ST. LAWRENCE COLLEGE

(Affiliated to Tribhuvan University)
Chabahil, Kathmandu

Final Year Project Report

on
“**Book Recommender**”
(CSC-404)

A Final Year Project Report submitted in the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University

Under the supervision of
“Mr.Tika Dahal”
“Lecturer / Supervisor”

Submitted by
Bhabuk Pokharel(T.U Exam Roll No: 5673/071)
Bijay Koirla(T.U Exam Roll No: 5675/071)
Lakpa Sherpa(T.U Exam Roll No: 5680/071)
Madan Shah(T.U Exam Roll No: 5681/071)

Submitted to

ST. LAWRENCE COLLEGE
Department of Computer Science
Chabahil, Kathmandu
September 19, 2018

DECLARATION

Books Recommender which is being submitted to the Department of Computer Science and Information Technology, St. Lawrence College, Chabahil, Kathmandu, Nepal for the fulfillment of the seventh semester as major project under the supervision of Mr. Tika Dahal.

This project is original and has not been submitted earlier in part or full in this or any other form to any university, here or elsewhere, for the award of any degree.

Submitted By:

BHABUK POKHAREL

BIJAY KOIRALA

LAKPA SHERPA

MADAN SHAH

RECOMMENDATION

This is to recommend that Bhabuk Pokhrel, Bijay Koirala, Lakpa Sherpa and Madan Shah have carried out research entitled "Book Recommender" for the fulfillment of seventh semester under my/ our supervision. To my/ our knowledge, this work has not been submitted for any other degree.

The team has fulfilled all the requirements laid down by the Tribhuvan University Department of Computer Science and Information Technology, Kirtipur, Kathmandu.

MR. Tika Dahal

Supervisor / Lecturer

St.Lawrence College

Chabahil, Kathmnadu, Nepal.

LETTER OF APPROVAL

September, 2018

On the recommendation of MR. Tika Dahal, this project report submitted by MR. Bhabuk Pokhrel, MR. Bijay Koirala, MR. Lakpa Sherpa, MR. Madan Shah, entitled “Book Recommender” in partial fulfillment of the requirement for the award of the Bachelor’s degree in Computer Science and Informational Technology is a bonafide record of the work carried out under my/our guidance and supervision at ST. Lawrence College, Kathmandu.

EVALUATION COMMITTEE

Mr.Deepak Thakur

Program Coordinator,

St.Lawrence College

Chabahil, Kathmandu Nepal

Acknowledgement

We must take this opportunity to acknowledge our sincere gratitude to the St. Lawrence College for providing quality education in the field of Computer Science and Information Technology which helps the students to broaden the concept of Computer Science and Information Technology.

In addition, we would like to give a word of thanks to entire member of IT department of St. Lawrence College, who encouraged us to perform work activities.

Abstract

Book recommendation is the system which provides the recommendation to the user on the basis of the choice of the user. It uses different algorithms to find the best result which can be recommended to the user. The system analyzes the users and their preferences when the user starts rating the book. Book recommendation system were carried out by the team of members, who were devoted to their respective field to develop the best possible. Book recommendation system uses different programming languages, hardware platforms, software platforms to make the system as a whole. During the development phase different modules were made separately by different members and were integrated at last. Along with it testing were done. After the system has been developed we found different algorithms gave different level of correctness. The platform where it was deployed as played a major role in the smoothness of the program.

LIST OF ACRONYMS AND ABBREVIATIONS

SQL	Structure Query Language
JSON	JavaScript Object Notation
PHP	PHP Hypertext Preprocessor
KNN	K-Nearest Neighbor
API	Application Program Interface
JS	JavaScript
CSS	Cascading Style Sheet
HTTP	Hypertext Transfer Protocol
ML	Machine Learning
RE	Recommendation Engine
RMSE	Root Mean Square Error

List of Figures

List of Table

Table of Contents

CHAPTER 1

1.1 Introduction

Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information per user’s preferences, interest, or observed behavior about item.

Recommender systems have become extremely common in recent years. They are applied in a variety of applications. The most popular ones are probably movies, books, news, music, research papers, social tags, and products in general. Most recommender systems typically produce a list of recommendations in one of the three ways- through collaborative-filtering technique, content-based technique, and hybrid algorithm.

Books recommendation system is a web application to provide user with all books information that they are trying to search for. The objective of Book recommendation system project is to create recommendation system which can easily be used in websites or web applications. It mainly focusses e-commerce websites. Our book recommendation project focus those websites to recommend books for the users. Our recommendation engine will help customers find information, products, services they might not have thought of. Book recommendation system has used strategy like knn alogirthm and popularity algorithm. It is helpful in a wide variety of industries and businesses. Book recommendation system analyzes the past user's rating and provides the recommendation according to it.

1.2 Problem Definition

Books are the sources of the knowledge. We all know the room without a book is like the body without a soul. We love to read the books we like in the time we can manage. It's like a hobby or a passion for the many people to read new book. But in this tight schedule we are not able to find the proper book to read that matches our mood and the choice. We try to consult with the friends or the family to get the idea about the stores where we can find the books and the recommendation to our mood types. But always asking for the help with the others might not help in these kinds of situations because different people have different choice of their own. Other might now understand your choice and recommend you with the complete opposite books. There are many book recommending sites, for example amazon which is business oriented. They cannot or will provide the best possible suited result for the user because they don't focus on the user's choice but on the business. These problems are important because there are many genuine readers out there who want to read the books according to their choices but are unable to find it fast and accurate.

1.3 Objective

The objective of this system is to provide a precise recommendation of books for individual according to their choice and rating by the use of best algorithm. Our system is solely developed for the user. However, it also aims to serve the following objectives:

a) Helps to find the users the books according to their needs pattern and provide the best result.

b) To allow a user to rate the books they have completed reading for the other readers in the future.

1.4 Scope

The project is based on Recommendation Engine using one of the machine learning approaches, to improve performance and accuracy. The project intends to create a useful knowledge base from the data that will be useful for decision making.Users can get recommendation through a mathematical analysis of their behavior.

Our recommendation system is the academic project so we have the lack of resource. We need a larger server to run the programs smoothly and efficiently. When the large number of data are manipulated then off course it needs the more processing power which we can in our project. We have tested our project in the limited platform.

1.5 Significance of the Project

Books are the great source of information and knowledge. How much we adore the technological way the best way is to read the books at last. It’s fun and cheap way of acquiring the knowledge and information that we love to acquire. At deep, the project also exercises three important domains that can be executed independently in larger applications. Implementation of various machine learning approaches as Cosine Similarity to find similarity among two objects and Brute force search to find points nearest using KNN are the implication of this project.

1.6 Project Feature

Book recommender system lists recommended books. User can select the books they like in this application. Beside this, users are also privileged to like, rate particular book to show their opinion about that book.

Books recommender system is a very simple and user friendly. We have created as far as simple graphical user interface for the ease of the user. All the functions related to the books like searching, ratings are easily be done with simple way. The page contains the functions that are easy to use for any kinds of users.

CHAPTER 2

2.1 Literature Review

2.1.1 Existing System

The purpose of this system is to create an engine which can recommend books to the user on the basis of their past ratings to books. Many of the individuals and groups have worked and also been working on to develop an accurate and efficient recommender system. Though, this project aims to provide distinct features, it is obviously not the first one. It would have been impossible to proceed without the study of previously developed similar type of projects and system. The projects developed in other countries have focused on the need according to their environment while the projects developed in Nepal have tried to justify the problem in our circumstances and provide a solution for the same. Thus, study and analysis of few of such project was a surplus input to our project. The Books Recommendation System has used various principles and techniques to recommend by developing a predictive model of users' and rating about the books. Using available dataset, the system extracted features of the user preferences and made use of collaborative filtering algorithm. Book recommendation system implemented Nearest Neighbor Search algorithm as K-nearest neighbor, and several other learning algorithms. Similarly, Preference-based Books Recommendation System is a books recommendation system for individuals and group in accordance to their fondness. This system also maximized minimum happiness across a group of users, as an alternative to other group recommendation systems where the most commonly recommended books across individual users is selected.

2.2 Requirement Analysis

2.2.1 Functional Requirements

The functional requirements of book recommender system are as follow:

Recommend books to the user using mathematical analysis of their behavior.

Book recommendation system should allow users to rate books.

In case of cold start, book recommendation system should recommend most popular books to the users.

Book recommendation system should allow users to register and rate books.

Allow user to view site without logging to the site.

Book recommendation system should create a useful knowledge base from the data that will be useful for making recommendation.

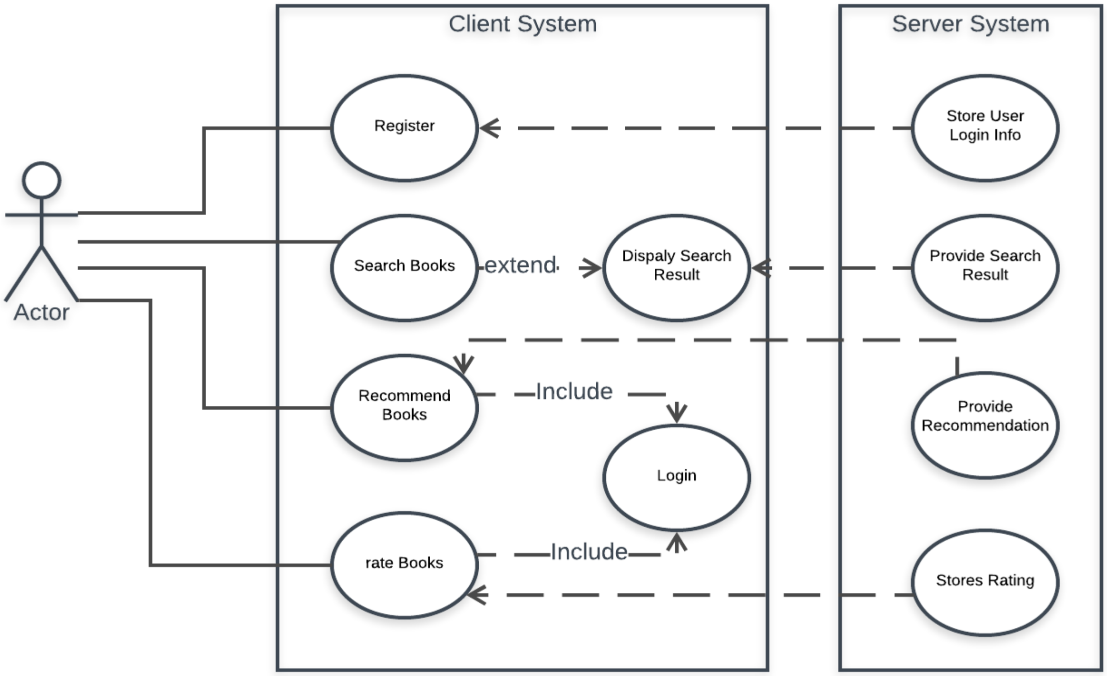


Figure 1 : Use Case Diagram

2.2.2 Non-functional Requirements

The non-functional requirements are:

Performance

Book recommender should recommend the users promptly. It should be efficient in resource utilization like memory, CPU, storage etc.

Scalability

Book recommender should be scalable i.e. it should be able to server large number of users as per demand.

Reliability

Book recommender should provide reliable recommendation to the users. It must recommend book based on KNN algorithm.

Recoverability

In case of failure of book recommender, there should be some mechanism to recover the system. Efficient backup plans must be made.

Usability

Book recommendation system can be used by any user by registering to the site.

Interoperability

Book recommendation system is built by integrating php, python, and varieties of APIs. So, they must operate together.

2.3 Feasibility Analysis

Feasibility study provides viability and quality information for decision making. A well conducted feasibility study has been done to evaluate that the project has potential for success. Thus, following feasibility analysis is taken into consideration for the book recommendation system:

2.3.1 Economical Feasibility

Software to be used in the development of the book recommendation system are MySQL database server, Apache web server which are open source and available for free. For computing the data frame using cloud Virtual Machine, free azure subscription is available to us. Hardware like computer, switch to be used in this project are personnel belonging.

2.3.2 Operational Feasibility

Book recommendation system is designed to provide book recommendation to users based upon their preference with some level of error. It is planned to a web-based recommendation system with a very simple user interface. It can be easily used by normal users. There is desire for this kind of system as Nepal does not have book recommendation system.

2.3.3 Technical Feasibility

Book recommendation system is to be developed using php, Laravel, python and MySQL our team comprise of people who can develop web application using Laravel, PHP and Python. Doing research and self-study we plan to develop a recommendation system based on python. We have technical knowledge of MYSQL database and APIs.

2.3.4 Schedule

The Gantt chart of the book recommendation project is as follow:

Weeks	1	2	3	4	5	6	7	8	9	10
Project Activities										
Planning and design	■									
Development		■			■	■				
Evaluation			■				■			
Revise				■				■		
Testing									■	
Implementation										■

Figure 2 : Gantt Chart

The schedule of the book recommendation system has six major activities. Planning and design is scheduled to be completed within first week of starting the project. Development phase starts in second week. RAD methodology is selected for system development. So, development, evaluation and revise will continue in loop until seventh week. Testing is to be done in eighth week and implementation is to be done in ninth week.

2.4. Structuring System Requirements

2.4.1. Data Modeling - ER Diagram

Book recommendation system has three entities: user, book, rating. User has attributes like ID, name, location. Book has attributes ISBN, author, publication. Rating has attributes like ISBN, ID, rating.

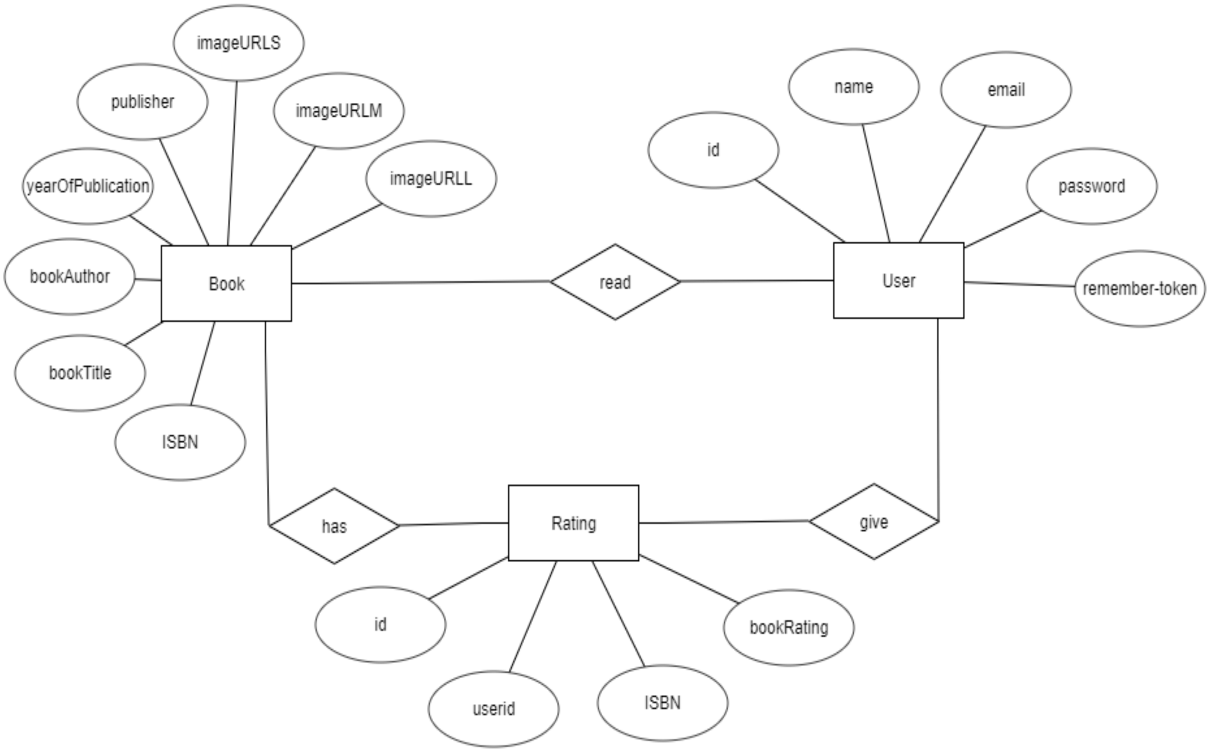


Figure 3: ER Diagram

2.4.2. Process Modeling - DFD

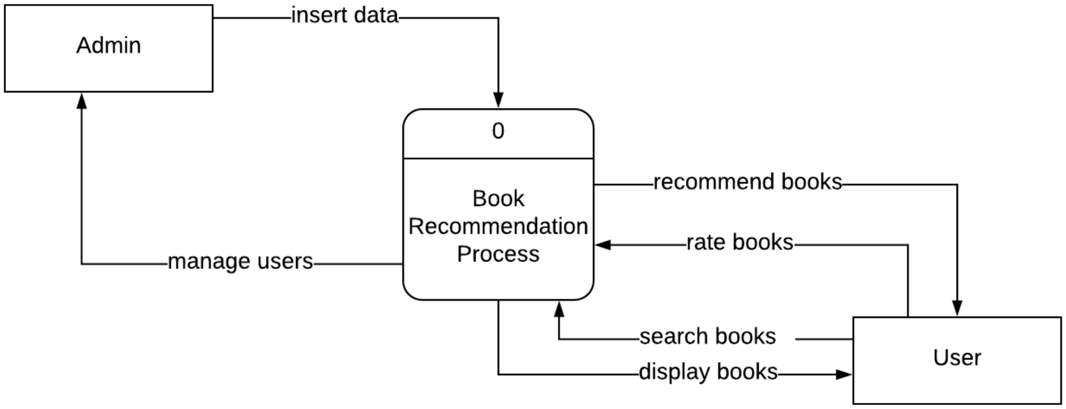


Figure 4 : 0 Level DFD

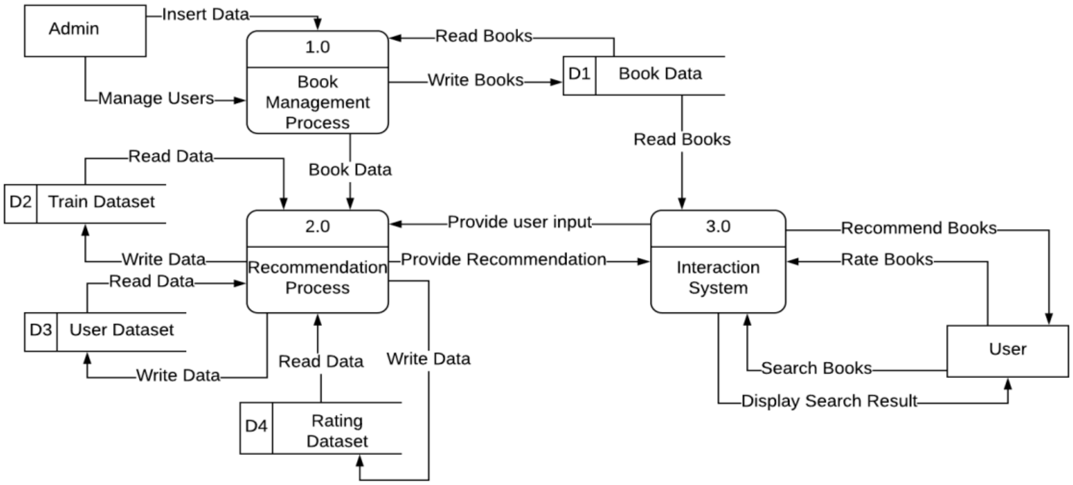


Figure 5 : Level 0 DFD

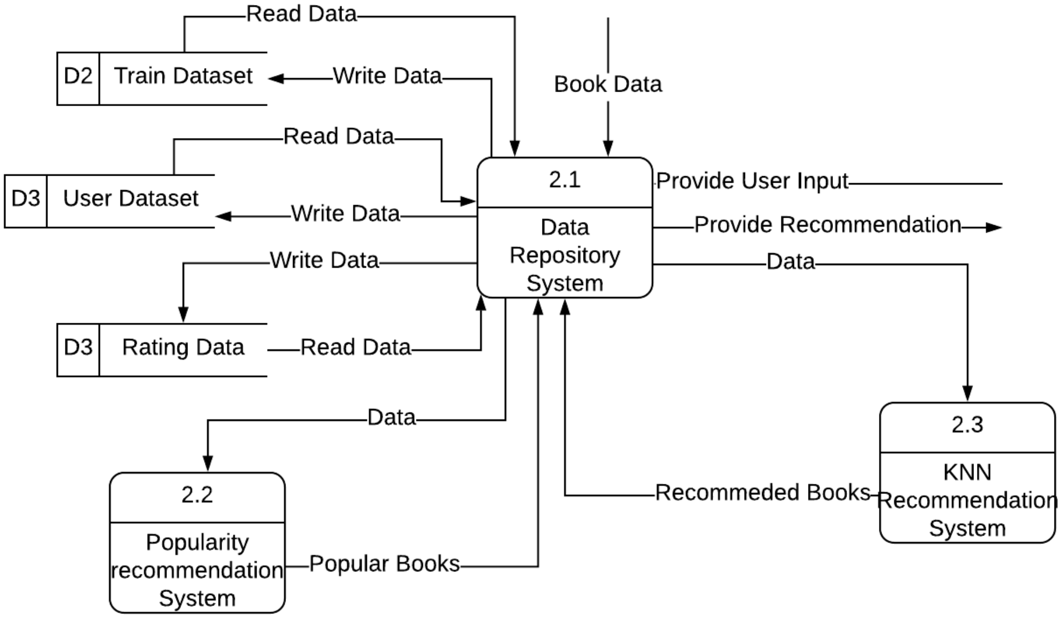


Figure 6: Level 1 DFD of Recommendation Process

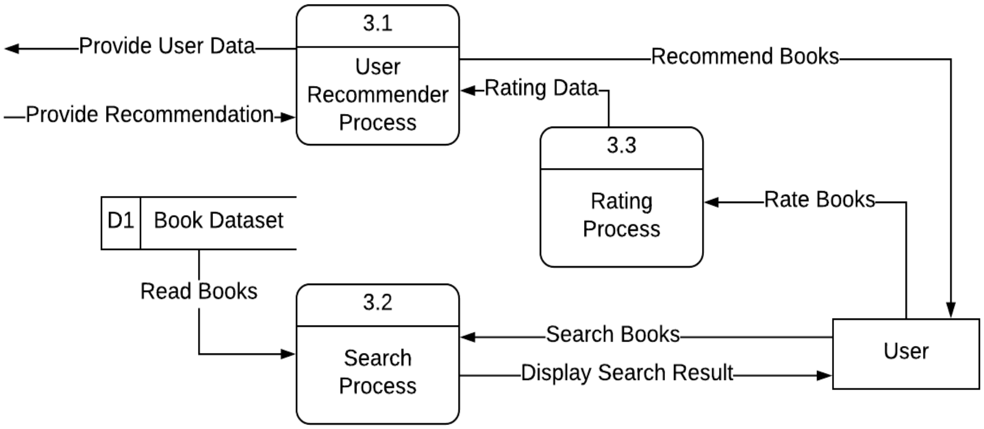


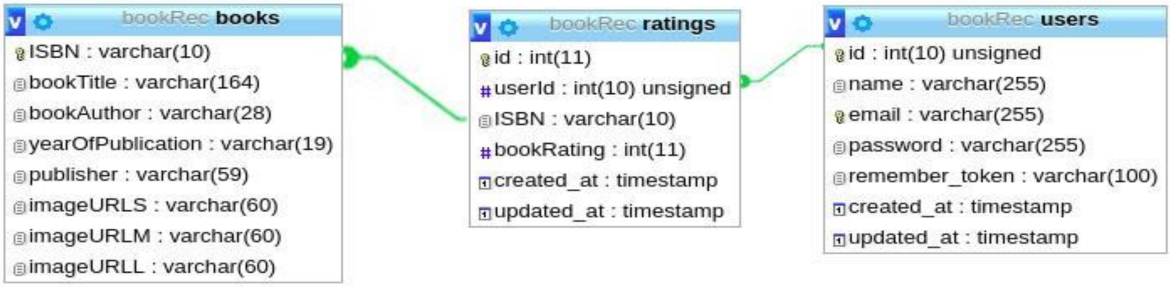
Figure 7: Level 1 DFD of Interaction System

Chapter 3

3.1 System Design

3.1.1 Database Schema Design

Book recommendation system uses database to store data used in the system. It comprises of 3 tables. They are: Books, Ratings, Users.



bookRec **books**

ISBN : varchar(10)
bookTitle : varchar(164)
bookAuthor : varchar(28)
yearOfPublication : varchar(19)
publisher : varchar(59)
imageURLS : varchar(60)
imageURLM : varchar(60)
imageURLL : varchar(60)

bookRec **ratings**

id : int(11)
userId : int(10) unsigned
ISBN : varchar(10)
bookRating : int(11)
created_at : timestamp
updated_at : timestamp

bookRec **users**

id : int(10) unsigned
name : varchar(255)
email : varchar(255)
password : varchar(255)
remember_token : varchar(100)
created_at : timestamp
updated_at : timestamp

The tabular structure of database schema is as follow:

Entities:

User: The client for the system

Books: Books registered in the system Rating

Ratings: The ratings of a user for a book

Table Attribute Description:

Table 1: User Table

Attribute	Data Type	Description
Id	Int	Unique identity of user (primary key)
Name	Varchar	Username of the user
Password	Varchar	Password of the user
Email	Varchar	
Created_at	Timestamp	
Updated_at	Timestamp	

Table 2: Book Table

Attribute	Data Type	Description
ISBN	Varchar	unique identity of books (primary key)
bookTitle	Varchar	Books name
bookAuthor	Varchar	books Author
Year of Publication	Varchar	Books Publication years
Publisher	Varchar	Books Publisher name
Image URLS	Varchar	Small Image URLS
ImageURLM	Varchar	Medium Image URLS
ImageURLL	Varchar	Large Image URLS

Table 3: Rating Table

Attribute	Data Type	Description
ID	Int	Identity for Ratings
UserID	Int	User identity (foreign key)
ISBN	Int	Books identity(primary key)
bookRatings	Int	book Ratings by User
Created_at	Timestamp	
Updated_at	Timestamp	

3.1.2 Interface Design

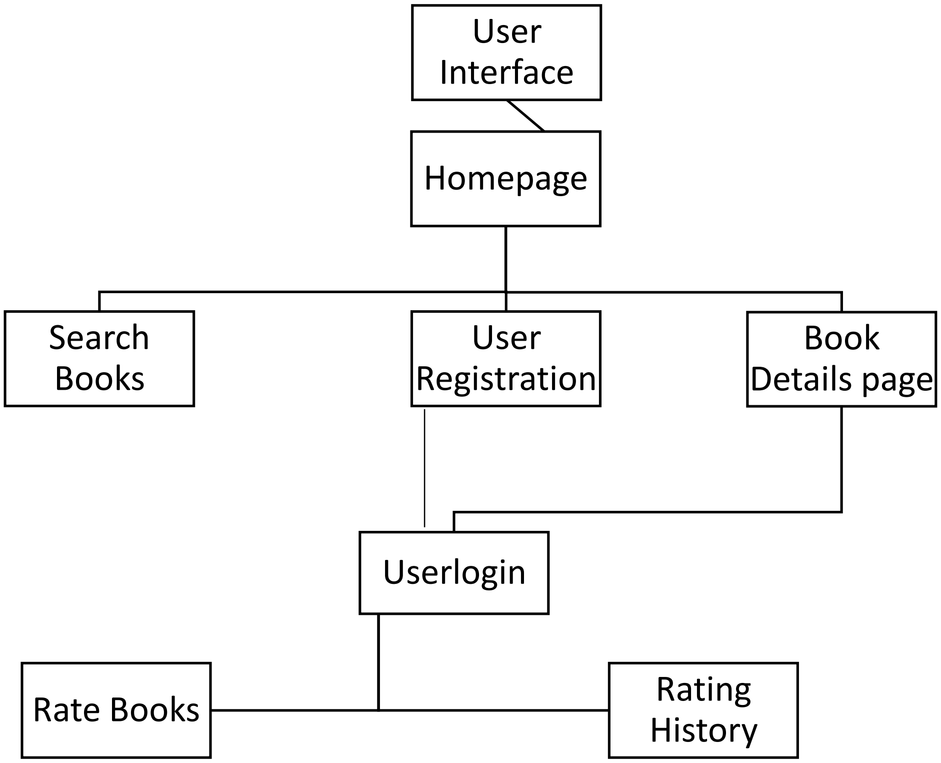


Figure 8: User Interface

User interface has been designed using html, CSS, bootstraps, JavaScript. Our interface consists of user login form, registration form, user dashboard, guest dashboard, rating interface.

User Login Form

It is a login interface where user logs into the book recommendation system. It asks for user’s email address and password. User can successfully login to the system if and only if the user name and password matches the data available in the registration table.

Registration Form

User use this interface to register themselves to the book recommendation system. To register user needs to provide user’s email address and password. Once registered user can login the system.

User dashboard

It is the dashboard for the registered user who has logged into the system. It shows four types of book categories: all, new, featured and recommended. We can access different categories by clicking the button. Recommended book are displayed according to KNN brute force algorithm that uses cosine similarity model.

Guest dashboard

It is the dashboard for user who has not logged into the system. It is similar to user dashboard. The only difference is the recommendation category of book. It shows recommendation based on popularity of books.

Rating Interface

It is the interface where user can rate the books. User need to be logged in to rate the books. User can rate books where 1 is minimum represented by half star and maximum is 10 represented by five starts.

3.1.3 Dialogue Design

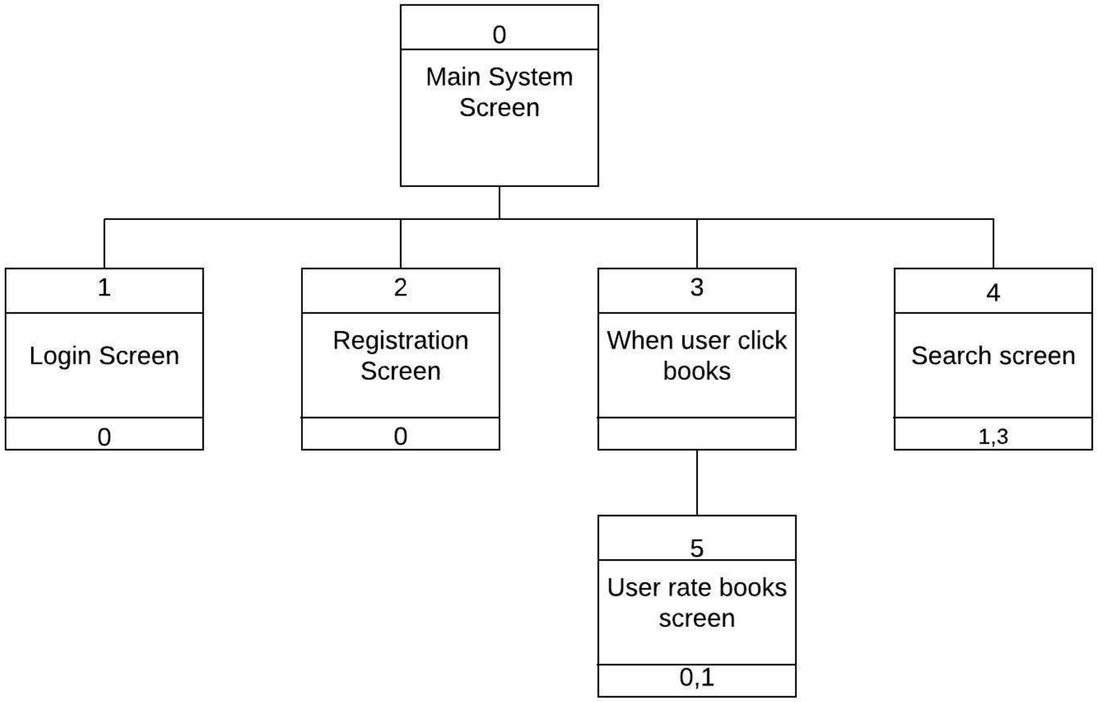
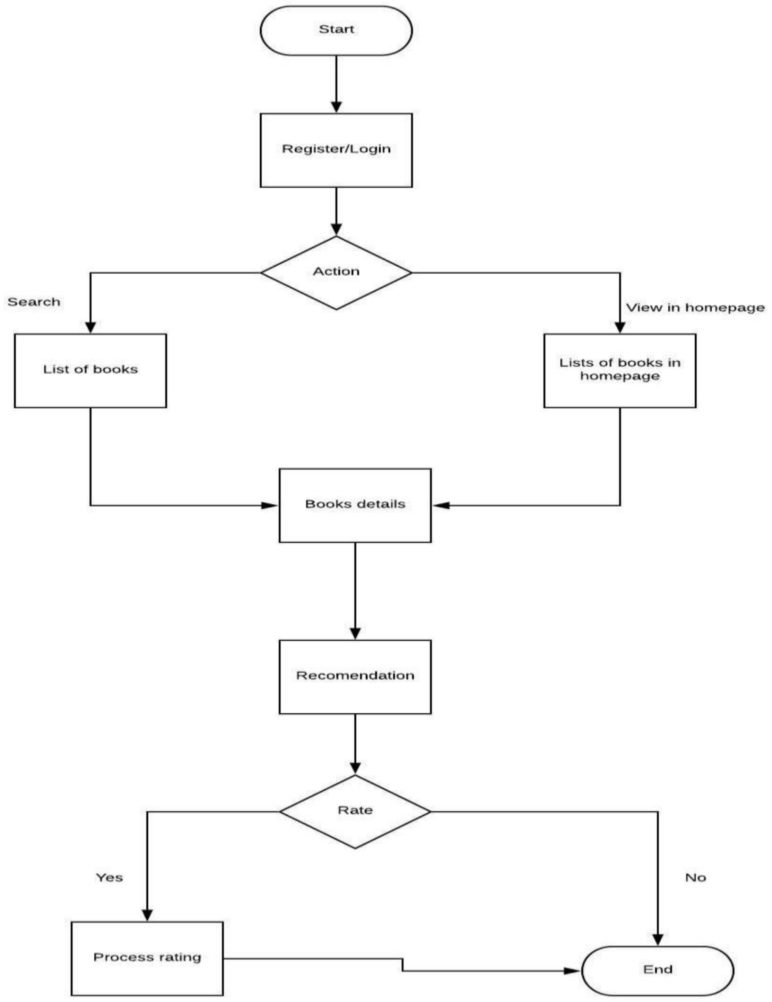


Figure 9: Dialogue Design

Dialogue Diagram are those steps that demonstrate the straight through path. In our system first user enter in home page (Main System screen), if user have their own account then user can go to the login page and login our system. If users has no account in our system then user go in the Registration page and register their information and login. When guest user enters any book from our system, firstly books detail page is opened and if user wants to rate the book then login page in opened. If user want to search book then user clicks on the search button and the search screen appears and users can search the books.

3.1.4. Process Design



The process model defines the overall recommendation system. Here each and every process of book recommendation system is defined. Every process are related to each other and every unit works together to form a good book recommendation system.

CHAPTER 4

4.1 Implementation and Testing

4.1.1 Implementation

Rapid application development is a form of agile software development methodology. Unlike Waterfall methods, RAD emphasizes working software and user feedback over strict planning and requirements recoding. In other words, RAD is less talk, more action and testing, lots and lots of testing. While RAD de-emphasizes strict planning, there are still a handful of steps or phases each development project goes through when using the rapid application development methodology.

a) Implementation in Python

In book recommending system to communicate between the web app and the RE server, first we need to create API. This API was created in Python using Flask framework along with Gunicorn server. After the API was created, the web server, which serves the client will request the recommendation server with a URL. Once the request is made, the server responds with the data in JSON. This JSON data will then be parsed and displayed in the client machine.

b) Implementation in Jupyter Notebook

In addition to displaying/editing/running notebook documents, Jupyter Notebook was used as major component during the development of Recommendation Engine in this project. It was also used for the reporting and visualization of RE results.

c) Implementation in PHP and Laravel

In book recommending system we used PHP as main scripting language with it's Laravel framework in Apache web server for building web application. It handles all clients request, including providing best recommendation taken for recommendation server through API.

4.1.2 Tools used

During the project, many different technologies were used. For front-end development of web application HTML, CSS and JS were used. Moreover, most popular framework like Bootstrap (CSS), Jquery(JS) were used for rapid and responsive development of common tasks with fewer lines of code. We can use front-end frameworks like Reactjs and Angularjs to develop web app, but for this project, only JS is used.

A JSON Array or Object was used often to transport data between the recommendation server and the Laravel application. The JSON results were further treated and used by the application. MySQL is used as a database at the web server. Our Laravel application will communicate with the API Controller with necessary parameters and communicates with MySQL database and will fetch the result, and return the results to client requesting it.

For building RE, Python (3.5) is used as core programming language which helped in data wrangling, visualizations and allowed to implement ML algorithms.

4.1.3 Description of methods

	User ID	ISBN	Book Rating	Book Title
0	2214	0345412X	0	The Notebook
1	2256	0345413X	5	The Notebook
2	4587	0345414X	0	The Notebook
3	2236	03454155X	5	The Notebook
4	2215	0345419X	9	The Notebook

Building Dictionary of Critics

For this project, different users and their preferences were represented in a data frame by importing from CSV files in Python by using Pandas. Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools. This data frame has ranking from 1 to 10 as a way to express how much each of these books critics (and the user) rated a given book. Sample critics data frame is given below:

Cleaning data

DataFrame was used because they are convenient for experimenting with the algorithm and for illustrative purposes. It provides efficient way to search and modify the DataFrame through list of methods. We can handle missing values (say NA or NaN) using Pandas and Numpy. We have created a DataFrame with missing values. In the output, NaN means Not a Number. After finding the NaN values, we have cleaned those NaN and NA values which helped to minimize the DataFrame.

Filter to users in US and Canada only

In order to improve computing speed, and not run into the “MemoryError” issue, we will limit our user data to those in the US and Canada. And then combine the user data with rating data and total rating count data.

	User ID	ISBN	Book Rating	Book Title	Total Rating count	Location
0	2214	0345412X	0	The Notebook	60	Texas, usa
1	2256	0345413X	5	The Notebook	60	Ohio, usa
2	4587	0345414X	0	The Notebook	60	Missouri, usa
3	2236	03454155X	5	The Notebook	60	Missouri,usa
4	2215	0345419X	9	The Notebook	60	Origon,usa

Finding Nearest Neighbour

We have used unsupervised learner for implementing neighbor searches provided by sklearn.neighbours package. NearestNeighbors in this package implements unsupervised nearest neighbors learning technique. We used “brute” as nearest neighbors algorithm to compute the nearest neighbors, and specified “metric=cosine” to use for distance computation.

Finally, we fit the model as

```
from  sklearn.neighbors  import NearestNeighbors
model_knn = NearestNeighbors(metric = “cosine” ,
algorithm=”brute”)model_knn.fit(us_canada_users_rating_matrix)
```

Serialize model for deployment of Model

Our model if followed with serialization process for converting an object into a stream of bytes to store the object or transmit it to memory, a database, or a file. Its main purpose is to save the state of an object in order to be able to recreate it when needed. Python has pickle package which provides dump method to pickle any python object for serialization process.

Thus created recommender model is serialized to be deployed in production as APIs using Flask.

```
 with open(‘pickles/knn_item_model.pickle’, ‘wb’) as f:

pickle.dump(knn_item_model, f)

pickle_in = open(‘pickles/knn_item_model.pickle’ , ‘rb’)

knn_item_model = pickle.load(pickle_in)
```

4.1.3 List of Algorithm

4.1.3.1 Popularity model

Popularity model is a common baseline approach. Popularity model used in the book recommendation system is not actually personalized - it simply recommends to a user the most popular items that the user has not previously consumed. The book displayed to user depends upon the sum of total rating of books. The top ten popular books are provided to user.

Algorithm for generating recommendation for user A,

a. Create a DF of critics of every user and their preference.

b. Group a DF by book's ISBN.

c. Sum book rating of each book.

d. Sort book DF with book rating in descending order.

```
e. Compare user A's preference with popular books DF:
for each book in popular_books:
        if book not in popular_books:
                recommendation = recommendation.append(book)
f. Return the recommendations for user A
```

4.1.3.2 KNN model

The recommendation model is built with NearestNeighbors, which is unsupervised learner for implementing neighbor searches. It is based on the similarity score which is calculated using cosine similarity. Here, Suppose A, and B be two users and Bi are the books, where i = 1, 2, 3…. n. The value of n equals the total no. of books. Assuming, we have following data from users:

Given,

Books	B1	B2	B3	B4
User A	3.0	4.0	3.5	4.5

Now, at first a dataframe of critics is created, which is a nested dictionary of user’s rating. Assuming, there are only two users now,

critics= {‘A’:{‘B1’:3, ‘B2’:7, ‘B3’:8, ‘B4’:4}, ‘B’:{‘B1’:5, ‘B2’:9, ‘B3’:6, ‘B5’:10}}

Now if the user B wants recommendations, then the recommendation model compares it with every other users’ in the critics based on the similarity score calculated using cosine similarity.

Algorithm for generating recommendation for user A,

a. Create a dataframe of critics of every user and their preference.

```
b. Create 2D matrix of user and books filling values with their relevant ratings.
c. Compare user A with every other user in critics using KNN based on distance.
If distance > 0:
I) Sort distance in descending order
II) Compare books that user A has not rated but similar user has rated,
                i) score[i]= distance * rating of book from similar user
ii) Calculate average scores for scores calculated for each book,
                        avg_score = sum(score)/ len(score)
iii) Sort avg_score of books in descending order
III) Rank the books according to their avg_scores
IV) The recommendations for user A is the books within the top N rank.
recommendation = rankings[:N]
d. If len(recommendations) == 0:
i)return the top N most popular books from the data
recommendation = topBooks[:N]
e. Return the recommendations for user A
```

4.1.4 Lists of major classes

KNN User Recommender Class

The main purpose of this class is to process and provide recommendations based on KNN user-user based CF. It takes fitted KNN model and provides recommendation bases on the distance of similar users. It has attributes like

MODEL_NAME: Stores model name i.e KNN

us_canada_user_rating: Stores user canada ratings DataFrame

user_rating_pivot: Stores pivot data of dataframe

Methods are:

get_model_name() : Return model name

getSimilarUsers(user_id) : Return similar user of given user

recommend_users(user_id): Return recommended users of given user

recommend_items(user_id, books_to_ignore, topn, verbose): Return recommended items of user. It may take books_to_ignore, topn and verbose as parameter

Popularity Recommender Class

This class provides popular items as recommendation by computing booking rating count DF. It has two attributes i.e

MODEL_NAME: Stores model name i.e popularity

popularity_df: Stores DF of books with their total ratings

Methods are:

get_model_name():Return model name

recommend_items(user_id, books_to_ignore, topn, verbose): Return recommended items of user. It may take books_to_ignore, topn and verbose as parameter

4.1.5 Evaluation

The book recommendation system aim is to provide recommendation of good items to users which is likely to accept. Our evaluation aims at recommending some books to the user is recommending the most relevant items to a user. We chose to perform an automatic evaluation to assess the quality of the computed recommendations. The main focus of the evaluation lies on assessing the accuracy of the computed recommendations. This is due to the fact that such an automated offline evaluation is able to assess the quality of the algorithms fast and efficiently.

The evaluation of the proposed approach is done via a Holdout method. One key aspect of evaluation is to ensure that the trained model generalizes for data it was not trained, on using Cross-validation techniques. We are using here a simple cross-validation approach named holdout, in which a random data sample (20% in this case) are kept aside in the training process, and exclusively used for evaluation. All evaluation metrics reported here are computed using the test set.

Metrics

In Recommender Systems, there are a set metrics commonly used for evaluation. We chose to work with Top-N accuracy metrics, which evaluates the accuracy of the top recommendations, provided to a user, comparing to the items the user has actually rated in test set.

The Top-N accuracy metric chosen was **Recall@N** which evaluates whether the rated item is among the top N items (hit) in the ranked list of 101 recommendations for a user.

This evaluation method works as follows:

- For each **user**
    - For each **item** the user has rated in test set
        - Sample 100 other items the user has never rated.
        - Ask the recommender model to produce a ranked list of recommended items, from a set composed one rated item and the 100 non-rated ("non-relevant!) items
        - Compute the **Top-N** accuracy metrics for this user and rated item from the recommendations ranked list
- Aggregate the global Top-N accuracy metrics

Result

In this section, the results of the previously described evaluation are presented. Firstly, the performance of the different recommendation model is presented and discussed.

We performed the evaluation of the KNN Recommender model, according to the evaluation method described above. Evaluating this Collaborative Filtering model (KNN), we observe that we got average **Recall@5** of 0.37, which means that about 37% of interacted items in test set were ranked by KNN Recommender model among the top-5 items (from lists with 100 random items). And **Recall@10** was even higher **(43%)**, as expected. This model was able to provide **Recall@10** of 1.00 **and Recall@5** of 0.93 at some situation where as it can also give 0.00 at both **Recall** for some user.

Evaluating the Popularity model, we observe that we got **Recall@5 (24%)** and **Recall@10 (31%)** values lower than KNN Recommender model. It is surprising to us that usually Popularity models could perform so well which can be used along with CF method or partly solve cold start problem.

4.2 Testing

4.2.1 Unit testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Book recommendation system contains different types of individual parts that are tested. Some of the test cases are:

Test Case I:

Table 4: User registration

Steps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	Navigate to register page		User should be able to register	Successful registration	Pass
2	Input name(100 max char)	Name : abc123			
3	Input email address(255 max char)	Email id: example@gmail.com			
4	Input password(min 6 char)	Password: xyz123			
5	Re input conformation password(match with first one)	Password : xyz123			
6	Click on register				

Test Case II:

Table 5: User login test

Steps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	Navigate to login page		User should be able to login by the same username and password previously created.	Successful login	Pass
2	Input email(valid email)	User : example@gmail.com			
3	Input password(valid password)	Password: xyz123			
4	Click on login				

Test Case III:

Table 6: Book Search

sSteps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	Navigate to search bar		Result	Error: sorry no data for “123vv” on this database.	Fail
2	Type the book name	Book name:123vv			
3	Press enter				

4.2.2 Integration Testing

Integration testing combines the individual components of the system and checks it for any error. Book recommender system has also gone through the integration testing, some of the test cases are:

Test Case I:

Table 7: User rating log

Steps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	User login	User name and password	Update in database	Updated database	pass
2	View the book				
3	Rate the book				
4	Can be managed in rating page.				

Test Case II:

Table 8: User login check

Steps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	Visit the website without logging		Login page appears	Login page appears as expected.	Pass
2	View the book on the page	Username and password			
3	Rate the book				
4	Login page appears				

4.2.3 System Testing

System testing is the overall testing of the system after integrating all the function of the project. When all the function of the book recommender system is integrated then system testing is done.

Test Case I:

Table 9: Different Recommendation test

Steps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	Go to login page	Username and password (1)	Different recommendation to different users	Different recommendation	Pass
2	Getthe recommendation				
3	Logout				
4	Again login by another username and password	Username and password(2)			
5	Get the recommendation				

Test Case II:

Table 10: Cold Start Checking

Steps	Test Step	Test Data	Expected results	Actual results	Status(pass/fail)
1	Visit the website		Showing the recommendation by popularity for the new users.	Shows the recommendation according to the popularity model.	Pass
2	Register for new user	Email:xyz1@gmail.com Pass: 123123			
3	Login by the same user name	Email:xyz1@gmail.com Pass: 123123			
4	Go to the recommendation page				
5	View the recommendation on the page.				

Chapter 5

5.1 Maintenance and Support

5.1.1 Maintenance

Book recommendation system is implemented in the local server. Our system is implemented within 3 servers' Web server, ML recommendation server and MySQL server. When the new users are added to the database, all the training set for the new users are needed to be maintained. In these types of cases we need to train the database time to time. Since our server is locally implemented and having the low memory, we might face the memory error in frequent amount of time and we need to maintain the server frequently. We need to clear the memory manually.

5.1.2 Support

We have used different kind of models in our book recommender system likes popularity model and KNN recommendation model. We don't have enough processing power for computing the models. We have the minimum processing power and minimum memory resource for the data training so with the support of azure cloud service we were able to compute the models used in the projects.

Chapter 6

6.1 Conclusion

We proposed a Book recommender system for book in Book management system and the according recommendation and ranking algorithms. The comprehensive evaluation showed that such a recommender system is capable of providing the users of our platform with suitable recommendations for books reaching recall values of about 43%. These recommendations aim at helping to find the books according to user's pattern which contributes to better recommendation result.

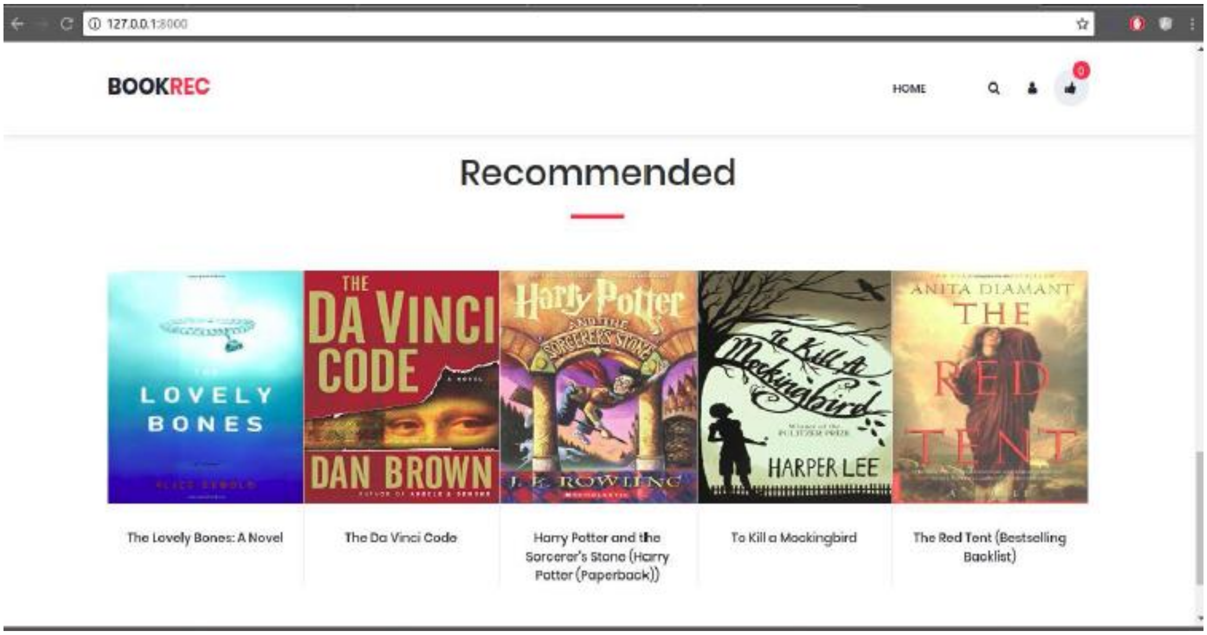
This system is created on schedule with the help of all the team members. All the members have contributed through their heart to the project. All the project members completed the task and kept the good communication with the other team members for the betterment of the project. Once the project has been completed, the project achieved the great success in the objective. Many of the help needed for the project was gathered.

Like our book recommender system, many other website gives this service as well. But others systems are business oriented and ours different. Our project is able to contribute for the real book readers because it made easy for them to find the books according to their choice. All of our objectives for the project have been achieved. We focused mainly on our objectives and we were fully able to accomplish all those objectives. There is no any of our objectives that we are not able to meet in our project.

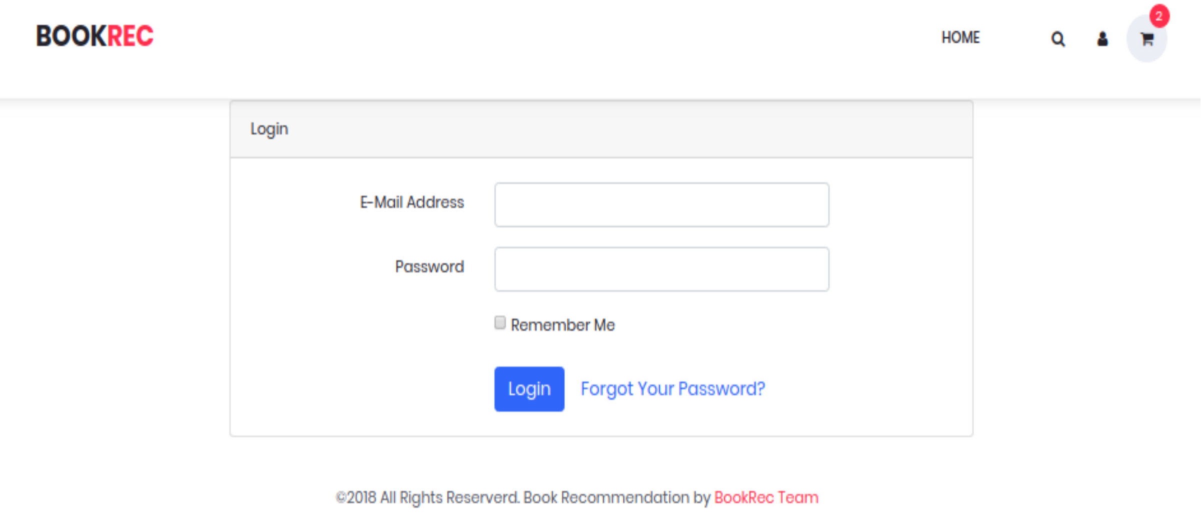
6.2 Recommendation

For future work, the recommendation system can be upgraded to a more advanced version by adding content based filtering or using algorithms that include neural network. In this project, since collaborative filtering is used the recommendation depends on user data. If user data is no available then the recommendation system is not able to give any recommendation. Since recommendation system is memory based it is mandatory to retrain model time to time on regular basis for better recommendation for new user and newer books. Also, the effectiveness of the recommendation depends on the diversity of the entire user’s data. Moreover, the recommendation system does not recommend that have not been rated by any user. On top of that, for new users that has not rated any books. The recommendation model also faces cold start problem at the beginning of deployment as no ratings are available at first.

User interface in client browser corresponding to the API

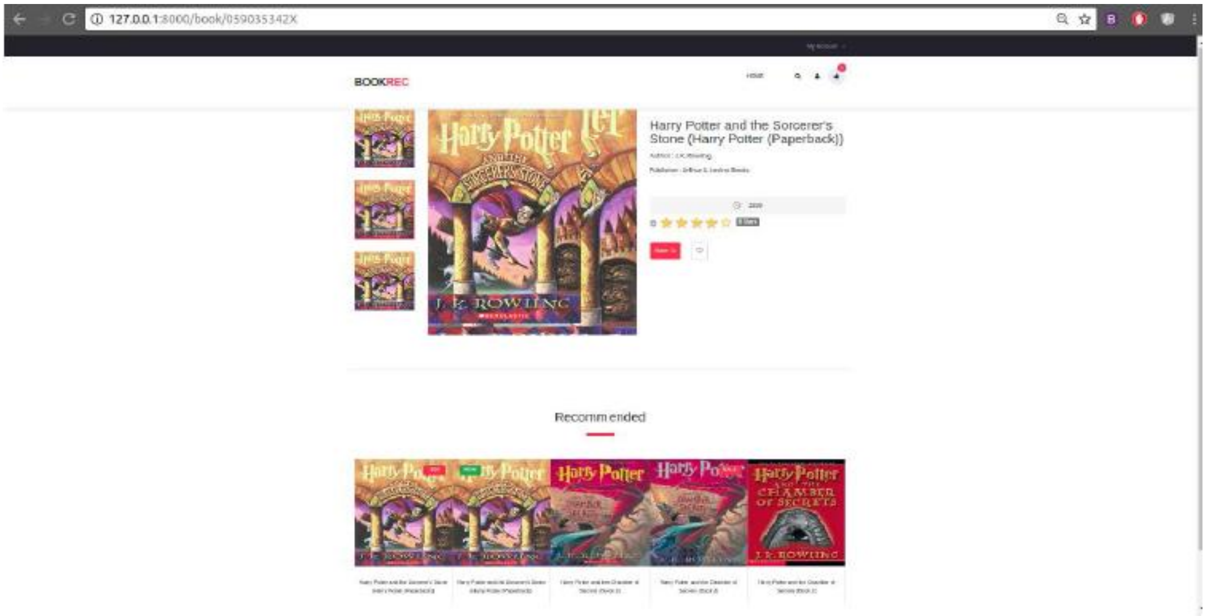

127.0.0.1:8000
BOOKREC
HOME
0
Recommended
The Lovely Bones: A Novel
The Da Vinci Code
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
To Kill a Mockingbird
The Red Tent (Bestselling Backlist)

Login Page



©2018 All Rights Reserverd. Book Recommendation by BookRec Team

Figure: Recommended list of books



127.0.0.1:8000/book/059035342X

BOOKREC

Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))

Recommended

REFERENCES

1. C. Pan, W. Li. “Research paper recommendation with topic analysis. In Computer Design and Applications.” IEEE. 4 (2010), pp. V4-264

2. C. Zaiontz. "Real Statistics Using Excel", Real Statistics Using Excel, 2017. .

3. E. Gabrielova, C. Lopes, et al. "The Yelp dataset challenge - Multilabel Classification of Yelp reviews into relevant categories", Ics.uci.edu, 2017. [Online]. Available: http://www.ics.uci.edu/~vpsaini/.

4. E. R. Hedrick, LoCajori, Florian (1952) [1929]. A History of Mathematical Notations. 2 (2 (3rd corrected printing of 1929 issue) ed.). Chicago, USA: Open court publishing company. [Accessed 14 June 2017]

5. F. Ricci, L. Rokach, B. Shapira. (2011). “Introduction to Recommender Systems Handbook. Springer.” [Online]. pp. 1-35.

6. H. Jafarkarimi, A.T.H. Sim and R. Saadatdoost. (2012, June). “A Naïve Recommendation Model for Large Databases.” International Journal of Information and Education Technology.

7. J. A. Konstan, J. Riedl. “Recommender systems: from algorithms to user experience User Model User-Adapt Interact.” 22 (2012), pp. 101–123

8. R. J. Mooney and L. Roy. “Content-Based book recommendation using learning for text categorization”. In Proc. Fifth ACM conference on digital libraries, 2010, pp. 195-204