK_ID,:].VALUES.reshape(1,-1),
N_NEIGHBORS=6)

FOR I IN RANGE(LEN(SUGGESTION)):
    BOOKS = BOOK_PIVOT.INDEX[SUGGESTION[I]]
    FOR J IN BOOKS:
        IF J == BOOK_NAME:
            PRINT(F"YOU SEARCHED '{BOOK_NAME}'\n")
            PRINT("THE SUGGESTION BOOKS ARE:\n")
        ELSE:
            PRINT(J)
```

---

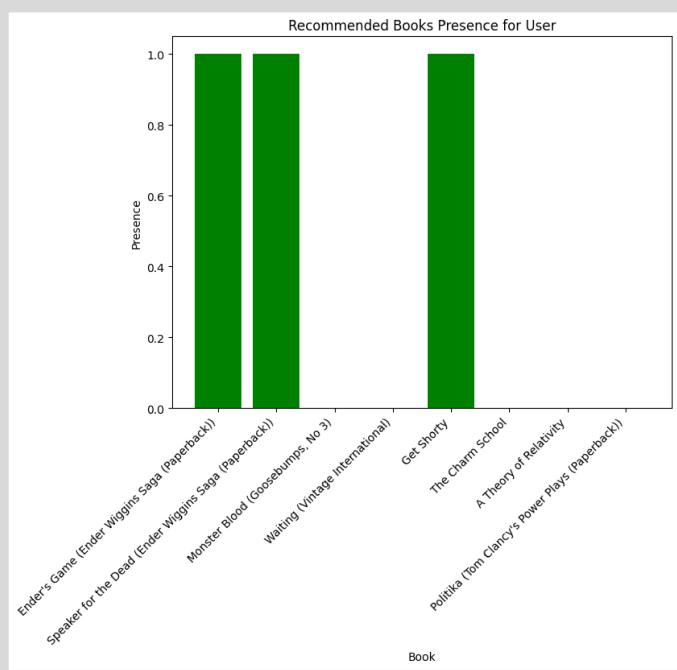
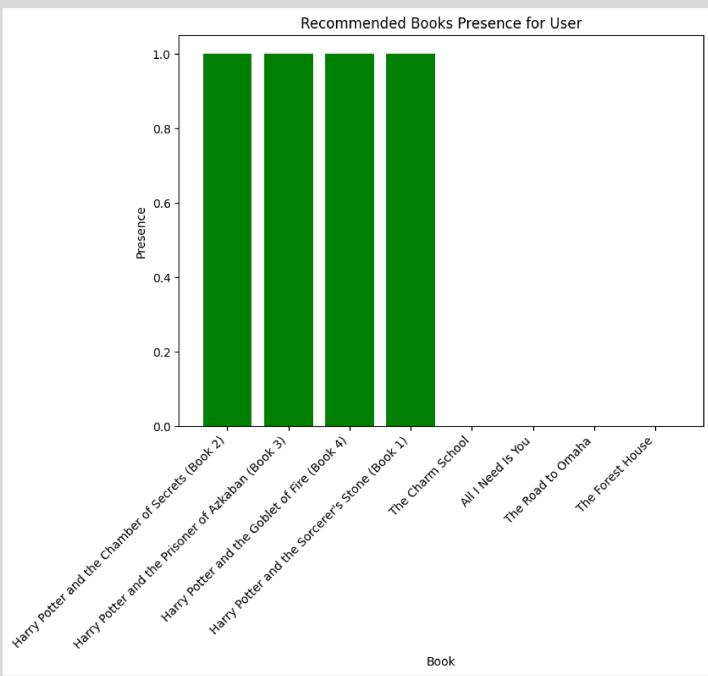
# MODEL OUTPUT

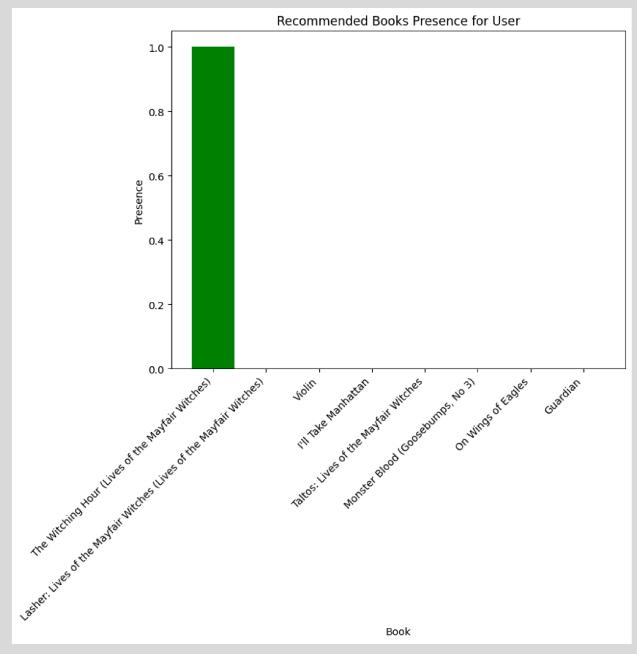
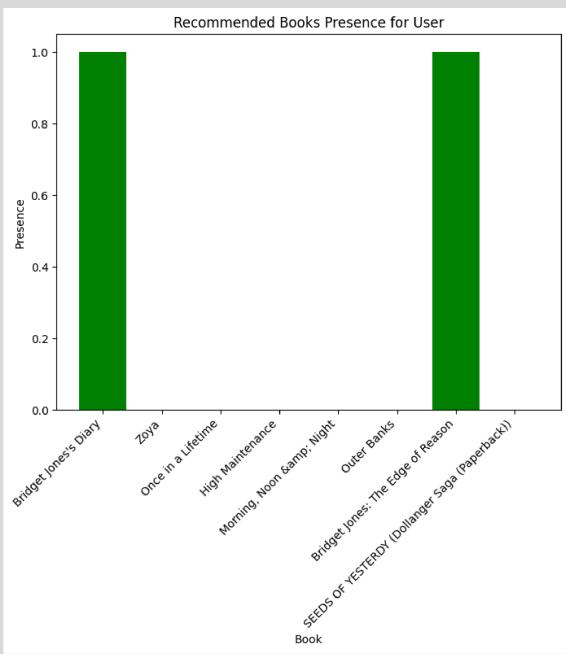
```
recommend_book('Harry Potter and the Chamber of Secrets (Book 2)')

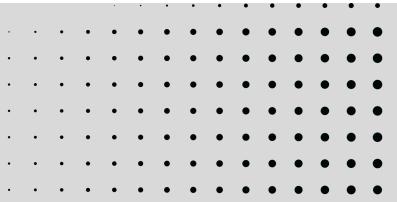
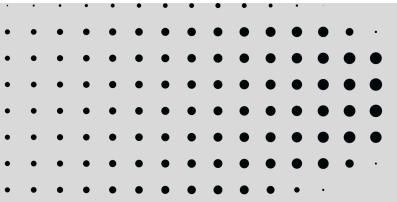
You searched 'Harry Potter and the Chamber of Secrets (Book 2)'

The suggestion books are:

Harry Potter and the Prisoner of Azkaban (Book 3)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Sorcerer's Stone (Book 1)
The Charm School
All I Need Is You
```

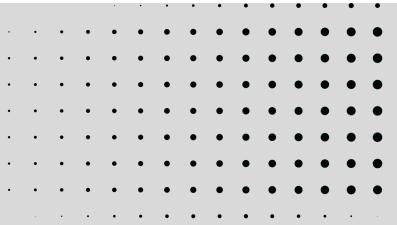
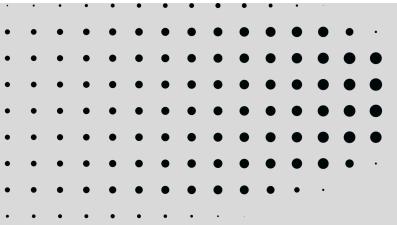






# BACK-END

- USING MVC MODEL FOR THE BACKEND. (MODEL VIEW CONTROLLER)
  - SERVER CONNECTION ESTABLISHED
  - CONTROLLERS AND ROUTES CREATED
  - POSTMAN API FOR VALUE POST AND GET
  - MONGO DB SERVER SETUP
  - ATLAS AND COMPASS CONNECTED
  - DATABASE ATLAS AND COMPASS CONNECTED TO THE BACKEND
  - SENSITIVE INFORMATION PROTECTED USING THE .ENV FILE
  - CREATED THE SCHEMA IN MODELS
  - INSERTED REGISTRATION SAMPLE DATA USING POSTMAN
-



# **BACK-END**

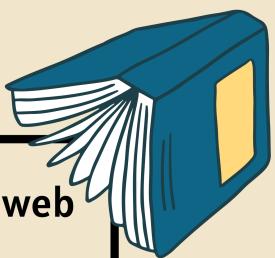
- CHECK FOR THE EMAIL EXISTENCE
  - PASSWORD ENCRYPTION USING BCRYPT {HASHING} .
  - TOKENS CREATED USING JWT FOR USER AUTHENTICATION AND AUTHORIZATION
  - LOGIN VERIFICATION DONE BY COMPARING THE PASSWORDS.
  - REGISTRATION ZOD VALIDATION FOR ERROR HANDLING
  - PICKLE FORMAT FOR MODAL SAVING
  - CLIENT RESPONDS VIA JSON REQUEST
-

# WORK DIVISION

Process	February				March				April			
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Project Selection	Start	End										
Abstract Report		Start	End		Start	End						
SRS Documentation					Start	End						
Design Report								Start	End			
UI Design								Start	End			
Dataset Selection		Start	End									
Basic Backend								Start	End			
Basic Frontend							Start	End				

# SOFTWARE/HARDWARE REQUIREMENT

- Flask for integrating machine learning algorithms with web application
- Python for machine learning algorithms (TensorFlow)
- HTML, CSS, and JavaScript for front-end development.
- Integrated Development Environment (IDE) such as Visual Studio Code, PyCharm, etc.
- Git for version control.
- Package managers like npm (Node.js) and pip (Python).
- Any modern devices capable of using web browsers like Google



# RESULTS

localhost:8501

## ML Based Books Recommender System

This is a collaborative filtering based recommendation system!

Type or select a book from the dropdown

Train Recommender System

Show Recommendation

Harry Potter and Harry Potter and Harry Potter and Exclusive The Cradle Will

Harry Potter and the Prisoner of Azkaban  
J. K. ROWLING

Harry Potter and the Goblet of Fire  
J. K. ROWLING

Harry Potter and the Sorcerer's Stone  
J. K. ROWLING

SANDRA EXCLUSIVE BROWN

# RESULTS

localhost:8501

## ML Based Books Recommender System

This is a collaborative filtering based recommendation system!

Type or select a book from the dropdown

Train Recommender System

Show Recommendation

No Safe Place    A Civil Action    Foucault's Pendulum    Long After Midn    Abduction

# Conclusions

1



In conclusion, our design report for the book recommendation system underscores the importance of personalized experiences in enhancing engagement and satisfaction.



Factors such as user preferences, content analysis, and collaborative filtering techniques, we have crafted a robust system architecture capable of delivering tailored recommendations.



We remain committed to refining our algorithms, incorporating user feedback, advancements in machine learning to continually enhance the efficiency and relevance of our recommendation system.



# FUTURE ENHANCEMENT

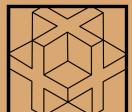
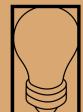
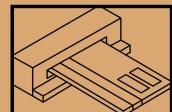
- UNIVERSAL BOOK SEARCH
- USER PROFILING
- LIBRARY MANAGEMENT
- FURTHER USER PERSONALISATION



# REFERENCES

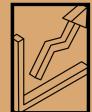
- **Express JS**

<https://expressjs.com/en/5x/api.html>



- **Kaggle**

<https://www.kaggle.com/docs>



*Thank  
You*

## **Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY  
(AUTONOMOUS)**

**RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039**

**(Affiliated to APJ Abdul Kalam Technological University)**



**Vision, Mission, Programme Outcomes and Course  
Outcomes**

**Institute Vision**

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

**Institute Mission**

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

**Department Vision**

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

## **Department Mission**

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

## **Programme Outcomes (PO)**

Engineering Graduates will be able to:

- 1. Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and re-

sponsibilities and norms of the engineering practice.

**9. Individual and Team work:** Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

**10. Communication:** Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

**11. Project management and finance:** Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

**12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

### **Programme Specific Outcomes (PSO)**

A graduate of the Computer Science and Engineering Program will demonstrate:

#### **PSO1: Computer Science Specific Skills**

The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

#### **PSO2: Programming and Software Development Skills**

The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

#### **PSO3: Professional Skills**

The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

### **Course Outcomes**

After the completion of the course the student will be able to:

**CO1:**

Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)

**CO2:**

Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)

**CO3:**

Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)

**CO4:**

Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)

**CO5:**

Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

## **Appendix C: CO-PO-PSO Mapping**

## COURSE OUTCOMES:

After completion of the course the student will be able to

<b>SL. NO</b>	<b>DESCRIPTION</b>	<b>Blooms' Taxonomy Level</b>
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)	Level 3: Apply
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)	Level 3: Apply

## CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PS O3
C O1	3	3	3	3		2	2	3	2	2	2	3	2	2	2
C O2	3	3	3	3	3	2		3	2	3	2	3	2	2	2
C O3	3	3	3	3	3	2	2	3	2	2	2	3			2
C O4	2	3	2	2	2			3	3	3	2	3	2	2	2
C O5	3	3	3	2	2	2	2	3	2		2	3	2	2	2

3/2/1: high/medium/low

## JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/ MEDIUM/ HIGH	JUSTIFICATION
101003/CS6 22T.1-PO1	<b>HIGH</b>	Identify technically and economically feasible problems by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.1-PO2	<b>HIGH</b>	Identify technically and economically feasible problems by analysing complex engineering problems reaching substantiated conclusions using first principles of mathematics.
101003/CS6 22T.1-PO3	<b>HIGH</b>	Design solutions for complex engineering problems by identifying technically and economically feasible problems.
101003/CS6 22T.1-PO4	<b>HIGH</b>	Identify technically and economically feasible problems by analysis and interpretation of data.
101003/CS6 22T.1-PO6	<b>MEDIUM</b>	Responsibilities relevant to the professional engineering practice by identifying the problem.
101003/CS6 22T.1-PO7	<b>MEDIUM</b>	Identify technically and economically feasible problems by understanding the impact of the professional engineering solutions.
101003/CS6 22T.1-PO8	<b>HIGH</b>	Apply ethical principles and commit to professional ethics to identify technically and economically feasible problems.
101003/CS6 22T.1-PO9	<b>MEDIUM</b>	Identify technically and economically feasible problems by working as a team.
101003/CS6 22T.1-PO10	<b>MEDIUM</b>	Communicate effectively with the engineering community by identifying technically and economically feasible problems.
101003/CS6 22T.1-P011	<b>MEDIUM</b>	Demonstrate knowledge and understanding of engineering and management principles by selecting the technically and economically feasible problems.
101003/CS6 22T.1-PO12	<b>HIGH</b>	Identify technically and economically feasible problems for long term learning.
101003/CS6 22T.1-PSO1	<b>MEDIUM</b>	Ability to identify, analyze and design solutions to identify technically and economically feasible problems.
101003/CS6 22T.1-PSO2	<b>MEDIUM</b>	By designing algorithms and applying standard practices in software project development and Identifying technically and economically feasible problems.
101003/CS6 22T.1-PSO3	<b>MEDIUM</b>	Fundamentals of computer science in competitive research can be applied to Identify technically and economically feasible problems.
101003/CS6 22T.2-PO1	<b>HIGH</b>	Identify and survey the relevant by applying the knowledge of mathematics, science, engineering fundamentals.

101003/CS6 22T.2-PO2	<b>HIGH</b>	Identify, formulate, review research literature, and analyze complex engineering problems get familiarized with software development processes.
101003/CS6 22T.2-PO3	<b>HIGH</b>	Design solutions for complex engineering problems and design based on the relevant literature.
101003/CS6 22T.2-PO4	<b>HIGH</b>	Use research-based knowledge including design of experiments based on relevant literature.
101003/CS6 22T.2-PO5	<b>HIGH</b>	Identify and survey the relevant literature for getting exposed to related solutions and get familiarized with software development processes by using modern tools.
101003/CS6 22T.2-PO6	<b>MEDIUM</b>	Create, select, and apply appropriate techniques, resources, by identifying and surveying the relevant literature.
101003/CS6 22T.2-PO8	<b>HIGH</b>	Apply ethical principles and commit to professional ethics based on the relevant literature.
101003/CS6 22T.2-PO9	<b>MEDIUM</b>	Identify and survey the relevant literature as a team.
101003/CS6 22T.2-PO10	<b>HIGH</b>	Identify and survey the relevant literature for a good communication to the engineering fraternity.
101003/CS6 22T.2-PO11	<b>MEDIUM</b>	Identify and survey the relevant literature to demonstrate knowledge and understanding of engineering and management principles.
101003/CS6 22T.2-PO12	<b>HIGH</b>	Identify and survey the relevant literature for independent and lifelong learning.
101003/CS6 22T.2-PSO1	<b>MEDIUM</b>	Design solutions for complex engineering problems by Identifying and survey the relevant literature.
101003/CS6 22T.2-PSO2	<b>MEDIUM</b>	Identify and survey the relevant literature for acquiring programming efficiency by designing algorithms and applying standard practices.
101003/CS6 22T.2-PSO3	<b>MEDIUM</b>	Identify and survey the relevant literature to apply the fundamentals of computer science in competitive research.
101003/CS6 22T.3-PO1	<b>HIGH</b>	Perform requirement analysis, identify design methodologies by using modern tools & advanced programming techniques and by applying the knowledge of mathematics, science, engineering fundamentals.
101003/CS6 22T.3-PO2	<b>HIGH</b>	Identify, formulate, review research literature for requirement analysis, identify design methodologies and develop adaptable & reusable solutions.

101003/CS6 22T.3-PO3	<b>HIGH</b>	Design solutions for complex engineering problems and perform requirement analysis, identify design methodologies.
101003/CS6 22T.3-PO4	<b>HIGH</b>	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6 22T.3-PO5	<b>HIGH</b>	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools.
101003/CS6 22T.3-PO6	<b>MEDIUM</b>	Perform requirement analysis, identify design methodologies and assess societal, health, safety, legal, and cultural issues.
101003/CS6 22T.3-PO7	<b>MEDIUM</b>	Understand the impact of the professional engineering solutions in societal and environmental contexts and Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6 22T.3-PO8	<b>HIGH</b>	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions by applying ethical principles and commit to professional ethics.
101003/CS6 22T.3-PO9	<b>MEDIUM</b>	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6 22T.3-PO10	<b>MEDIUM</b>	Communicate effectively with the engineering community and with society at large to perform requirement analysis, identify design methodologies.
101003/CS6 22T.3-PO11	<b>MEDIUM</b>	Demonstrate knowledge and understanding of engineering requirement analysis by identifying design methodologies.
101003/CS6 22T.3-PO12	<b>HIGH</b>	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by analysis, identify design methodologies and develop adaptable & reusable solutions.
101003/CS6 22T.3-PSO3	<b>MEDIUM</b>	The ability to apply the fundamentals of computer science in competitive research and prior to that perform requirement analysis, identify design methodologies.
101003/CS6 22T.4-PO1	<b>MEDIUM</b>	Prepare technical report and deliver presentation by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.4-PO2	<b>HIGH</b>	Identify, formulate, review research literature, and analyze complex engineering problems by preparing technical report and deliver presentation.

101003/CS6 22T.4-PO3	<b>MEDIUM</b>	Prepare Design solutions for complex engineering problems and create technical report and deliver presentation.
101003/CS6 22T.4-PO4	<b>MEDIUM</b>	Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions and prepare technical report and deliver presentation.
101003/CS6 22T.4-PO5	<b>MEDIUM</b>	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and Prepare technical report and deliver presentation.
101003/CS6 22T.4-PO8	<b>HIGH</b>	Prepare technical report and deliver presentation by applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS6 22T.4-PO9	<b>HIGH</b>	Prepare technical report and deliver presentation effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.
101003/CS6 22T.4-PO10	<b>HIGH</b>	Communicate effectively with the engineering community and with society at large by prepare technical report and deliver presentation.
101003/CS6 22T.4-PO11	<b>MEDIUM</b>	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work by prepare technical report and deliver presentation.
101003/CS6 22T.4-PO12	<b>HIGH</b>	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change by prepare technical report and deliver presentation.
101003/CS6 22T.4-PSO1	<b>MEDIUM</b>	Prepare a technical report and deliver presentation to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas.
101003/CS6 22T.4-PSO2	<b>MEDIUM</b>	To acquire programming efficiency by designing algorithms and applying standard practices in software project development and to prepare technical report and deliver presentation.
101003/CS6 22T.4-PSO3	<b>MEDIUM</b>	To apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs by preparing technical report and deliver presentation.
101003/CS6 22T.5-PO1	<b>HIGH</b>	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS6 22T.5-PO2	<b>HIGH</b>	Identify, formulate, review research literature, and analyze complex engineering problems by applying engineering and management principles to achieve the goal of the project.

101003/CS6 22T.5-PO3	<b>HIGH</b>	Apply engineering and management principles to achieve the goal of the project and to design solutions for complex engineering problems and design system components or processes that meet the specified needs.
101003/CS6 22T.5-PO4	<b>MEDIUM</b>	Apply engineering and management principles to achieve the goal of the project and use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS6 22T.5-PO5	<b>MEDIUM</b>	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO6	<b>MEDIUM</b>	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities by applying engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO7	<b>MEDIUM</b>	Understand the impact of the professional engineering solutions in societal and environmental contexts, and apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO8	<b>HIGH</b>	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice and to use the engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO9	<b>MEDIUM</b>	Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO11	<b>MEDIUM</b>	Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PO12	<b>HIGH</b>	Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PSO1	<b>MEDIUM</b>	The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas. Apply engineering and management principles to achieve the goal of the project.

101003/CS6 22T.5-PSO2	<b>MEDIUM</b>	The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry and to apply engineering and management principles to achieve the goal of the project.
101003/CS6 22T.5-PSO3	<b>MEDIUM</b>	The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur and apply engineering and management principles to achieve the goal of the project.

