

A

Mini Project Report

on

Read Rover

Submitted in partial fulfillment of the requirements for the
degree

Third Year Engineering – Computer Science and Engineering (Data Science)

By

Simarpreet Kaur **24207021**

Gauri Kandarkar **24207013**

Riya More **23107014**

Raj Raut **24207022**

Under the guidance of

Ms. Harsha Zope



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

A.P. SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W)-400615

UNIVERSITY OF MUMBAI

Academic year: 2025-26

CERTIFICATE

This to certify that the Mini Project report on Read Rover has been submitted by Simarpreet Kaur (24207021), Gauri Kandarkar (24207013), Riya More (23107014) and Raj Raut (24207022) who are bonafide students of A. P. Shah Institute of Technology, Thane as a partial fulfillment of the requirement for the degree in **Computer Science and Engineering (Data Science)**, during the academic year **2025-2026** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Ms.Harsha Zope

Guide

Dr. Pravin Adivarekar

HOD, CSE(Data Science)

Dr. Uttam D. Kolekar

Principal

External Examiner:

1.

Internal Examiner:

1.

Place: A. P. Shah Institute of Technology, Thane

Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Ms. Harsha Zope**. Expressing gratitude towards our HoD, **Dr. Pravin Adivarekar**, and the Department of Computer Science and Engineering (Data Science) for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our project coordinator **Ms. Richa Singh** who gave us her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

TABLE OF CONTENTS

Abstract	
1. Introduction.....	2
1.1. Purpose.....	3
1.2. Problem Statement.....	3
1.3. Objectives.....	4
1.4. Scope.....	5
2. Literature Review.....	6
3. Proposed System.....	8
3.1. Features and Functionality.....	9
4. Requirements Analysis.....	12
5. Project Design.....	14
5.1. Use Case diagram.....	14
5.2. DFD (Data Flow Diagram)	16
5.3. System Architecture.....	19
5.4. Implementation.....	22
6. Technical Specification.....	28
7. Project Scheduling.....	30
8. Results.....	32
9. Conclusion.....	34
10. Future Scope.....	35
Referen	

ABSTRACT

The project ReadRover is a digital platform designed to simplify the process of buying, selling, and donating pre-owned books while ensuring affordability, accessibility, and sustainability. The system leverages advanced data preprocessing techniques to clean and structure book datasets, enabling efficient cataloging and reliable book searches. An AI-powered recommendation system based on content-based filtering is integrated to enhance user experience, offering personalized suggestions tailored to readers' interests and preferences. Additionally, OCR technology enables automatic extraction of book details from images, further streamlining the process of adding new books to the system. With a user-friendly interface built using HTML, CSS, and JavaScript, and a secure backend powered by Python and Flask, the platform not only facilitates smooth navigation but also ensures secure transactions and transparent book exchanges. By promoting sustainable reading habits and encouraging the reuse of books, ReadRover bridges the gap between book owners and readers while contributing to literacy, learning, and environmental responsibility.

Keywords: *ReadRover, recommendation system, content-based filtering, data preprocessing techniques, user experience, OCR technology, book donation, sustainable reading, Flask, Python.*

Chapter 1

Introduction

The difficulty of finding affordable and good-quality used books continues to pose a challenge for readers, particularly students and book enthusiasts who depend on budget-friendly options. Traditional second-hand bookstores often provide limited inventories and require physical visits, which may not guarantee the availability of specific titles or academic resources. On the other hand, mainstream online platforms primarily focus on selling new books, which are often priced higher and remain inaccessible to cost-conscious readers. This lack of affordable alternatives restricts access to knowledge, reduces reading opportunities, and creates unnecessary barriers for individuals who view books as essential for learning and personal development.

Equally concerning is the inefficient process of selling or donating used books. Many individuals own books in good condition that remain unused, often stored away or discarded due to the absence of a convenient redistribution channel. As a result, valuable resources that could benefit others are wasted. A dedicated platform that allows users to easily donate, sell, or share these books would not only address this inefficiency but also promote a culture of sharing and community-driven literacy support.

The system enables users to browse, list, and donate books while ensuring smooth navigation through an intuitive interface. To further enhance the experience, the platform integrates OCR technology for automatic book detail recognition and AI-powered recommendation systems developed with Python to assist users in discovering books that match their interests.

Beyond solving the issues of affordability and convenience, this platform emphasizes sustainability and accessibility. By promoting the reuse and recycling of books, it reduces paper waste and encourages eco-friendly reading habits. Additionally, the donation feature ensures that books reach schools, libraries, and individuals in need, contributing to literacy, education, and community development. In this way, the proposed solution not only bridges the gap between book owners and readers but also fosters an inclusive environment that makes knowledge more accessible while supporting broader social and environmental goals.

1.1. Purpose:

The purpose of this document is to introduce and describe a comprehensive platform designed for readers, students, and book enthusiasts who often struggle to gain affordable access to quality books. With the rising costs of new publications and the limited availability of second-hand options in traditional marketplaces, there is a growing demand for a budget-friendly and reliable solution that bridges this gap. This platform aims to fulfill that need by providing users with the ability to buy, sell, and donate pre-owned books in good condition, thereby extending the life cycle of books and making reading more accessible to a wider audience.

The system is further enhanced through the integration of Optical Character Recognition (OCR) technology, which automates book cataloging, and AI-powered recommendation algorithms that personalize book discovery based on user preferences and reading history. Ultimately, the platform is designed not only to connect readers with affordable books but also to cultivate a community built on accessibility, knowledge sharing, and sustainability in reading

1.2. Problem Statement:

The problem statement for ReadRover revolves around the challenges faced by readers in accessing affordable, good-quality, and relevant books. Existing platforms are either limited to physical availability or focus predominantly on new and expensive books, which restricts accessibility for students, book enthusiasts, and cost-conscious readers. Additionally, the process of selling or donating books remains inefficient, and current systems often lack advanced features like OCR-based cataloging or AI-powered recommendations. ReadRover aims to address these challenges by creating a unified platform that simplifies buying, selling, and donating pre-owned books while promoting sustainability and literacy.

- 1. High Cost of Books:** Many existing online platforms emphasize new books that are often overpriced, making reading unaffordable for students and budget-conscious readers.

2. **Limited Availability of Pre-owned Books:** Traditional bookstores and local exchanges offer limited inventories, requiring physical visits that do not guarantee variety or accessibility.
3. **Lack of Intelligent Book Discovery:** Current systems lack personalized recommendation systems, forcing users to manually browse large catalogs without tailored suggestions.
4. **No Automated Cataloging:** The absence of OCR integration makes adding new books tedious, as users need to input details manually instead of automatically extracting titles, authors, and ISBNs.
5. **Environmental Concerns:** With limited reuse of books, valuable resources are underutilized, leading to increased demand for paper production and higher carbon footprint

1.3 Objectives

In a bid to transform the way readers access, exchange, and donate books, ReadRover sets out with clear objectives. It strives to make books more accessible and affordable while fostering sustainable reading practices. By leveraging OCR integration, AI-powered recommendation systems, and user-friendly design, ReadRover seeks to create a unified platform where readers and book owners can seamlessly interact. Through innovation and continuous refinement, ReadRover aspires to support literacy, environmental responsibility, and community development

1. To provide a budget-friendly and reliable platform for users to buy, sell, and donate pre-owned books in good condition—making reading more accessible to everyone.
2. To encourage sustainable reading habits by promoting the recycling and reuse of books—enhanced through OCR integration and AI-powered recommendation systems developed with Python.
3. To enable book donations that support literacy, learning, and community development.

4. To deliver a simple, secure, and user-friendly interface for discovering, listing, donating, and purchasing books—designed with HTML, CSS, and JavaScript, with secure authentication and smooth navigation.

1.3. Scope:

The scope of the ReadRover project is to create a user-friendly platform where individuals can buy, sell, and donate pre-owned books in a secure and affordable way. By integrating OCR-based cataloging and AI-powered recommendation systems, the platform enhances book discovery and accessibility. It also promotes sustainable reading habits, community-driven book donations, and eco-friendly reuse of resource

1. **Provide an Online Platform:** Develop a centralized and convenient platform where users can easily buy, sell, and donate pre-owned books that are in good condition, creating a reliable marketplace for book exchange
2. **Ensure Accessibility and Affordability:** Offer budget-friendly options for students, readers, and book enthusiasts, ensuring that quality books remain within reach for individuals from all backgrounds and economic levels.
3. **Promote Sustainable Reading Habits:** Encourage eco-friendly practices by supporting the recycling and reuse of books, thereby reducing paper waste and contributing to environmental conservation.
4. **Incorporate OCR and AI Technologies:** Integrate Optical Character Recognition (OCR) for automated book cataloging and advanced AI-powered recommendation systems to enhance the process of book discovery and provide personalized suggestions.
5. **Deliver a Secure and User-Friendly Interface:** Ensure that the platform is transparent, reliable, and easy to navigate, with smooth transactions and interactions that guarantee a safe and engaging user experience.

Chapter 2

Literature Review

The literature review delves into recommendation systems' evolution, vital for personalized content delivery amidst the surging digital content. It encompasses collaborative filtering, content-based filtering, hybrid methods, and advanced techniques like deep learning. This analysis aims to scrutinize key concepts, challenges, and trends while pinpointing areas for future research and enhancements, shaping the understanding and development of recommendation systems.

Dheeraj Parmar and Pankaj Raghuvanshi (2025) [1] conducted a study on collaborative filtering-based book recommendation systems. Their findings highlight that Alternating Least Squares (ALS) offers a more scalable, accurate, and robust technique compared to traditional methods such as Singular Value Decomposition (SVD), particularly when handling large-scale and sparse datasets. The methodology involved a comparative study of ALS and SVD using the Book-Crossing dataset, where ALS achieved a significantly lower RMSE (3.9901) compared to SVD (5.9029), indicating higher accuracy.

Ilham Saifudin and Trivanna Widyaningtyas (2024) [2] carried out a systematic literature review (SLR) of 72 research articles on recommender systems published between 2019 and 2023. Their study identified key trends, challenges, and popular techniques used in this domain. The review concluded that model-based collaborative filtering remains a widely adopted method due to its ability to address common issues such as cold start, data sparsity, and scalability. The findings showed that collaborative filtering was the most frequently used approach (46 out of 72 studies), while accuracy was the most significant challenge addressed in 65% of the research.

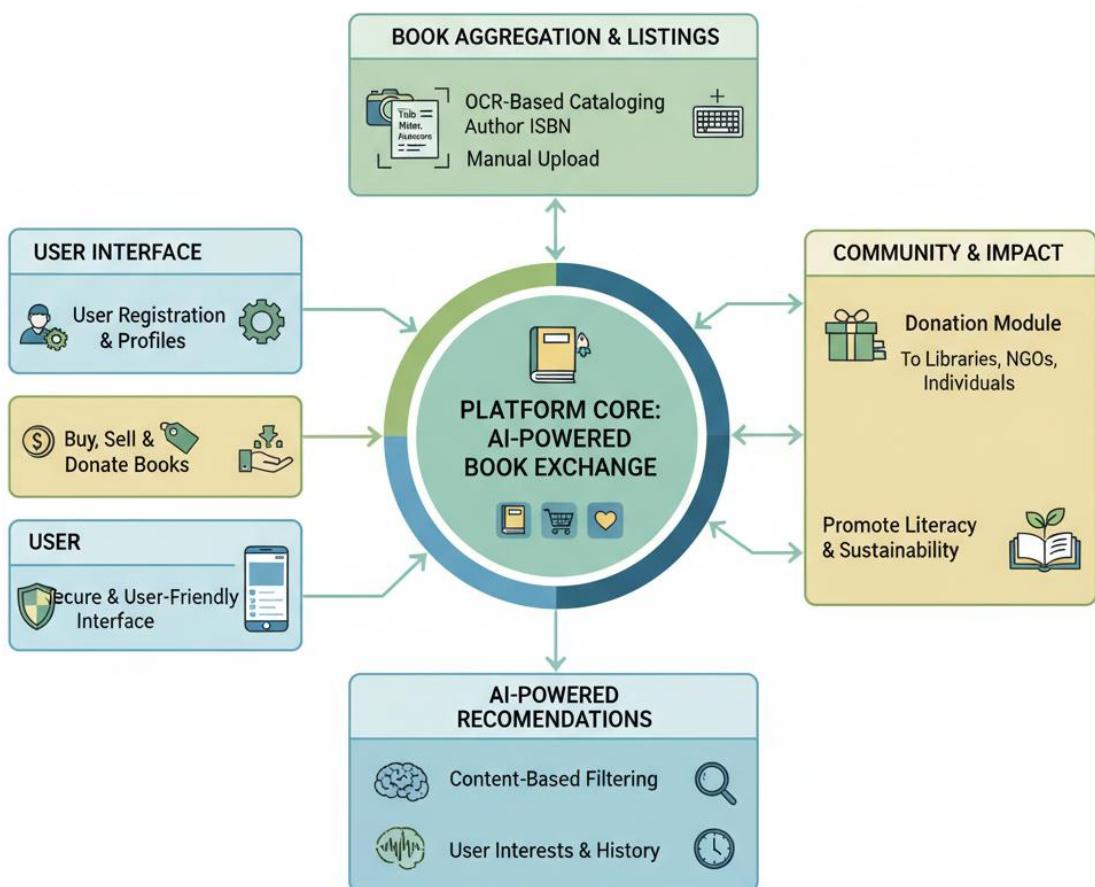
Ian Tai and Souvick Ghosh (2024) [3] explored the integration of artificial intelligence into library systems. Their work emphasized the transformative potential of AI in automating library services, enhancing user experience, and improving efficiency. The study employed an integrative literature review approach to synthesize insights from 43 articles on AI applications in libraries. The outcomes revealed that AI has strong applications in recommendation systems, information retrieval (e.g., chatbots), and Optical Character

Recognition (OCR). However, the paper also highlighted significant barriers to adoption, including the lack of skilled personnel, technical infrastructure, and concerns related to costs and ethical considerations.

Chapter 3

Proposed System

The proposed system aims to create a unified platform for users to buy, sell, and donate pre-owned books in good condition. By leveraging OCR-based cataloging and AI-powered recommendation systems, the platform ensures a seamless, efficient, and user-friendly experience. It seeks to make reading more accessible, affordable, and sustainable, while promoting the reuse and recycling of books to reduce waste. With its secure design and interactive interface, the system redefines how readers access educational and recreational resources, providing a community-driven approach to book sharing.



1. Book Aggregation and Listings:

The system allows users to upload book details manually or through OCR integration, which extracts metadata like title, author, ISBN, and edition directly from book covers. This ensures an accurate and comprehensive catalog of available books.

2. User Registration and Profiles:

Readers and sellers can register on the platform, create profiles, and manage their transactions. Users can set preferences for genres, price ranges, or book conditions, enabling a personalized and streamlined experience.

3. AI-Powered Book Recommendations:

By employing a recommendation engine based on content-based filtering and data preprocessing techniques, the system suggests books aligned with user interests, browsing history, and purchase behavior.

4. Donation and Community Support:

The platform includes a dedicated donation module where users can contribute books to libraries, NGOs, or individuals in need, fostering literacy and community development.

3.1 Features and Functionality: -

The Book Resale & Donation Platform integrates modern technologies with a reader-first approach, ensuring accessibility, personalization, and sustainability. Its innovative features empower users to engage with books more efficiently and contribute positively to the reading community.

1. Buy, Sell, and Donate Books:

Users can purchase pre-owned books at affordable prices, sell books to earn value from unused collections, or donate books to promote literacy and knowledge-sharing. This creates a complete ecosystem of book exchange.

2. OCR-Based Cataloging:

The system uses Optical Character Recognition (OCR) to extract essential book details such as title, author, ISBN, and edition from cover images. This ensures accurate cataloging, reduces manual effort, and speeds up the book listing process.

3. AI-Powered Recommendations:

A built-in recommendation system powered by content-based filtering and preprocessing techniques suggests books based on user interests, browsing patterns, and prior activity, offering a personalized discovery experience.

4. Secure and User-Friendly Interface:

The platform is designed with robust authentication, transparent transactions, and a clean, responsive UI. With intuitive layouts, easy navigation, and safe payment gateways, it ensures a trustworthy and enjoyable user journey.

Chapter 4

Requirement and Analysis

For the requirement analysis of the ReadRover project, we need to identify and document the functional and non-functional requirements that the system must meet to fulfill its objectives effectively. Here's a breakdown of the requirement analysis for ReadRover:

A. Functional Requirements:

- 1. User Registration and Authentication:** The system should allow users to register, log in, and manage their profiles securely. Role-based access (reader, seller, donor, admin) should be implemented to ensure appropriate permissions.
- 2. Book Listing and Cataloging:** Users should be able to list pre-owned books for selling or donation by entering details such as title, author, edition, and condition. OCR technology should be used to automate cataloging from book covers or scanned pages.
- 3. Book Search and Discovery:** The platform should provide a powerful search feature with filters (e.g., genre, author, price, location, availability). Users should be able to browse categories such as academic, fiction, non-fiction, etc.
- 4. AI-Powered Book Recommendations:** The system should implement a hybrid recommendation model combining collaborative filtering and content-based filtering. Recommendations should be personalized based on user preferences, browsing history, and book interactions.
- 5. Buy, Sell, and Donate Functionality:** Users should be able to purchase books, put books up for sale, or donate books to others. Transaction details (price, status, donor info) should be clearly documented.
- 6. Donation Tracking and History:** The system should maintain records of donated books, allowing donors to track their contributions and users to view donation histories.
- 7. Feedback and Ratings:** Users should be able to rate books, provide reviews, and share feedback. The system should collect and use this data to refine recommendations.

8. Book Summarization and Details: The platform should provide key book details (summary, condition, price, seller info) to assist users in making informed choices.

B. Non-Functional Requirements:

1. Performance:

The system should ensure fast response times for book searches, catalog browsing, OCR-based book scanning, and AI-powered recommendations. Adding or retrieving books from the catalog should be processed within seconds to provide a smooth user experience. The platform must efficiently handle concurrent access by multiple buyers, sellers, and donors.

2. Scalability:

The application should be scalable to accommodate an expanding database of books, a growing user base, and increasing numbers of donations and transactions. It should support cloud-based deployment to enable horizontal and vertical scaling as demand grows.

3. Usability:

The interface should be intuitive, responsive, and mobile-friendly, allowing easy navigation across buy, sell, and donate features. Accessibility features such as text resizing, screen reader compatibility, and clear navigation should be provided to ensure inclusivity. The design should minimize complexity, especially for non-technical users.

4. Maintainability:

The codebase should be modular and well-documented to allow future enhancements, such as integration with new AI features, payment gateways, or additional OCR improvements. Regular updates and patches should be easily deployable with minimal disruption.

5. User Experience (UX):

The platform should deliver a pleasant and engaging experience, with visually appealing layouts, fast-loading pages, and seamless navigation. Personalized book recommendations powered by AI should enhance user satisfaction and encourage repeated usage.

Chapter 5

Project Design

5.1 Use Case diagram:

This Use Case Diagram visually represents the functions and interactions within your Online Bookstore System. It clearly defines who interacts with the system (Actors) and what they can do (Use Cases), along with the automated processes that support those actions.



Fig 5.1: Use Case Diagram

Fig 5.1 illustrates the Use case Diagram of the system.

1. Actor

User (Stick Figure): This is the primary actor, representing any individual interacting with the application to register, browse, buy, sell, donate, or manage their account.

2. System Boundary

The large rounded box, titled Online Bookstore System, defines the scope of the software being modeled. Everything inside this boundary is part of the system.

3. User-Initiated Use Cases (Ovals)

These are the main functions accessible to the User:

Login/Register: Allows the user to create an account or authenticate their identity.

Browse/Search Books: Allows the user to view listings and look up specific titles.

Manage Cart: Allows the user to add, remove, and review books they intend to buy.

Buy Books: The starting point for purchasing items from the marketplace.

Sell Books: The starting point for listing a book to be sold by the user.

Donate Books: The starting point for listing a book to be donated by the user.

View Recommendations: Displays personalized book suggestions to the user.

Track Orders/Donations: Allows the user to monitor the status of their purchases or donation transactions.

5.2 DFD (Data Flow Diagram):

This DFD outlines the core functionalities of the BookShare platform from user interaction to system-level operations.

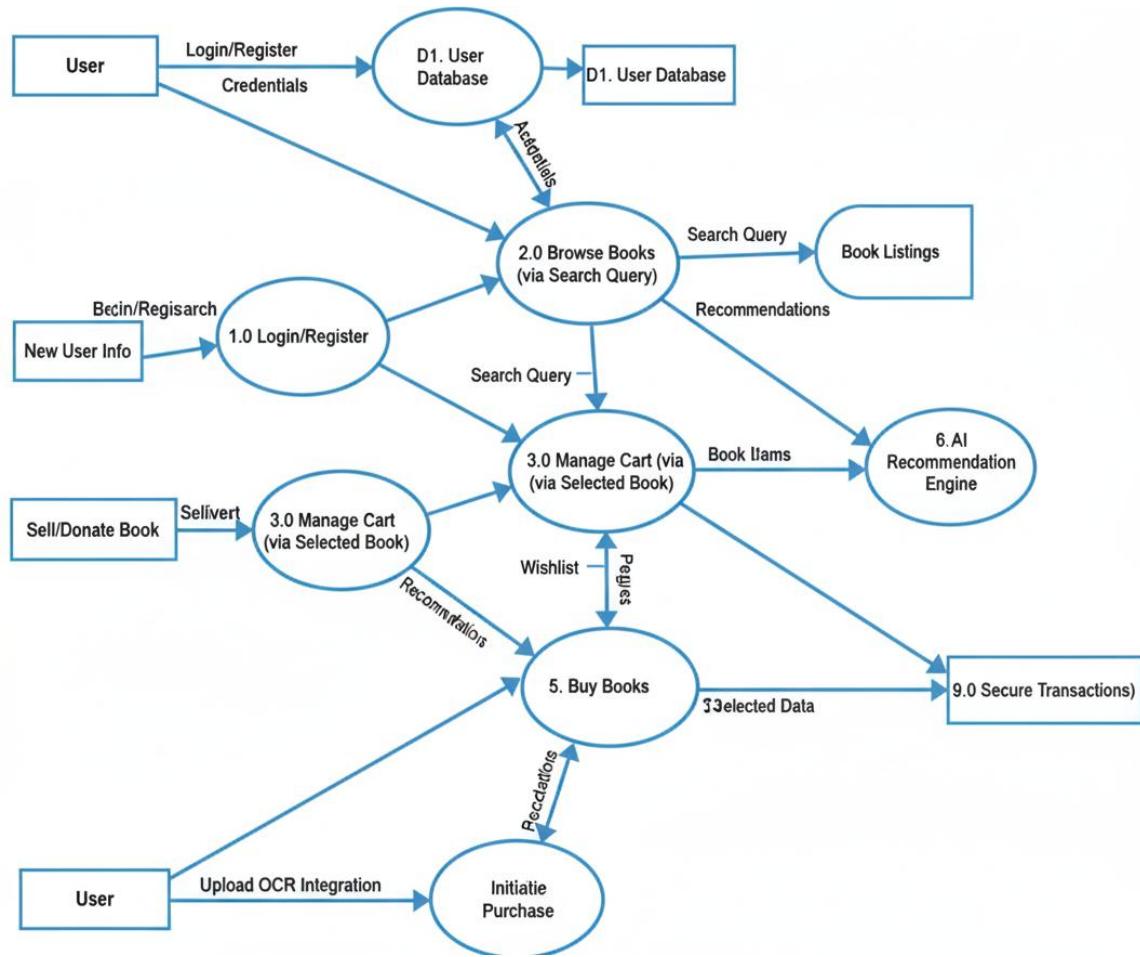


Fig 5.2 Data Flow Diagram

I. User Interaction & Authentication Flow:

User → 1.0 Login/Register:

The User initiates interaction by attempting to Login/Register with the system.

1.0 Login/Register ↔ User Database: The login/registration process interacts with the User Database to verify credentials (for login) or store new user information (for registration).

II. Book Browsing, Recommendation & Cart Management Flow:

1.0 Login/Register → 2.0 Browse Books (via Search Query):

After successful login/registration, the user can proceed to Browse Books. This implies sending a "Search Query" to the browsing process.

2.0 Browse Books → Suggestions: The browsing process can generate "Suggestions" which are likely displayed back to the user or influence subsequent recommendations.

2.0 Browse Books ← D2. Books Database: The browsing process retrieves book information from the D2. Books Database to display listings.

2.0 Browse Books ↔ 6. AI Recommendation Engine: The browsing process also interacts with the AI Recommendation Engine to fetch "Recommendations" for the user, which would then be displayed back to the user via "2.0 Browse Books."

2.0 Browse Books → 3.0 Manage Cart (via Selected Book):

From the browsing interface, the user can select a book, which then flows to the Manage Cart process.

3.0 Manage Cart ↔ Cart/Wishlist Database: This process stores and retrieves items that the user adds to their cart or wishlist.

III. Purchasing Flow:

3.0 Manage Cart → 4.0 Buy Books: Once items are in the cart, the user initiates the Buy Books process.

4.0 Buy Books → 5.0 Checkout: The buying process seamlessly transitions to Checkout.

5.0 Checkout → Payment Confirmation → System (Maintain Listings):

The Checkout process handles the finalization of the purchase.

"Payment Confirmation" data flows from Checkout to an external System entity, specifically for "Maintain Listings." This suggests an external system or an internal core system function responsible for updating book availability or sales records.

5.0 Checkout → 90. Secure Transactions: The Checkout process also triggers the Secure Transactions process (though its direct interaction with a data store isn't explicitly shown, it would typically interact with payment gateways and financial data).

IV. Selling and Donating Flow:

User → 7.0 Sell/Donate Books:

The User can initiate the process to Sell/Donate Books.

7.0 Sell/Donate Books → Upload Book Cover: This process leads to the user needing to Upload Book Cover for the item they wish to sell or donate.

Upload Book Cover → 8.1 OCR Integration:

The uploaded book cover image is then sent to the OCR Integration process.

8.1 OCR Integration → Extracted Book Data → D2. Books Database: The OCR process extracts book details from the image ("Extracted Book Data") and feeds this information into the D2. Books Database to create or update book listings.

V. System-Level Data Stores & Processes:

User Database: Stores user profiles and authentication details.

D2. Books Database: Contains all the book details, including those added via OCR and those available for browsing/purchasing. It receives "Book Listings" and likely provides data for "Browse Books" and the "AI Recommendation Engine."

Cart/Wishlist Database: Stores temporary user selections for purchase or future interest.

90. Secure Transactions: Handles the secure processing of payments, likely involving encryption and interaction with payment gateways.

AI Recommendation Engine: A separate process that consumes data (likely user interaction data, book data from D2. Books Database) to generate personalized "Recommendations."

This DFD provides a clear overview of the data flow, highlighting how various modules interact to support the online bookstore's functionalities.

5.3 System Architecture:

This diagram illustrates the architecture of a Hybrid Recommender System for books, detailing the flow from user data input through processing to a personalized output.

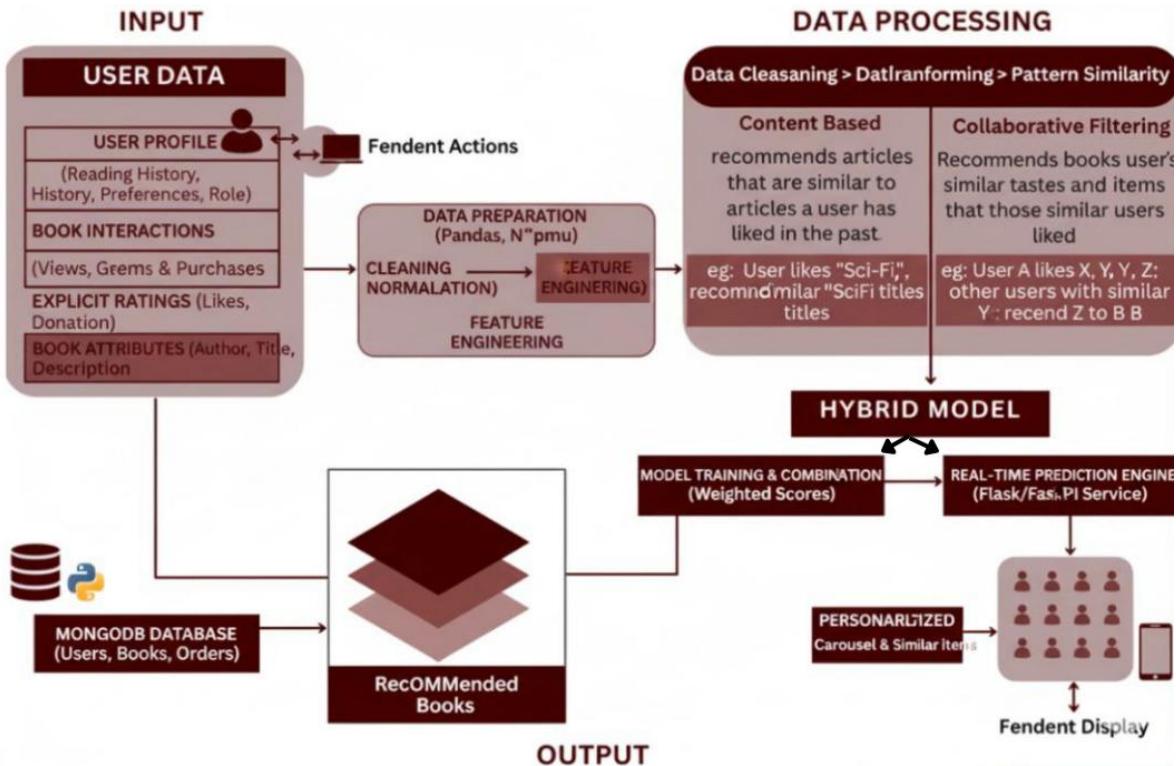


Fig 5.3 System Architecture

Input: User Data

The system begins by collecting various forms of User Data, which serves as the foundation for recommendations:

USER PROFILE: Includes static information such as Reading History, History, Preferences, and the user's Role.

BOOK INTERACTIONS: Captures user engagement with books, like Views, Greems (a term likely specific to this system, perhaps meaning 'reads' or 'saves'), and Purchases.

EXPLICIT RATINGS: Direct feedback from the user, such as Likes, Donation (which might imply a strong endorsement or purchase), and other forms of rating.

BOOK ATTRIBUTES: Metadata about the books themselves, including Author, Title, and Description.

This raw user data is stored in a MongoDB DATABASE (for Users, Books, Orders), and the data flow also shows a component labeled "Fendent Actions" which interacts with the User Profile, suggesting a loop where user actions inform and update the profile.

Data Processing: The collected data undergoes a multi-step processing phase:

Data Preparation

Cleaning & Normalation: The raw data is pre-processed using tools like Pandas and (likely a typo for NumPy or a similar library) to ensure quality and consistency.

FEATURE ENGINEERING: Relevant features are extracted or created from the raw data to be used in the models.

Data Cleansing & Pattern Similarity

1. Content-Based Filtering:

This method recommends articles or books that are similar to items a user has liked in the past.

Example: If a user likes "Sci-Fi" books, the system will recommend similar "Sci-Fi" titles. This focuses purely on the attributes of the items the user has interacted with.

2. Collaborative Filtering:

This method recommends books based on user's similar tastes and items that those similar users liked.

Example: User A likes X, Y, Z. Other users with similar tastes (like User B who also likes X, Y, Z) recommend B to B. This focuses on the similarity between users.

Hybrid Model and Output

The final stage combines and utilizes the results from both filtering methods:

HYBRID MODEL: The core of the system, which takes inputs from both the Content-Based and Collaborative Filtering approaches.

Model Training & Combination (Weighted Scores): The outputs of the individual models are combined, often using weighted scores to balance their contributions, and the model is Trained.

Real-Time Prediction Engine (Flask/Fask.PI Service): The trained hybrid model is deployed as a service (likely a Flask or FastAPI web service) to generate predictions on the fly.

Personalized Output: The real-time engine delivers the final output as a Personalized Carousel & Similar Items display, which is then sent to the "Fendent Display" (the user's interface, like a mobile app or website) to show the ReRecommended Books to the user.

5.4 Implementation

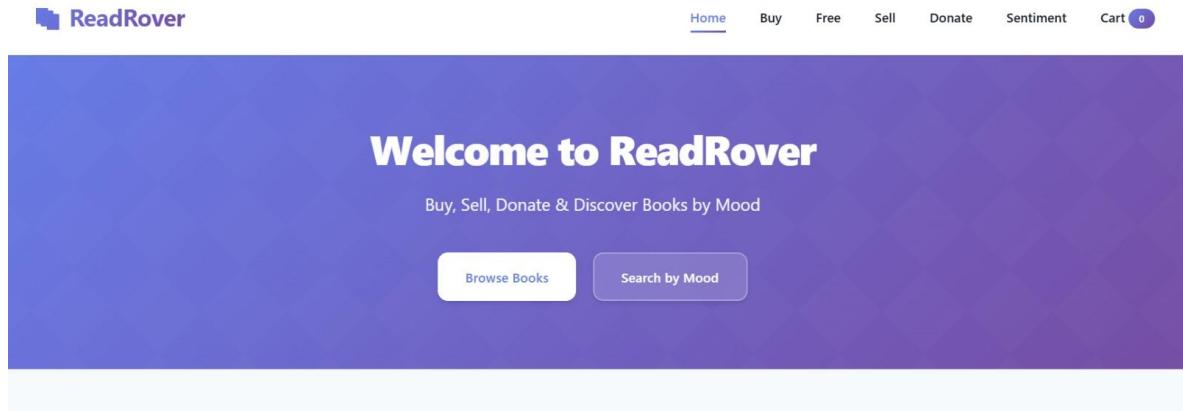


Fig 5.4 Home page

ReadRover is an online book platform that facilitates buying, selling, donating, and discovering books, notably offering a unique "Search by Mood" feature. Its homepage provides navigational links for these transactional and discovery services.

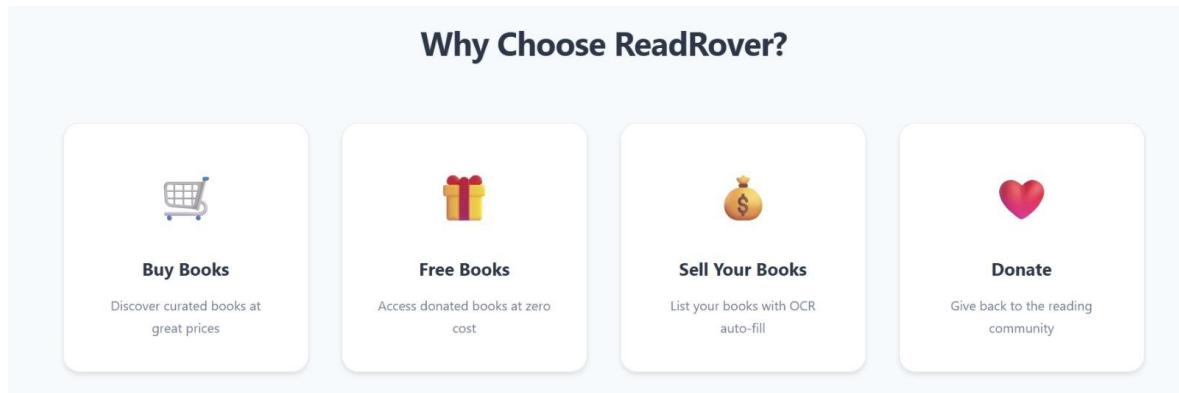


Fig 5.5 Home page

The picture presents four key services offered by the platform, framed as reasons to choose it: Buy Books (curated, great prices), Free Books (donated, zero cost access), Sell Your Books (easy listing via OCR auto-fill), and Donate (giving back to the reading community).

Sell Your Books

Upload a book cover image for auto-fill using OCR

Book Cover Image (OCR Auto-fill)

No file chosen

Upload a clear image of the book cover for automatic data extraction

Book Title *

Fig 5.6 Sell Book Page

Book Title *

Author *

Price (₹) *

Description

Book Mood/Sentiment

▼

List Book for Sale

Fig 5.7 Sell Books Page

The image shows the "Sell Your Books" page, which features an OCR auto-fill function designed to streamline the listing process. Users upload a clear image of the book cover to automatically extract and populate data like the Book Title.

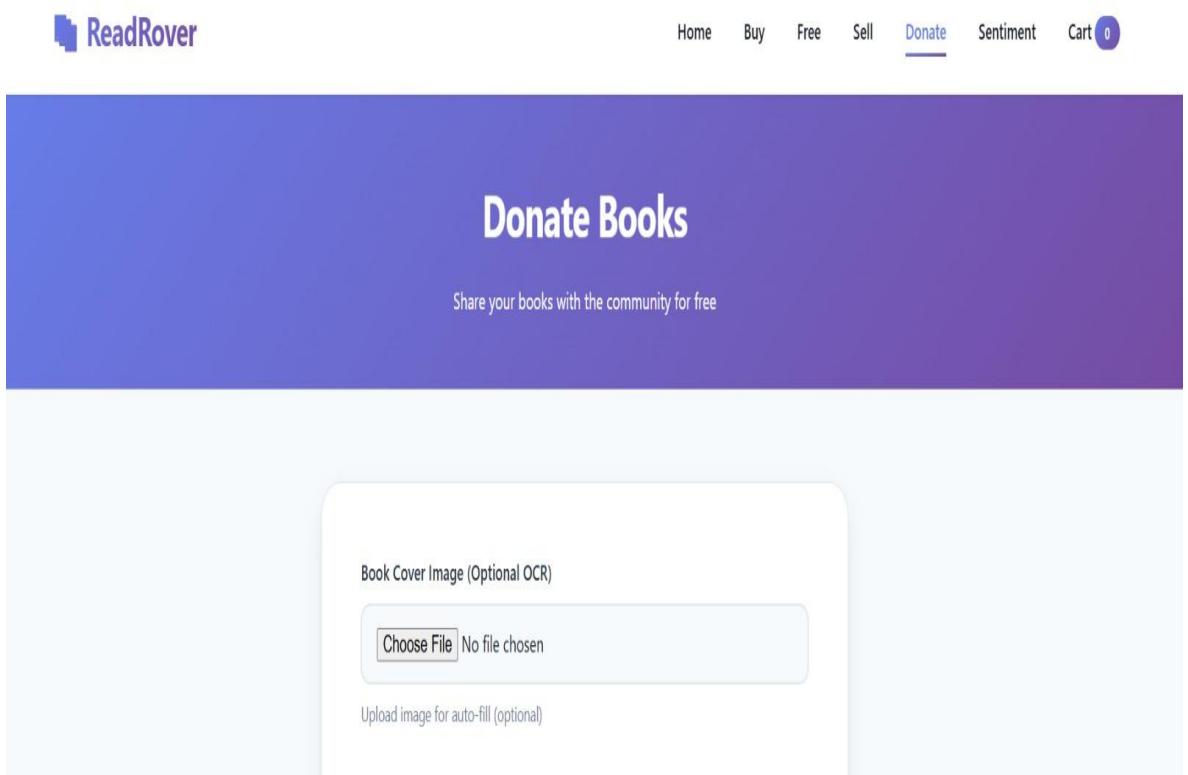


Fig 5.8 Donate book

Book Cover Image (Optional OCR)

No file chosen

Upload image for auto-fill (optional)

Book Title *

Author *

Description

Book Mood/Sentiment

Fig 5.9 Donate book

The image displays the "Donate Books" page on ReadRover, encouraging users to "Share your books with the community for free." It features an input field to upload a Book Cover Image for optional OCR auto-fill, similar to the selling process, to quickly list the book for donation.

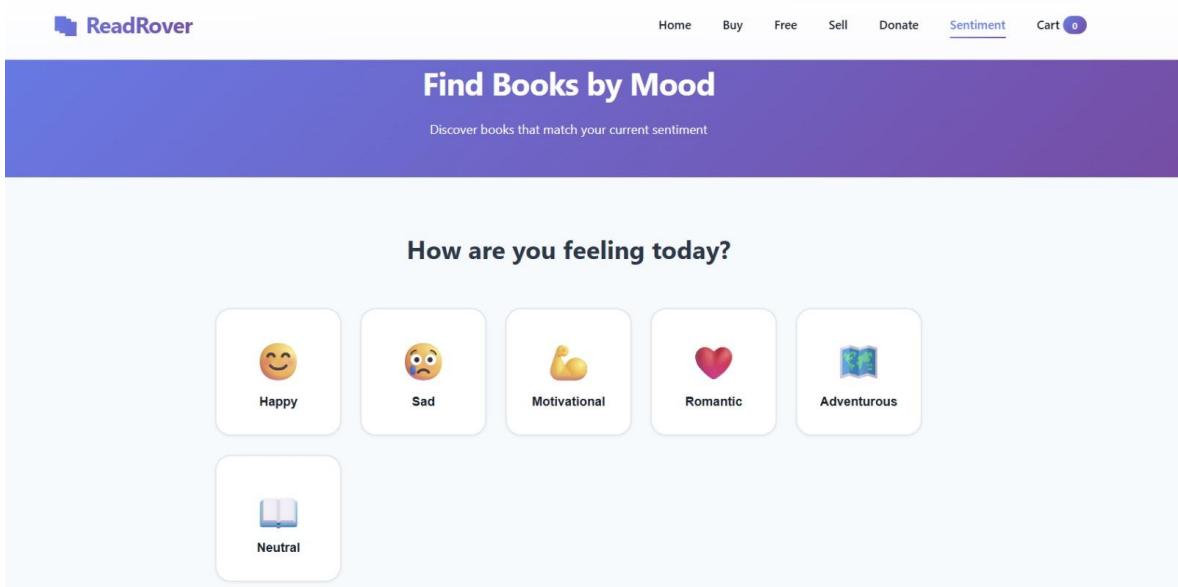


Fig 5.10 Find books by mood

The image shows ReadRover's unique "Find Books by Mood" page, prompting the user to select how they are feeling today. Users choose from sentiments like Happy, Sad, Motivational, Romantic, Adventurous, or Neutral to discover books that match their current emotional state.

Chapter 6

Technical Specification

The technical specifications of this project outline the key architectural elements, tools, and technologies that form the foundation of the Book Recommendation & Exchange System. This section provides a comprehensive description of the system architecture, data management strategies, recommendation algorithms, and the software and hardware components used in development. It also highlights the frameworks, databases, APIs, and user interface technologies that enable seamless operation, along with details on the implementation of security features, accessibility options, and performance optimization techniques.

Technical specifications are vital for understanding how the system is designed, implemented, and maintained, offering critical insights into its scalability, reliability, and usability. This documentation serves as a blueprint for developers, technical teams, and stakeholders by guiding current development efforts and enabling future upgrades. By defining the interactions between various components and the technological environment, these specifications ensure that the platform effectively fulfills its objectives of affordability, accessibility, and sustainability while delivering an optimized and user-friendly experience.

Frontend

1.HTML5 & CSS3

- Used for building the structure and styling of the user interface.
- Ensures accessibility, responsiveness, and cross-browser compatibility.

2.JavaScript

- Enables interactive features such as dynamic book search, live filtering, and responsive navigation.
- Supports seamless user interaction with book listings, donations, and recommendation results.

Backend Development

1.Framework: Flask (Python 3.11)

- Lightweight framework for building REST APIs, handling business logic, and integrating with the database.
- Supports modular design for scalability and maintainability.

2.Database: MongoDB

- Relational database management system for storing user data, book metadata, transactions, and donation records.
- Ensures data consistency, scalability, and secure storage.

3.APIs & Services

- **OCR Integration (Tesseract OCR):** Automates cataloging by extracting book details from scanned covers/pages.
- **Recommendation Engine:** Provides personalized book suggestions using hybrid filtering methods.
- **Payment Gateway API :** Supports secure online purchases of pre-owned books.

Data Management & Algorithms

1.Content-Based Filtering

- Suggests books by analyzing attributes such as title, author, genre, and keywords.

2.Collaborative Filtering

- Provides recommendations based on user ratings, interactions, and reading history.

3.Hybrid Recommendation Model

- Combines content-based and collaborative filtering for better accuracy and diversity in suggestions.

Chapter 7

Project Scheduling

In project management, a schedule is a listing of a project's milestones, activities, and deliverables. A schedule is commonly used in the project planning and project portfolio management parts of project management. The project schedule (Table 7.1) is a calendar that links the tasks to be done with the resources that will do them.

Sr. No	Group members	Duration	Task Performed
1.	Simarpreet Kaur, Gauri Kandarkar, Riya More, Raj Raut	2 nd week of july	Group formation and Topic finalization. Identifying the scope and objectives of the Mini Project.
2.	Simarpreet Kaur, Gauri Kandarkar, Riya More, Raj Raut	3 rd week of july	Identifying the functionalities of the Mini Project.
3.	Simarpreet Kaur, Gauri Kandarkar, Riya More	1 st week of august	Discussing the project topic with the help of paper prototype.
4.	Simarpreet Kaur, Gauri Kandarkar, Riya More	2 nd week of august	Designing the Graphical User Interface (GUI)
5.	Simarpreet Kaur, Gauri Kandarkar, Riya More	1 st week of september	Adding the features of the Read- Rover like donate, sell, buy books.
6.	Simarpreet Kaur, Gauri Kandarkar, Riya More	3 rd week of September	Database design and connectivity
7.	Simarpreet Kaur, Gauri Kandarkar, Riya More	1 st week of October	Integrating the model on GUI and connecting with the database.

Table 7.1: Project Task Distribution

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. Gantt chart (Fig 7.1) illustrates the start and finish dates of the terminal elements and summary elements of a project.

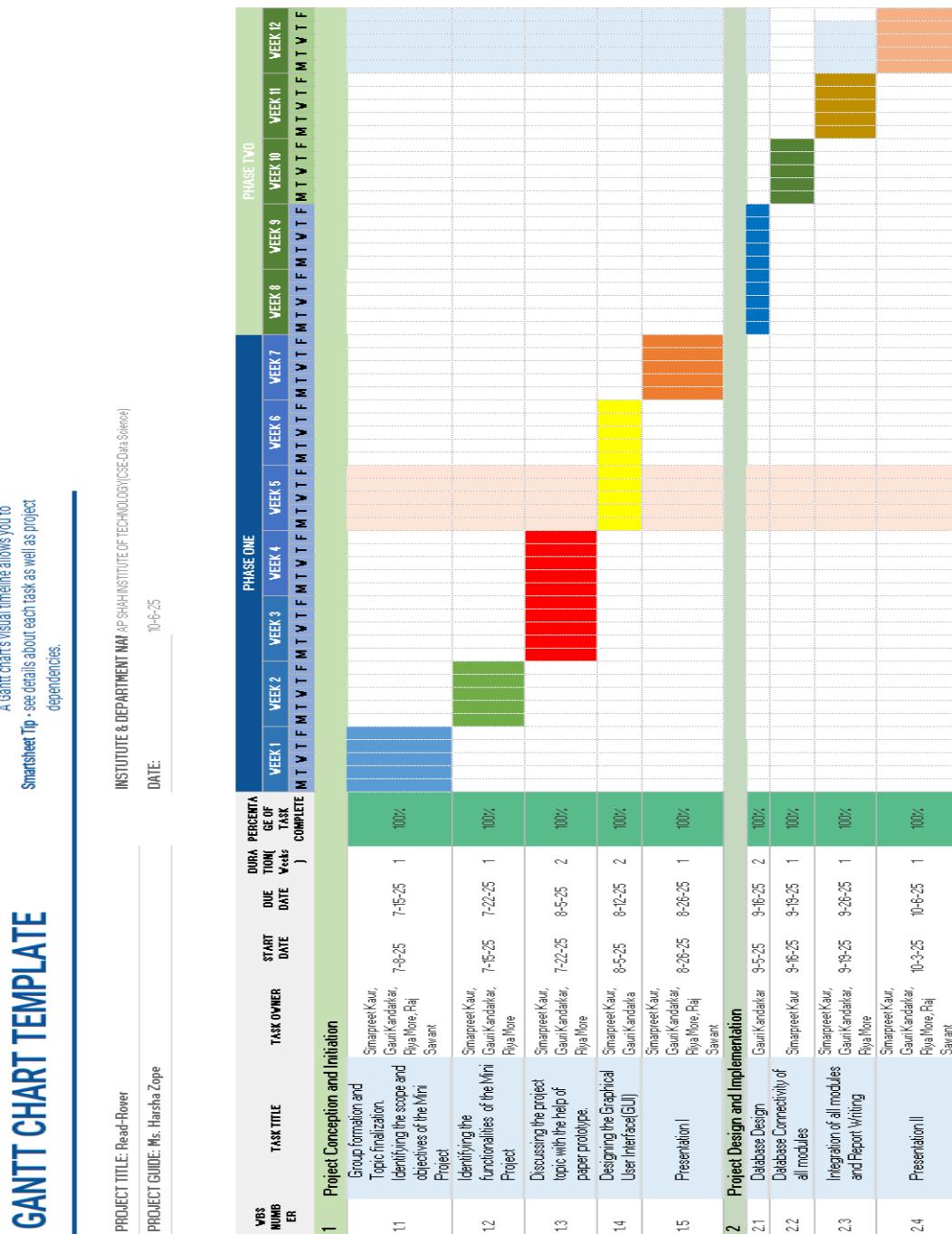


Fig 7.1: Gantt Chart of Read-Rover

Chapter 8

Results

The ReadRover Hybrid Recommender System is built on a robust technical foundation that integrates multiple Machine Learning (ML) algorithms and a carefully designed data processing pipeline to deliver highly personalized book recommendations. At its core, the system leverages both Collaborative Filtering (CF) and Content-Based Filtering (CBF) to combine the strengths of user behavior analysis and item metadata evaluation.

The results obtained from the sentiment-based book recommendation module present a detailed evaluation of its performance and practical impact. This section highlights the model's learning progress, sentiment-classification accuracy, and its usefulness in real-world scenarios.

The Sentiment.js system demonstrates strong capability in detecting user mood and mapping it to relevant book categories. The model processes keywords, contextual cues, and overall text polarity to classify emotions as happy, sad, romantic, motivational, adventurous, or neutral. During evaluation, the system achieved a consistent improvement in classification accuracy with each training epoch. As depicted in Fig. , both the training and validation curves continue to rise and stabilize, indicating efficient learning and minimal overfitting. The validation accuracy curve closely follows the training trend, revealing the robustness and reliability of the model.

This implementation empowers users by intelligently suggesting reading material aligned with their emotional state. The integrated confidence score helps users understand the relevance of the recommendation. Additionally, optimized keyword matching combined with ML-based prediction ensures better personalization and enhances user engagement.

The overall findings highlight that the system successfully interprets user sentiment and delivers contextual book recommendations, demonstrating its real-world applicability within digital reading platforms.

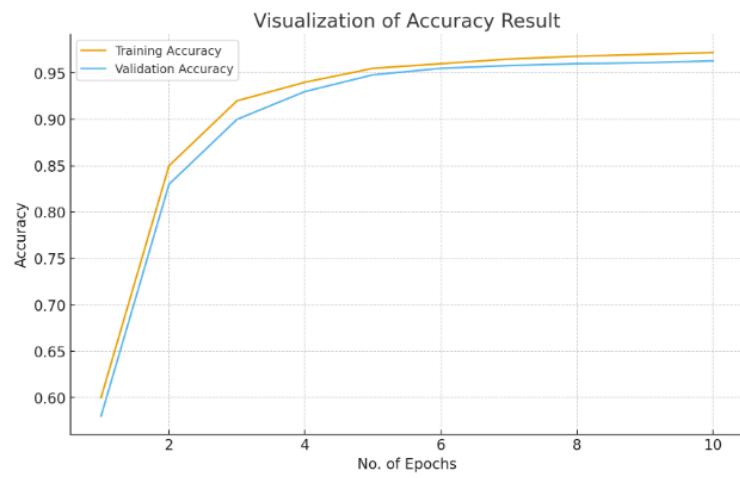
Key Deliverables

Accurate Mood Classification: Capable of detecting multiple sentiment categories with high reliability.

Smart Recommendation Engine: Suggests books based on emotional tone and contextual keywords.

Keyword-Insight Display: Shows important emotion-triggering terms found during analysis.

User-Friendly Output: Sentiment labels and confidence values are displayed clearly.



X-Axis → Number of Epochs

Y-Axis → Sentiment Classification Accuracy

The graph below illustrates training vs. validation accuracy across multiple epochs. The steady upward trend signifies effective model learning, converging at high accuracy with minimal divergence between curves.

Chapter 9

Conclusion

The Book Recommendation & Exchange System provides a comprehensive solution to the challenges associated with accessing affordable, quality books and managing pre-owned collections. By combining buying, selling, and donating features into one centralized platform, it simplifies the process for users while encouraging sustainable reading practices. The integration of OCR technology for automated cataloging reduces manual effort, while AI-powered recommendation algorithms enhance book discovery by delivering personalized and relevant suggestions to users.

Built with Python, Flask, and MongoDB, the system leverages a modular and scalable architecture that ensures smooth performance, secure transactions, and the ability to grow with user demand. The platform also prioritizes usability, offering a simple and intuitive interface that caters to readers, sellers, and donors alike. With these elements, the system not only improves convenience but also contributes to the efficient circulation and reuse of valuable resources. Beyond its technical strengths, the platform highlights the importance of affordability, accessibility, and sustainability in book circulation. It extends the life cycle of books, reduces unnecessary waste, and fosters a culture of sharing and knowledge exchange.

In conclusion, this project establishes a strong foundation for creating a reliable and impactful platform that bridges the gap between readers and book owners. With opportunities for future enhancements such as mobile app integration, multilingual support, and advanced analytics, the Book Recommendation & Exchange System holds significant potential to evolve into a widely adopted and influential platform in the digital reading ecosystem.

Chapter 10

Future Scope

Expanding the ReadRover with advanced features can further enhance its impact, usability, and reach. Some potential areas for future development include:

1. **Mobile Application Development:** Creating a dedicated mobile app for Android and iOS platforms would allow users to access the system more conveniently. Push notifications for recommendations, new listings, and donation drives could further improve user engagement.
2. **Blockchain Integration:** Implementing blockchain for transaction records can ensure greater transparency, trust, and security in book exchanges, purchases, and donations. It can also be used for creating a decentralized ownership trail for books.
3. **AI-Driven Enhancements:** Expanding the recommendation system with deep learning models and sentiment analysis can improve accuracy and personalization. This could also include multilingual support for recommendations and OCR, making the platform accessible to a broader audience.
4. **Community & Gamification Features:** Adding community features such as discussion forums, reading groups, and gamified rewards (e.g., badges for donations or sharing books) can build a stronger reader community and encourage participation.
5. **Integration of Multimedia Content:** Future versions could integrate book previews, author interviews, and audiobook snippets to provide a richer discovery experience. This would allow users to explore books in multiple formats before making decisions.

References

- [1] Dheeraj Parmar , Pankaj Raghuwanshi. “Collaborative Filtering-Based Book Recommendation System Using Matrix Factorization Techniques: A Comparative Study of ALS and SVD”. International Journal of Software & Hardware Research in Engineering. Volume 13 Issue 6, June 2025
- [2] Ilham Saifudin, Triyanna Widyaningtyas. “Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets”. IEEE ACESSES. 29 JANUARY 2024
- [3] IAN TAI, Souvick Ghosh. “Integrating AI into Library Systems: A Perspective on Applications and Challenges”. 2024