**PROJECT SYNOPSIS**

On

**BOOK EMPORIUM**

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering (2023)

By

Shivani Pandey (1901220100103)

Under the Guidance

Of

Er. Preeti Naval



SHRI RAMSWAROOP MEMORIAL COLLEGE OF

ENGINEERING AND MANAGEMENT LUCKNOW

Affiliated to

Dr. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

**CHAPTER-1**

**INTRODUCTION**

Most organizations have their recommendation system when they sell products online. But almost all the websites are not developed of the buyer interest. The organization force add-on sells to buyers by recommending unnecessary and irrelevant products. A personalized recommendation system (PRS) helps individual users find exciting and useful products from a massive collection of items. With the growth of the internet, consumers have lots of options on products from ecommerce sites. Finding the right products at the right time is a real challenge for consumers. A personalized recommendation system helps users find books, news, movies, music, online courses, and research articles.

In simple terms, the recommendation system is any system that automatically suggest content for website readers and users. These programs emerged as smart algorithms, which can produce results in the form of recommendations for users. They require a large database and a fast computer system that can perform calculations the same between half a second. Various alternatives have been proposed so far today by making recommendations. Machine learning has improved commendation programs, and it brings many opportunities to improve performance of the recommendation program. Machine learning methods use a multiple processing layers to learn hierarchical representations of data.

The technique used by recommender systems is Collaborative filtering. This technique filters information by collecting data from other users. Collaborative filtering systems apply the similarity index-based technique. The ratings of those items by the users who have rated both items determine the similarity of the items. The similarity of users is determined by the similarity of the ratings given by the users to an item. It involves collaborating large volumes of multiple view-point, agents, and sources. It can be applied in mineral exploration, weather forecasting, ecommerce, and web applications where a massive volume of data needs to be processed to make the predictions.

User interests are first analyzed and the result of the user profile analysis is compared with the items available in the system to provide user recommendations  to the user. In this project the user can register or login to their account and will be able to buy or sell their books. They can also play quizzes related to the books. If the book that user wants is currently not available then user can click the notify button so that when it comes in the stock user will get the notification regarding that book.

**CHAPTER-2**

**LITERATURE REVIEW**

**1.Collaborative Filtering with Jaccard Similarity to build a recommendation system**

Avi Rana and K. Deeba, et.al. (2019) proposed a paper “Online Book Recommendation System using Collaborative Filtering (With Jaccard Similarity)”. In this paper, the author used CF with Jaccard similarity to get more accurate recommendations because general CF difficulties are scalability, sparsity, and cold start. So to overcome these difficulties, they used CF with Jaccard Similarity. JS is based on pair of books index which is a ratio of common users who have rated both books divided by the sum of users who have rated books individually. Books with a high JS index are highly recommended.

**2.The Design and Implementation of Books Recommendation System[Yongen Liang, ShimingWan (2018)]**

Individualized recommendation technology is a new technology which can mine products by using stoner’s information, and that match stoner’s preferences through a series of algorithms, so as to achieve better recommendation effect. The number of books in university library is adding fleetly. How to find intriguing books from a large number of books is a problem that every anthology is concerned about. In order to help these druggies find the books that they're interested in, this author designs a books recommendation system grounded on cooperative filtering algorithm The system can principally meet the requirements of druggies to recommend functions, and achieved good results.

**3.** **Book Recommendation System using Machine learning [Fatima Ijaz (2020)]**

Suggestion framework is a common and cold e-commerce issue. Recommendation system performs in many ways including faculty member base on quality, suggestion for reciprocal filtering, and hint for the mix technique. This article proposes a collective suggestion filtering system focused on naive Bayesian approach. The  recommendation method does have a good performance, according to both the undertake experimentation, than numerous prior implementations, including the praised k-NN algorithm being used by suggestion especially at longer length.

**CHAPTER-3**

**PROBLEM DESCRIPTION**

The book readers buy several books and after reading it ,the books are of no use to them.So to get the books utilized by other readers,the user can sell the books or if want to buy can buy the book at comparatively lower price. Recommending books using Machine learning algorithm and buy / sell the book is the main goal of this project. Books are recommended by the collaborative filtering model and we are going to train and build using various features such as user’s rating, book description, book titles etc. User interests are first analyzed and the result of the user profile analysis is compared with the items available in the system to provide user recommendations  to the user. The system we would like to develop will also be able to find an average rating for each cluster and it is going to find top rated books of users from each cluster. All these books shortlisted by our system will be used for training our model . The prediction model needs to be trained so as to produce better results.The user can register or login to buy or sell the books. The user can play quizzes related to the books and the result will be based on other users answers.

**CHAPTER-3**

**PROJECT OBJECTIVE**

To create a book recommendation system for users through which users can enter a book and get similar book recommendations.

**Aims and Objectives:** The book recommendation system should:

a. suggest similar books to the book or topic entered by user

b. provide a clean and minimalist user-interface to avoid any distractions

c. should suggest at least three books for every book entered by user

d. provide various information such as author name, page number, preview link for the book etc.

e. User can buy and sell their books

Also, it should be:

a. portable to use on any browser across multiple devices

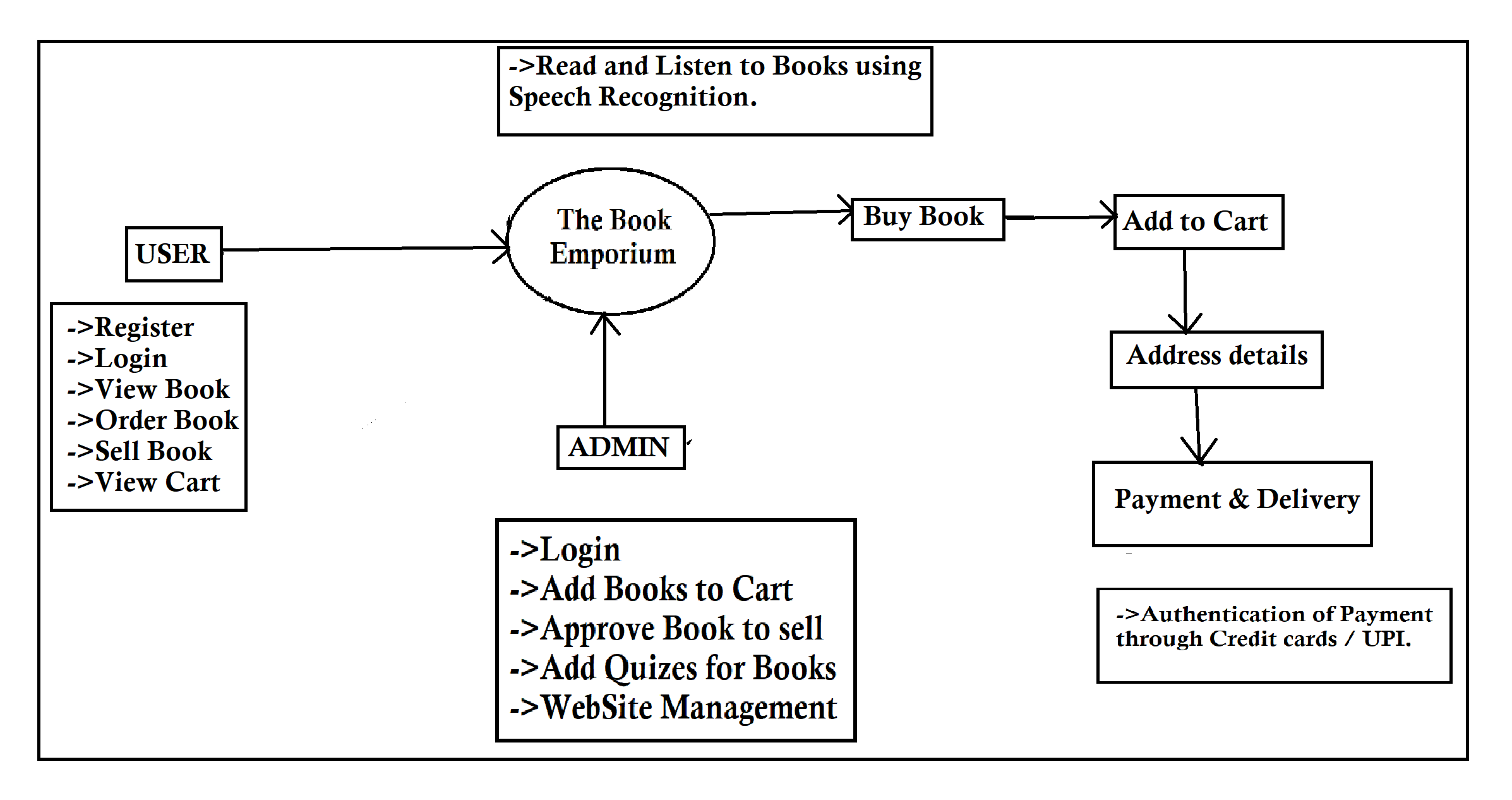
b. efficient and require less resource and time to suggest books

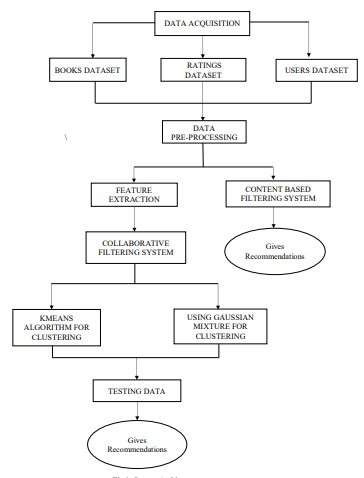
c. scalable to accommodate newer features

d. reusable to make it as a base for other recommender systems

e. fluid to make transitions from one view and page to another not look jittery and laggy.

**BLOCK DIAGRAM**





**CHAPTER-4**

**TECHNOLOGY USED**

***1.Python****:* Python is an interpreted, object- acquainted, high- position programming language with dynamic semantics. Its high- position erected in data structures, combined with dynamic typing and dynamic list, make it veritably seductive for Rapid Application Development, as well as for use as a scripting or cement language to connect being com-ponents together. Python’s simple, easy to learn syntax emphasizes readability and thus reduces the cost of program conservation. Python supports modules and packages, which encourages program modularity and law exercise. The Python practitioner and the expansive standard library are available in source or double form without charge for all major platforms, and can be freely distributed.

**2.Anaconda Navigator***:* Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch operations and fluently manage Anaconda packages, surroundings, and channels without using command- line commands. Navigator can search for packages on Anaconda.org or in a original Anaconda Repository. It's available for Windows, macOS, and Linux.

***3.Spyder****:* Spyder short form is “Scientific Python Development Environment.” It’s intended for use as a workbench for scientific computing with Python, and that’s reflected in the features set, the packaging, and the overall geste of the IDE. Spyder has useful features for general Python development, but unless you work substantially with Python and scientific computing packages, you ’re presumably more off with a different IDE. The biggest reason not to use Spyder as a general- purpose Python development terrain isn’t the point set, but the setup process. Spyder isn't delivered as a standalone executable in the manner of a product like Visual Studio or PyCharm. Rather, it’s installed as a Python package. Your easiest path to Spyder is to install a Python distribution that comes with it preloaded, similar as Continuum Analytics Anaconda.

**4. React.js Front End**

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

React’s strong suit is handling stateful, data-driven interfaces with minimal code minimal pain, and it has all the bells and whistles you’d expect from a modern web framework: great support for forms, error handling, events, lists, and more.

**5. Express.js and Node.js Server Tier**

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, unopinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

By making XML HTTP Requests (XHRs) or GETs or POSTs from your React.js front-end, you can connect to Express.js functions that power your application. Those functions in turn use MongoDB’s Node.js drivers, either via callbacks for using Promises, to access and update data in your MongoDB database.

1. **MongoDB Database Tier**

If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you’re going to want a database that’s just as easy to work with as React, Express, and Node.That’s where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they’re valid) stored directly in MongoDB for later retrieval.

**CHAPTER-5**

**IMPLEMENTATION ALGORITHM**

**5. Book Recommendation Algorithm Based on Collaborative Filtering and Interest**

**5.1. Book Interest Model**

Attributes related to books include search times, borrowing time, borrowing times, borrowing interval, and renewing times.

The proportion of search times of a certain book is the proportion of search times to the search times of all books. For normalization, the proportion is divided into five levels; the formula is as follows:

In which,  is the grade score of the search proportion,  is the number of searches, and  is the total number of searches for all books. The higher the ranking means the higher the score.

The length of the borrowing time can basically reflect the popularity of a book [16]. Of course, the borrowing time may be too long because you forget to return the book, or you cannot return the book because of the holiday. This special case is not considered here for the time being, and the formula is as follows:

In which,  is the grade score of the borrowing time,  is the number of borrowers of the book,  is the longest borrowing time of the book,  is the book return time, and  is the book borrowing time. The higher the ranking means the higher the score.

The number of borrowings can accurately reflect the popularity of the book [17]. The more borrowing times, the higher the popularity. The formula is as follows:

In which,  is the rating value of the number of borrowings,  is the number of borrowings of the book, and  is the total number of borrowings of all books. Divide the number of borrowings into five levels, and rank according to the number of borrowings. The ranking is within 20%, the highest level. The ranking is 81%-100%, the level is the lowest, and the score is the lowest.

The borrowing interval refers to the time interval for a book to be borrowed after being returned . If the time is shorter, the demand for the book is greater, or the popularity of the book is greater. On the contrary, if it is returned and is no longer borrowed, it means that the popularity of the book is very low. The formula is as follows:

In which,  is the grade score of the borrowing interval,  is the number of borrowers of the book,  is the longest borrowing time of the book,  is the return time, and  is the next person’s borrowing time.  is the sum. The higher the ranking means the higher the score.

The number of renewals of a book can also reflect the popularity of the book to a certain extent. The formula is as follows:

In which,  is the grade score of the proportion of the number of renewals,  is the number of renewals of the book, and  is the total number of renewals of all books. The higher the ranking means the higher the score.

Finally, the average of the five indicators is used as a comprehensive indicator of book interest; the formula is as follows:

**5.2. Collaborative Filtering Recommendation Model**

The basic idea of the collaborative filtering algorithm is to find similar users of the current user and predict the current user’s score based on the similar user’s score information to make recommendations. A recommendation system based on collaborative filtering does not analyze information from data but establishes an effective evaluation feedback mechanism to allow users to form a good feedback [19]. In other words, the recommendation users get may not be mined from the data at all but contributed by other users. There are three main steps: collecting scoring data, finding neighbors, and generating a recommendation list.

The user-based collaborative filtering algorithm is to find neighbor users with high similarity for the current user and then recommend items that the neighbor users have rated, and the current user has not rated to the current user. The steps include calculating the similarity between the current user and other users, sorting according to the similarity from the highest to the bottom and the user with the highest ranking as the current user’s neighbor, filtering the current user’s rating items from the neighboring user’s rating list, predicting the current user’s rating of unrated item scoring, selecting the one with the highest score, and recommending it to the current user.

Cosine similarity can describe the linear correlation between two sets of data, and its value range is between -1 and 1. The cosine similarity is calculated based on the set of items jointly evaluated by two users [20]. When using this method to calculate, it is necessary to remove the average value of all commodities evaluated by the user. Generally, the following calculation formula is used to calculate the similarity.

In which,  is the set evaluated by user ,  is the set evaluated by user ,  is the rating of user  on item , and  is the rating of user  on item .

In this paper, the interest degree and collaborative filtering are averaged for a comprehensive analysis.

**5.3. Evaluation Model**

Mean deviation and root mean square error are usually two standards to measure the accuracy of the recommended system.

Divide the attributes into five levels and rank them according to the data. The ranking is within 20%, the highest level; the ranking is 81%-100%, the lowest level, and the score is the lowest.

The formula for mean deviation is as follows:

In which,  is the predicted user rating, and  is the user’s actual rating. The smaller the deviation of the average value, the closer the predicted score of the recommendation algorithm is to the actual score.

The formula for the root mean square error is as follows:

In which,  represents the test data,  represents the size of the test data set,  represents the user,  represents the book,  represents the user’s actual score for the book, and  represents the user’s predicted score for the book.

**CHAPTER-6**

**SOFTWARE AND HARDWARE REQUIREMENTS**

1. **SOFTWARE REQUIREMENTS**
   * Python Version 3.0 or above
   * MERN stack
   * Operating System: Windows 10
   * Database server: MongoDb.
   * Tools: Microsoft Visual Studio, Xampp, Web Browser (Google Chrome or Firefox)
   * Python Libraries: Numpy, pandas, sklearn, Pickle, Matplotlib, Seaborn

1. **HARDWARE REQUIREMENTS**

* RAM: 4 GB or above
* Storage: 30 to 50 GB
* Processor: Any Processor above 500MHz

**CHAPTER-7**

**FUTURE SCOPE**

The System has adequate scope for modification in future if it is necessary. Development and launching of Mobile app and refining existing services and adding more service, System security, data security and reliability are the main features which can be done in future. In the existing system there are only some selected categories, so as an extension to the site we can add more categories as compared to existing site. Also we can add admin side with some functionalities like books management, User management etc.