PROJECT AND PRESENTATION [CSE 498]

ON

WEB-BASED BOOK SEARCHING AND BOOK RECOMMENDATION SYSTEM USING MACHINE LEARNING

BY

MD. TANVIR AHMED (181472598)

A Project Report submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

MD. ATAULLAH BHUIYAN

Senior Lecturer

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)

CITY UNIVERSITY

CITY UNIVERSITY

DHAKA, BANGLADESH

MARCH 2022

DECLARATION

I hereby declare that, this project has been done by me under the supervision of **Md. Ataullah Bhuiyan,** Senior Lecturer, Department of CSE City University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Submitted by:

Tanvir

Md. Tanvir Ahmed
ID: 181472598
Batch: 47
Department of CSE
City University, City Campus,
Dhaka, Bangladesh

Submitted To:

Ataullah

Md. Ataullah Bhuiyan,
Senior Lecturer, Department of CSE,
City University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

I fell grateful to and wish our profound our indebtedness to **Md. Ataullah Bhuiyan**, Senior Lecturer, Department of CSE City University, Dhaka. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

My special thanks go to the Head of the Department of CSE, **Md. Safaet Hossain**, who had given me the permission and encouraged us to go ahead. I am bound to the Honorable Dean of Department of Science Faculty, **Prof. Dr. Engr. Md. Huamaun Kabir**, for his endless support.

I am very grateful to all my faculty teachers who gave me their valuable guides to complete my graduation. I am also very grateful to all those people who have helped me to complete my project.

I would like to thank my entire course mate in City University, who took part in this discuss while completing the course work.
Finally, I must acknowledge with due respect the constant support and patients of my parents.

Tanvir

Md. Tanvir Ahmed
ID: 181472598
Department of CSE
City University

ABSTRACT

Many people rely on search engines to for education and getting knowledge and information from books. It's a continuous learning way. These online books helpful to solve many queries in every field whether it's related to medicine, engineering, commerce, science, political issues or arts. People can spend their spare time by online reading of their interesting subjects. Our PDF search option provides the ability to get variety of information for preparing lectures, assignments, preparation of tests. It provides best filter, refined, organize and precise information from eBooks and online material that assists in increasing knowledge and meets requirement. The developed system is a "Recommendation System Using Content based Filtering" which deals with developing an e-book website. The user can browse by catalog, search and get a recommendation of books in the website. The users are recommended based on book description using google word2vec model. To develop an e-book website, several Technologies was studied and understood. These include server and client-side scripting techniques, implementation technologies such as Python, Flask, HTML, CSS, Bootstrap, JavaScript, React JS, PHP and MySQL for database handling.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Relevance of the Project

The demand for e-books has exploded due to the rapid advancement of internet technology. During Covid-19, the demand for e-books hit an all-time high. This online e-book service uses a recommendation algorithm to help locate books online.

Nowadays, physical visits to shops and libraries have been drastically reduced due to their busy schedules and COVID-19 pandemic. Instead, e-marketplaces and e-libraries became popular hotspots. E-book reading platforms and online purchasing tendencies made users discover their favorite books from many items. As a result, users tend to get swift and smart decisions from an unprecedented number of choices using expert systems. Thus, recommendation systems came into the scene to customize users' searching and deliver the best-optimized results from a multiplicity of options.

The recommendation systems' algorithms were usually developed based on content-based filtering, associative rules, multi-model ensemble, and collaborative filtering. Multi-model ensemble algorithms can be used for personalized recommendation systems.

Recommendation systems are commonly used to propose the best relevant items to end consumers. This system generates efficient and effective suggestions by utilizing content-based filtering features. One of the most well-known, widely deployed, and mature technologies is content-based recommendation. This recommender system employs the word2vec model, which employs a word distance vector.

1.2 Problem Definition

If someone wants to buy a new product, they usually question their friends, do some research on the product's attributes, compare it to similar products, and read product reviews on the internet before making a decision. How convenient would it be if all of this was handled automatically and the product was recommended effectively? The answer to this question is a recommendation engine or recommender system.

Due to the effect of Covid-19, collecting Physical Lecture Sheets, books, and other materials has become difficult. As a result, we must rely on PDF or e-pub files. Finding appropriate books from a huge e-book space has become a major difficulty for internet users as the volume of online books grows tremendously as a result of the COVID-19 epidemic. The typical library system requires the user to visit a library to look for and read books, which is a time-consuming and laborious operation. Due to the library's insufficient distribution of books, it's difficult to find certain books on such a tight timeline. The user was unable to receive a recommendation for the best book to read.

1.3 Objective

The general objective of the project is to develop a web application to search and read books stored online.

The Specific Objectives are listed below:

- To provide platform to view the books online.
- To read books online.
- To search books online.
- To recommend a book to a user.

1.4 Scope

Our project has lots of scope for work in future. The project scope involves study of the Machine Learning. Below given some discussion about scope of this project.

- In future users' behavior-based collaborative filtering can be applied to suggest most appropriate books.

- The word2vec algorithm extracts features from the text for particular words. Using its ability with linear algebra, the algorithm can recognize context and words that have similar meanings. Word2vec algorithm can also be used for recommending other products like: Movies, Grocery items, Cloths etc.

1.5 Limitation

- Stable internet connection is required.
- Larger servers are required to execute the applications smoothly and efficiently, which we are unable to accomplish on our local server.
- Only similar books are recommended as this system does not facilitates hybrid recommendation techniques.

CHAPTER 2: REVIEW OF BACKGROUND AND RELATED STUDIES

2.1 Theoretical Background

Book recommendation system using opinion mining technique: This is a recommendation technique based on opinion mining to propose top ranked books on different discipline of the computer science. They have categorized and reviews of the users. Here the Weights are assigned to categorized features according to their importance and usage, and accordingly the ranks are given [1].

Book Recommendation System through content based and collaborative filtering method: This paper presents Book Recommendation System (BRS) based on combined features of content-based filtering (CBF), collaborative filtering (CF) and association rule mining to produce efficient and effective recommendation. This paper proposed a hybrid algorithm in which they combine two or more algorithms, thus it helps the recommendation system to recommend the book based on the user interest [2].

Web-based personalized hybrid book recommendation system: A web-based personalized hybrid book recommender system which exploits varied aspects of giving recommendations apart from the regular collaborative and content-based filtering approaches. The users of different age, gender and country, personalized recommendations can be made on these demographic parameters. Scraping information from the web and using the information obtained from this process are used in making recommendations [3].

Machine Learning based Efficient Recommendation System for Book Selection using User based Collaborative Filtering Algorithm: This paper proposes a simple understandable system for book recommendations that help readers to suggest the right book, which is to be studied next.

The proposed method works on training, feedback, management, reporting, configuration, and using it to offer useful information to the user in order to aid in decision-making and data item recommendations. They have used a User Based Collaborative Filtering (UBCF) approach and measured the performance of similarity measures in recommending books to a user [4].

Embedding Model Design for Producing Book Recommendation: Here the recommendation system has been tested on one random-picked user behavior and successfully has generated five recommended books by analyzing prior activities. The embedding model produced the recommended books with 59% of accuracy [5].

2.2 Comparison with existing work

Existing work	**Method**	**Feature**	**Limitations**	**Ref**
Existing work 1	Opinion mining	• Can rank the recommended book • Opinion mining technique is used to improve the accuracy of the recommendation system	• Limited to computer science-related books. • Only recommend 10 books for a particular query	1
Existing work 2	Content-Based and Collaborative Filtering	•Combined features from two widely used filtering techniques - content-based filter and collaborative filtering	The authors use some unnecessary attributes like name of a registered user, password of the registered user, and email	2

Existing work 3	•Item-Item Similarity Technique •Web Scraping Process	•Scraping information is useful for making recommendations. •Consider the temporal aspects while recommending books • Overcome the problems of the content-based and collaborative filtering	The authors failed to explain the impact of clustering in the recommendation system Web-based recommendation system needs to be secure	3
Existing work 4	User-Based collaborative filtering	Use a user-based similarity matrix to increase the accuracy of the collaborative filtering algorithm	Cluster can improve the accuracy and performance of the recommendation system	4
Existing work 5	users' behavior-based collaborative filtering	Users' behavior based collaborative filtering recommends a series of books	Low accuracy of the classifier, which is 59%	5

CHAPTER 3: SYSTEM SPECIFICATION REQUIREMENTS

3.1 Hardware Requirements

- A PC with Windows/Linux OS
- Processor with 1.7-2.4gHz speed
- Minimum of 8gb RAM
- 2gb Graphic card

3.2 Software Specification

- Text Editor (VS-code/WebStorm/ PyCharm Editor)
- Google Colab
- Python
- React JS
- Laravel
- Database (MYSQL)

3.3 React JS

The primary goal of ReactJS is to create User Interfaces (UI) that increase app speed. It makes use of virtual DOM (JavaScript object), which increases the app's performance. The virtual DOM in JavaScript is faster than the conventional DOM. ReactJS can be used on the client and server side, as well as in conjunction with other frameworks. It employs component and data patterns to improve readability while also assisting in the maintenance of larger apps.[7]

3.4 Laravel

In this project Lumen micro-framework is used. Lumen is a micro-framework built on top of Laravel’s core components. Lumen utilizes a lot of familiar concepts from Laravel such as Eloquent, caching, routing, middleware and it’s service container. One major advantage of Lumen is it’s speed, as a result the framework is designed for building fast micro-services or APIs.

3.5 Python

For the computation and analysis, we need certain python libraries which are used to perform analytics. Packages such as SKlearn, NumPy, pandas, Matplotlib, Flask framework etc. are needed.

SKlearn: It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Pandas: Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

StopWords: A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

Flask: It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug.

Cosine_Similarity: In data analysis, cosine similarity is a measure of similarity between two sequences of numbers. For defining it, the sequences are viewed as vectors in an inner product space, and the cosine similarity is defined as the cosine of the angle between them, that is, the dot product of the vectors divided by the product of their lengths. It follows that the cosine similarity does not depend on the magnitudes of the vectors, but only on their angle. The cosine similarity always belongs to the interval [-1,1]. For example, two proportional vectors have a cosine similarity of 1, two orthogonal vectors have a similarity of 0, and two opposite vectors have a similarity of -1. The cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

3.6 Google Colab

Project Jupyter is a project and community whose goal is to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages. It was spun off from IPython in 2014 by Fernando Pérez and Brian Granger. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R, and also a homage to Galileo's notebooks recording the discovery of the moons of Jupiter. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab. Jupyter is financially sponsored by NumFOCUS.

3.7 Functional Requirements

Allowing people to see the site without logging in is one of the functional requirements of the book recommendation system. Books are recommended to the user based on mathematical analysis of their behavior. Book recommendation system allow users to register and rate books, which creates a relevant knowledge base from which recommendations can be made.

3.7.1 Non- Functional Requirements

Non-functional requirements of the project are as follows:

Performance: Being a web-based system, the performance depends on the server itself, how the server responses to the request determine the performance of the system. The system inefficient in resource utilization like memory, CPU, storage etc.

Scalability: The system provides different features for both admin and users. The system is able to serve large number of users as per demand.

Reliability: The system provides reliable recommendation to the users based on NLP, Word Embedding System.

Usability: The system can be used by any user by registering to the site.

Interoperability: The system is built by integrating python Flask Framework, React JS Framework and PHP Laravel Framework with HTML, CSS. So, they operate together.

3.7.2 Economic Feasibility

Economic analysis is the most often utilized method for assessing a system's efficacy. The tangible advantages advocated that physical labor and load be minimized to the greatest extent practicable, resulting in a decrease in personnel requirements and costs associated with manpower. The system delivers several non-monetary benefits, such as user friendliness, more efficient user response, database maintenance, and so on.[8]

CHAPTER 4: SYSTEM ANALYSIS AND DESIGN

4.1 Planning

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

4.2 Designing

In this Phase after successfully analysis the requirements and planning the next and important part is design. Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term "web design" is normally used to describe the design process relating to the front-end (client side) design of a website including writing markup.

4.3 SDLC Models

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third-party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in Design Document Specification. The application of standard business practices to the development of software applications is known as the Software Development Life Cycle. Planning, Requirements, Design, Build, Document, Test, Deploy, and Maintain are the six to eight steps that are usually followed. Depending on the scope of the project, some project managers will combine, split, or eliminate steps. All software development efforts should include these fundamental components.

The SDLC is a method for evaluating and improving the development process. It enables for a fine-grained study of each process phase. As a result, businesses are able to maximize efficiency at each level. As processing capacity grows, the demand for software and developers grows as well. Companies must cut expenses, deploy software more quickly, and meet or exceed the expectations of their consumers. SDLC helps achieve these goals by identifying inefficiencies and higher costs and fixing them to run smoothly.[9]

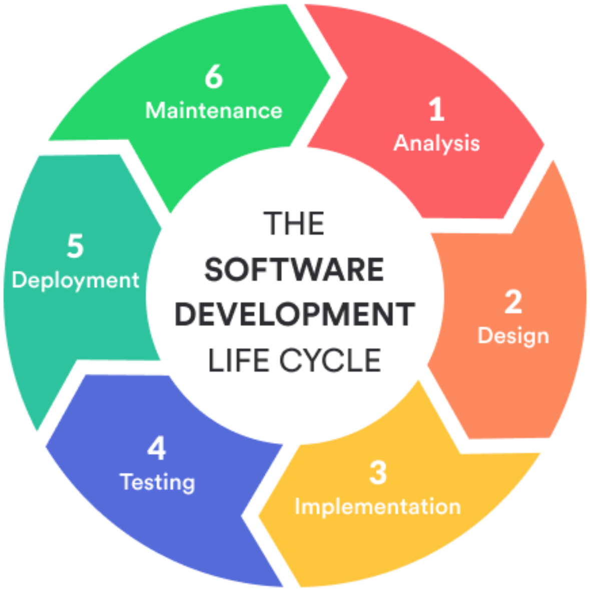


Fig 4.1: SDLC model

4.4 Principle of System analysis design

- Understand the problem before begin to create the analysis model.
- Develop prototypes that enable a user to understand how human machine interaction will occur.
- Record the origin of and the reason for every requirement.
- Use multiple views of requirements like building data, function and behavioral models.
- Work to eliminate ambiguity.

4.5 System architecture:

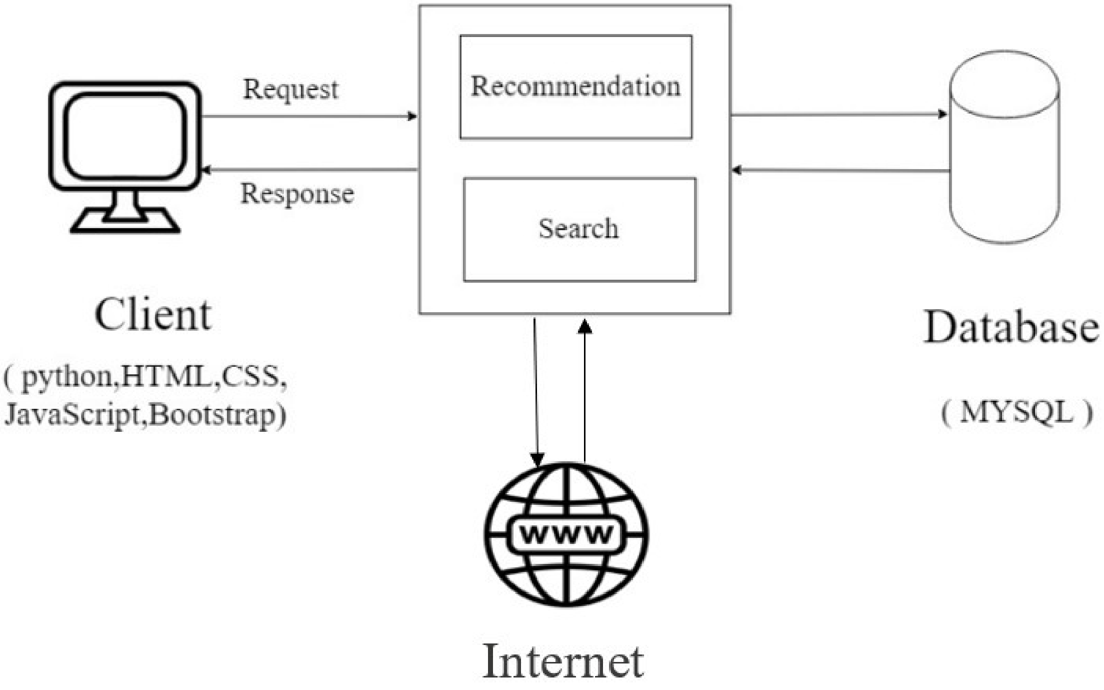


Fig 4.2: System Architecture

4.6 Use Case Diagram

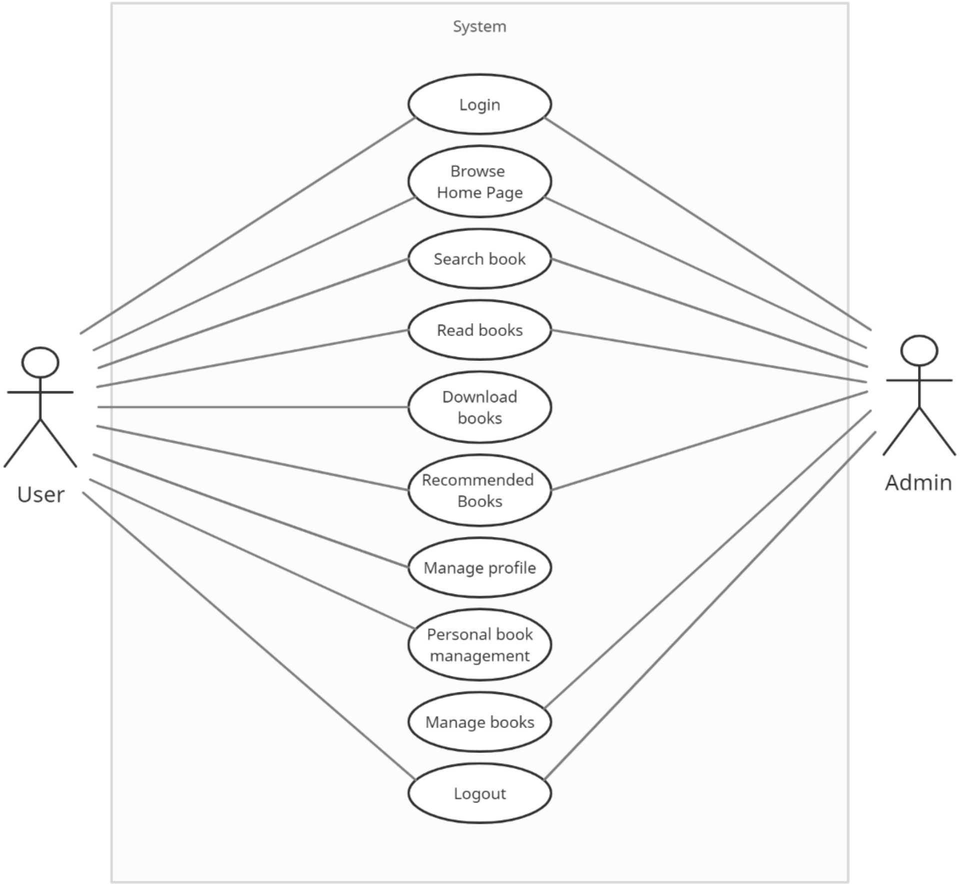


Fig 4.3: Use Case Diagram

The use case diagram for this website is shown in Figure 4.3. The two actors in the Online Book Store are User and Admin, with Admin being the super user. Login, explore the home page, search for a book, view recommended books, view profile, download books, view books, modify profile, and logout are all options available to the user. Admin can also take on the role of a user and perform user chores as well as manage users and books.

4.7 E-R diagram

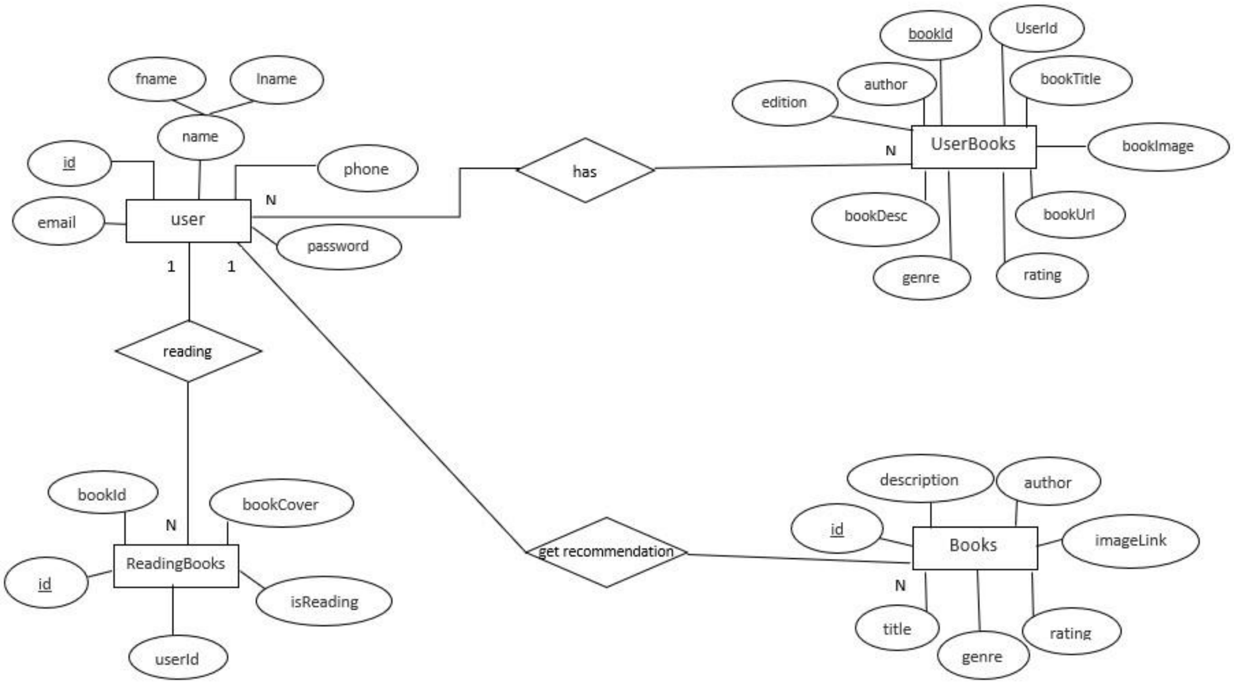


Fig 4.4: E-R Diagram

Figure 4.4 shows the logical relationship between numbers of entities. User, Order, User books, Reading, Books, Books are the entities and each entity has their respective attributes.

4.8 Level-0 DFD diagram

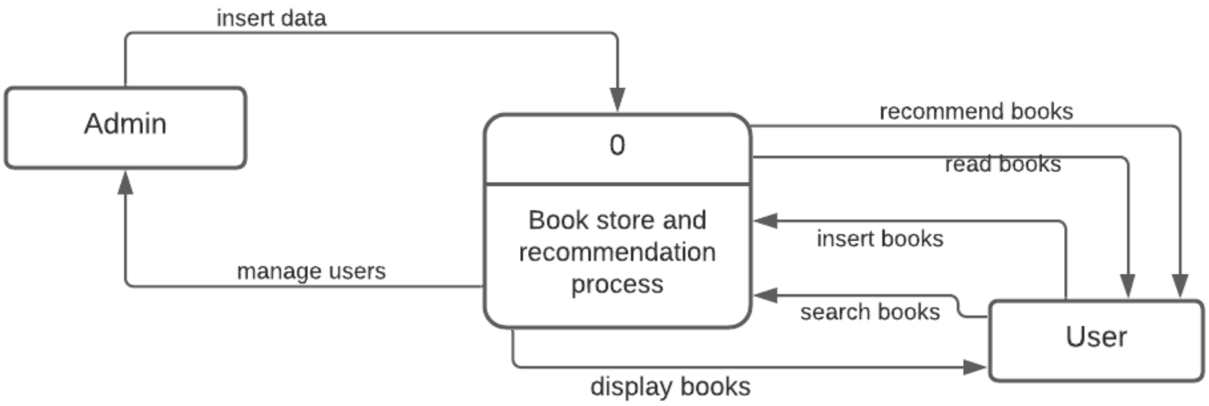


Fig 4.5: Level-0 DFD Diagram

The visualization of data processing and structured design of the Book Store process and working flow was done using the Figure 4.5 diagram. Level-0 DFD is frequently used as a first stage in creating an overview of the Book Store without going into great depth, which may then be expanded upon afterwards. It usually comprises of the overall application dataflow and Book Store process operations. It contains the whole user flow as well as their entities.

4.8.1 Level-1 DFD Diagram

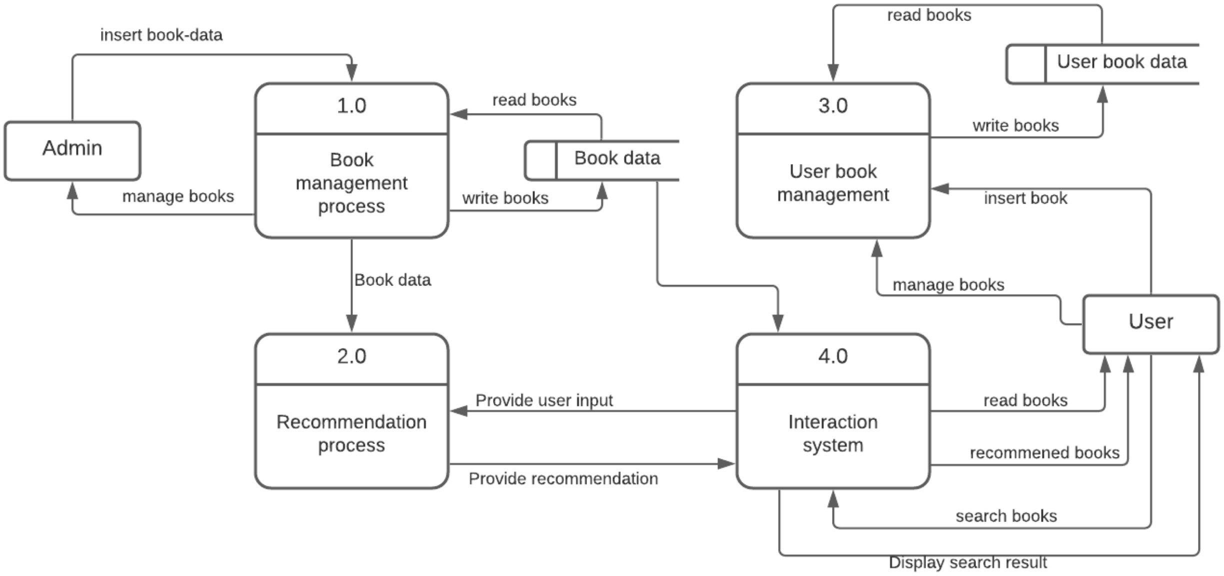


Fig 4.6: Level-1 DFD Diagram

Each of the processes in Figure 4.6 has its own input and output. We break down the process 0 Book Recommendation process into a variety of processes in Level-1 DFD.

4.9 Class Diagram

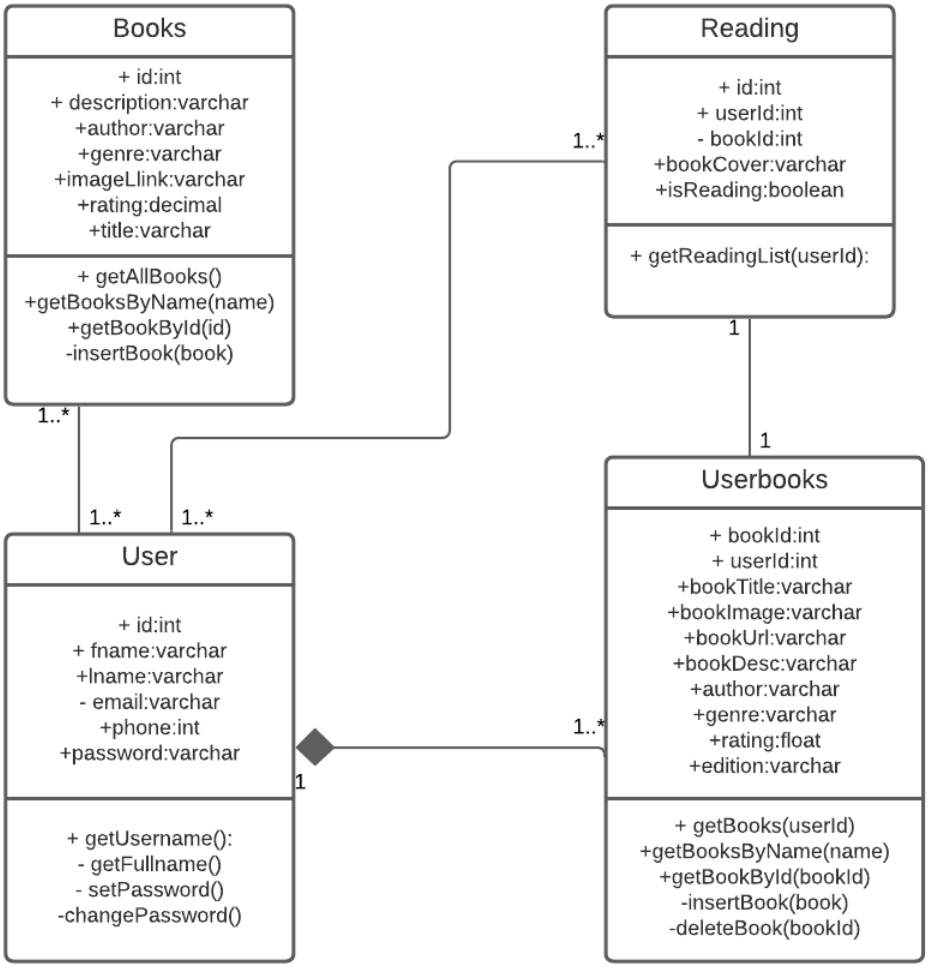


Fig 4.7: Class Diagram

Figure 4.7 is a static structural diagram that depicts the structure of our system by displaying the classes, properties, operations. and relationships between the objects. Classes are represented in the diagram above by boxes with three sections. The name of the class is centered and the first letter is uppercase in the top compartment. The attributes of the classes are kept in the center compartment. The operations that the class can do are kept in the bottom container. Relationships like association, dependency, aggregation, composition are represented using logical connectors between the boxes. Relation between Category and Product that a single category can contain multiple products.

4.10 Activity Diagram

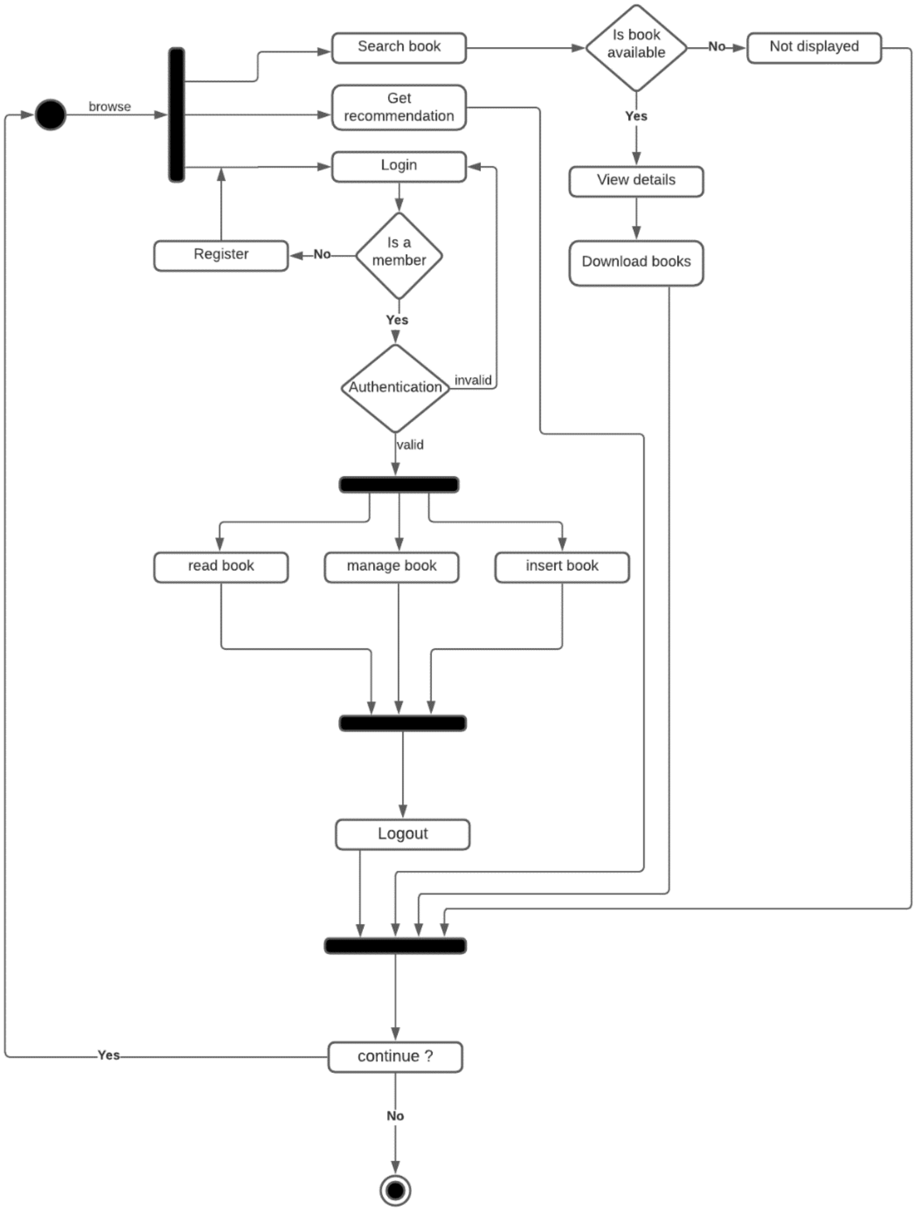


Fig 4.8: Activity Diagram

The user can browse the website and search for books or obtain book recommendations, as shown in Figure 4.8. If the book is accessible after a search, the user can read the book only if he or she is signed in. If the user does not have a unique account, he or she must register with the system. If the user is interested with the book after viewing it, he or she can add it to their reading list. After that, the user has the option of continuing to browse or logging out.

4.11 Schema Diagram

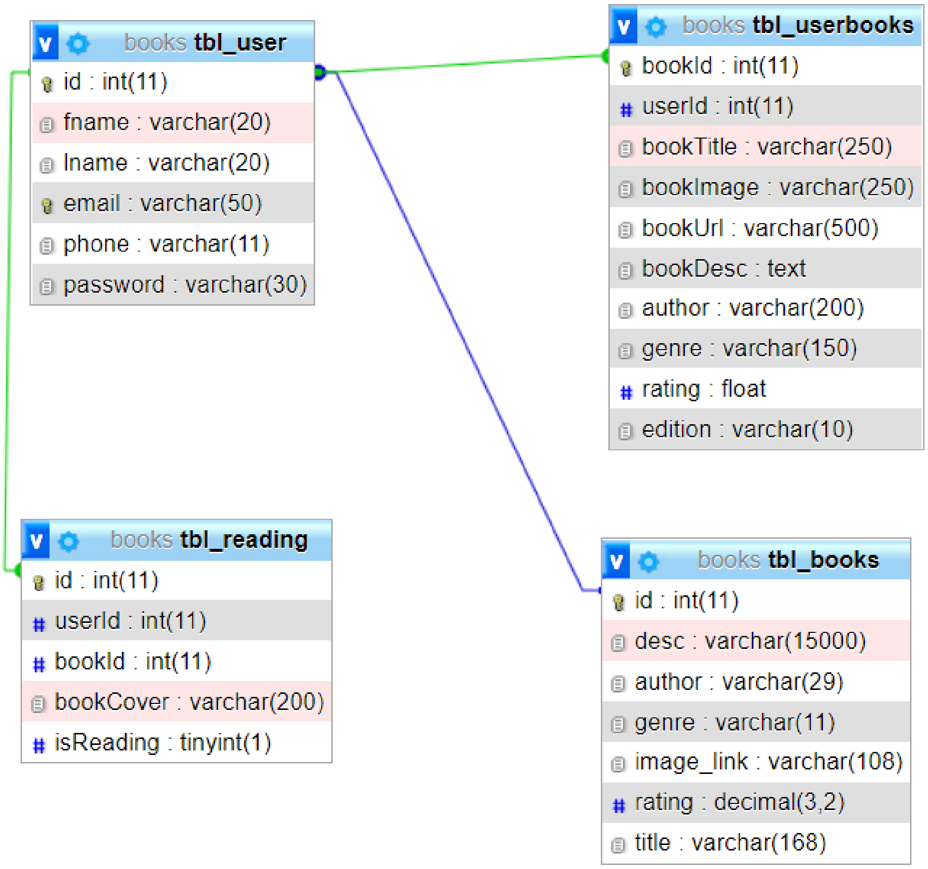


Fig 4.9: Schema Diagram DFD

The visual depiction of our database's organization and structure is shown in Figure 4.9. Tables, columns, data types, relationships, primary keys, foreign keys, and other schema components are included. It displays seven tables, as well as their data types, connections between tables, and main and foreign keys.

4.12 Gantt chart

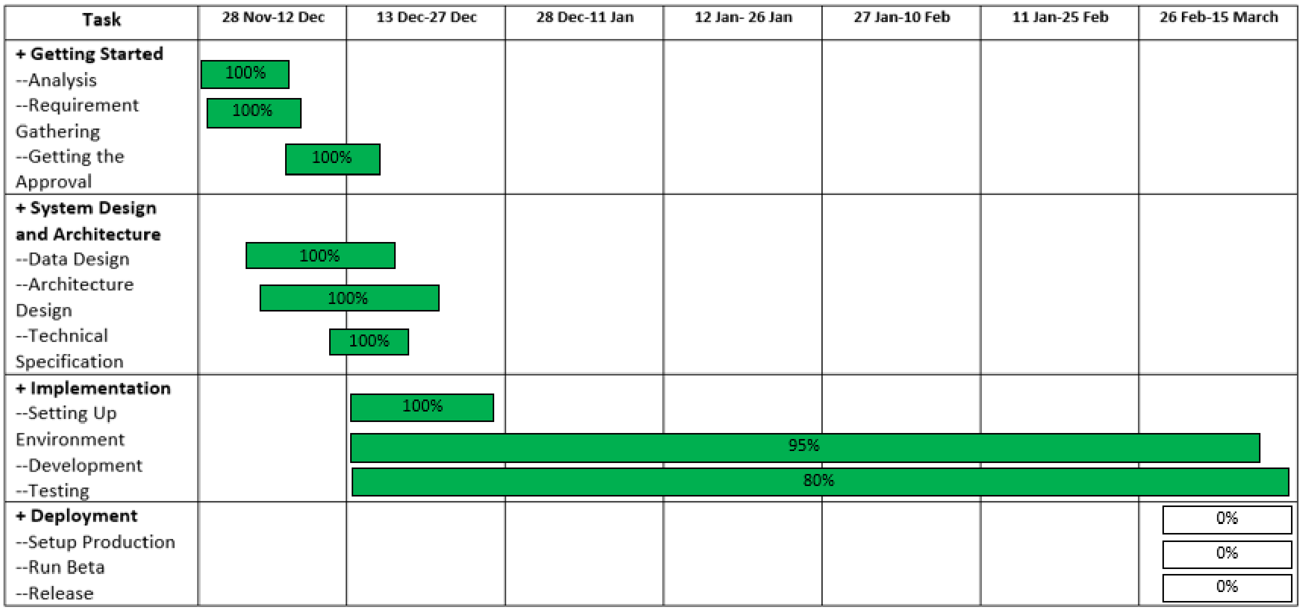


Fig 4.10: Gantt Chart

The chart identifies tasks that may be executed in parallel and those that can't be started or finished until others are complete. It can help detect potential bottlenecks and identify tasks that may have been excluded from the project timeline.

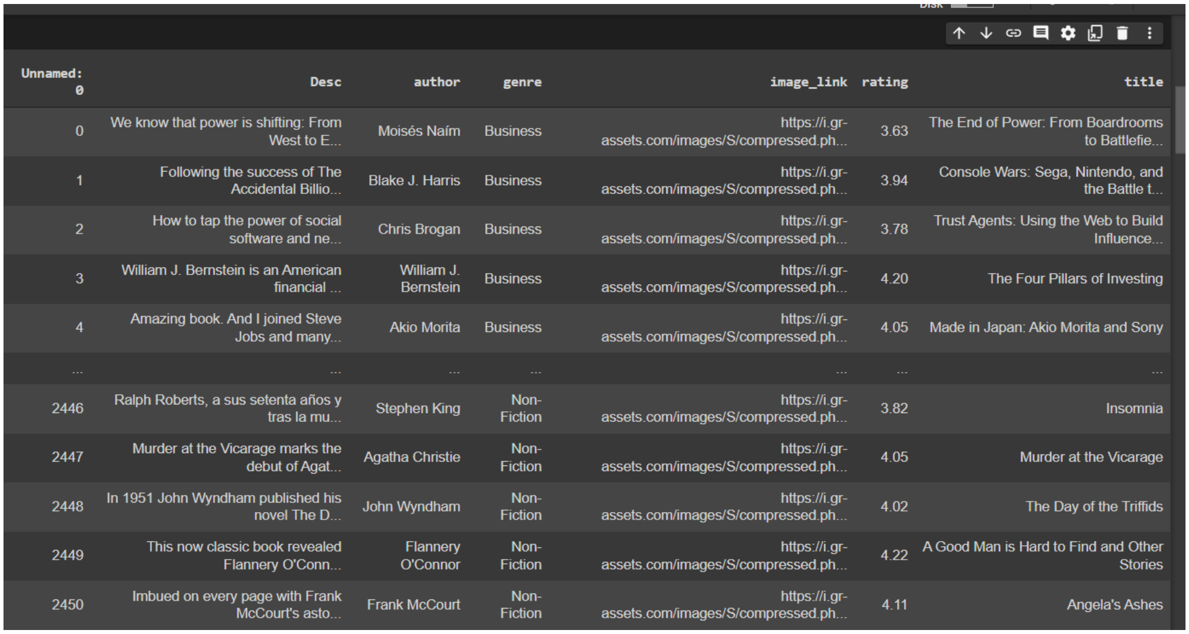
The chart depicts things like: task slack time or additional time for completion of a task that shouldn't delay the project; noncritical activities that may be delayed; and critical activities that must be executed on time.

CHAPTER 5: METHODOLOGY

The system has been implemented using the Python, PHP, JavaScript Programming Language with Flask, Laravel, React JS Framework respectively. Here, we implemented the Wrord2Vec based on content-based Filtering technique to provide recommendations to the user. In the Admin and user side, for each features, there are classes, models, functions in web pages involving the requests from users or display the result on the web pages.

5.1 ML Dataset

Dataset collection: Kaggle (Goodreads) and scrapped from Goodreads.

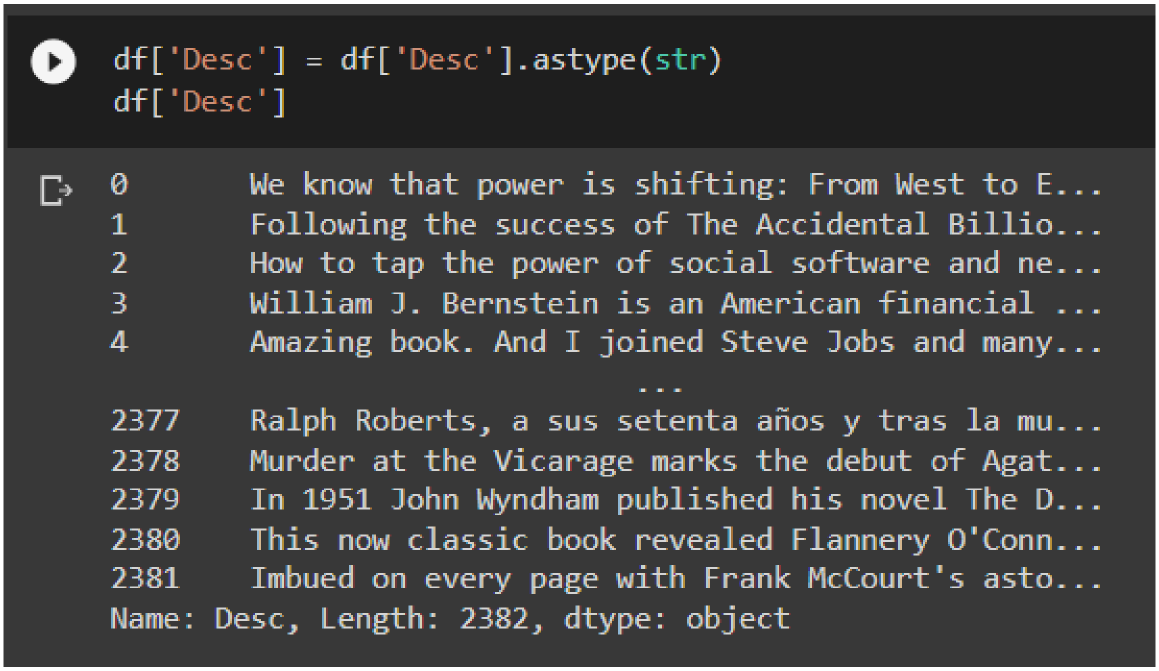


Unnamed: 0	Desc	author	genre	image_link	rating	title
0	We know that power is shifting: From West to E...	Moisés Naím	Business	https://i.gr-assets.com/images/S/compressed.ph...	3.63	The End of Power: From Boardrooms to Battlefie...
1	Following the success of The Accidental Billio...	Blake J. Harris	Business	https://i.gr-assets.com/images/S/compressed.ph...	3.94	Console Wars: Sega, Nintendo, and the Battle t...
2	How to tap the power of social software and ne...	Chris Brogan	Business	https://i.gr-assets.com/images/S/compressed.ph...	3.78	Trust Agents: Using the Web to Build Influence...
3	William J. Bernstein is an American financial ...	William J. Bernstein	Business	https://i.gr-assets.com/images/S/compressed.ph...	4.20	The Four Pillars of Investing
4	Amazing book. And I joined Steve Jobs and many...	Akio Morita	Business	https://i.gr-assets.com/images/S/compressed.ph...	4.05	Made in Japan: Akio Morita and Sony
...	...	...	...	...	...	...
2446	Ralph Roberts, a sus setenta años y tras la mu...	Stephen King	Non-Fiction	https://i.gr-assets.com/images/S/compressed.ph...	3.82	Insomnia
2447	Murder at the Vicarage marks the debut of Agat...	Agatha Christie	Non-Fiction	https://i.gr-assets.com/images/S/compressed.ph...	4.05	Murder at the Vicarage
2448	In 1951 John Wyndham published his novel The D...	John Wyndham	Non-Fiction	https://i.gr-assets.com/images/S/compressed.ph...	4.02	The Day of the Triffids
2449	This now classic book revealed Flannery O'Conn...	Flannery O'Connor	Non-Fiction	https://i.gr-assets.com/images/S/compressed.ph...	4.22	A Good Man is Hard to Find and Other Stories
2450	Imbued on every page with Frank McCourt's asto...	Frank McCourt	Non-Fiction	https://i.gr-assets.com/images/S/compressed.ph...	4.11	Angela's Ashes

Fig 5.1: Data Set

5.2 Data Cleaning and Preprocessing

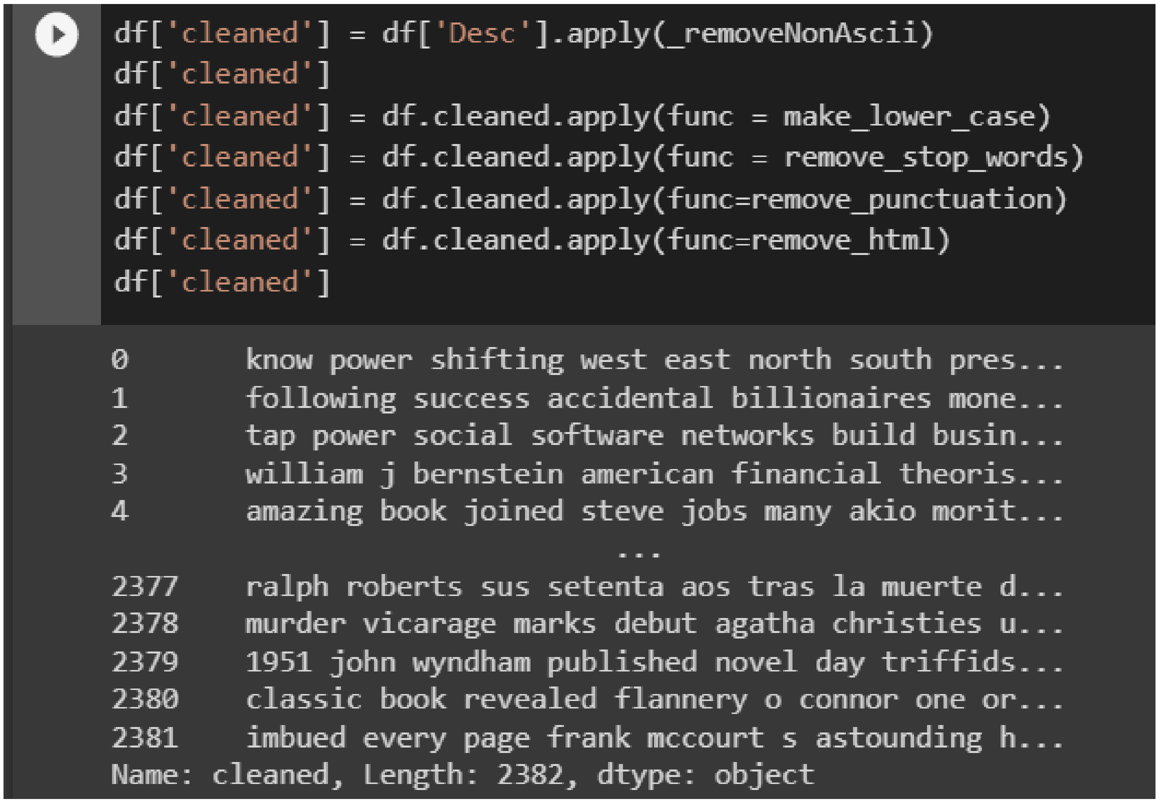
Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. At the very fast we remove the non-ASCII character. After that we make the text lower case and then remove punctuation mark and html tag and also remove the stopwords.



```
df['Desc'] = df['Desc'].astype(str)
df['Desc']
```

```
0       We know that power is shifting: From West to E...
1       Following the success of The Accidental Billio...
2       How to tap the power of social software and ne...
3       William J. Bernstein is an American financial ...
4       Amazing book. And I joined Steve Jobs and many...
                              ...
2377    Ralph Roberts, a sus setenta años y tras la mu...
2378    Murder at the Vicarage marks the debut of Agat...
2379    In 1951 John Wyndham published his novel The D...
2380    This now classic book revealed Flannery O'Conn...
2381    Imbued on every page with Frank McCourt's asto...
Name: Desc, Length: 2382, dtype: object
```

Fig 5.2: Working with Description



```
df['cleaned'] = df['Desc'].apply(_removeNonAscii)
df['cleaned']
df['cleaned'] = df.cleaned.apply(func = make_lower_case)
df['cleaned'] = df.cleaned.apply(func = remove_stop_words)
df['cleaned'] = df.cleaned.apply(func=remove_punctuation)
df['cleaned'] = df.cleaned.apply(func=remove_html)
df['cleaned']
```

```
0       know power shifting west east north south pres...
1       following success accidental billionaires mone...
2       tap power social software networks build busin...
3       william j bernstein american financial theoris...
4       amazing book joined steve jobs many akio morit...
                              ...
2377    ralph roberts sus setenta aos tras la muerte d...
2378    murder vicarage marks debut agatha christies u...
2379    1951 john wyndham published novel day triffids...
2380    classic book revealed flannery o connor one or...
2381    imbued every page frank mccourt s astounding h...
Name: cleaned, Length: 2382, dtype: object
```

Fig 5.3: Data Cleaning

5.3 Algorithm Used For Recommendation

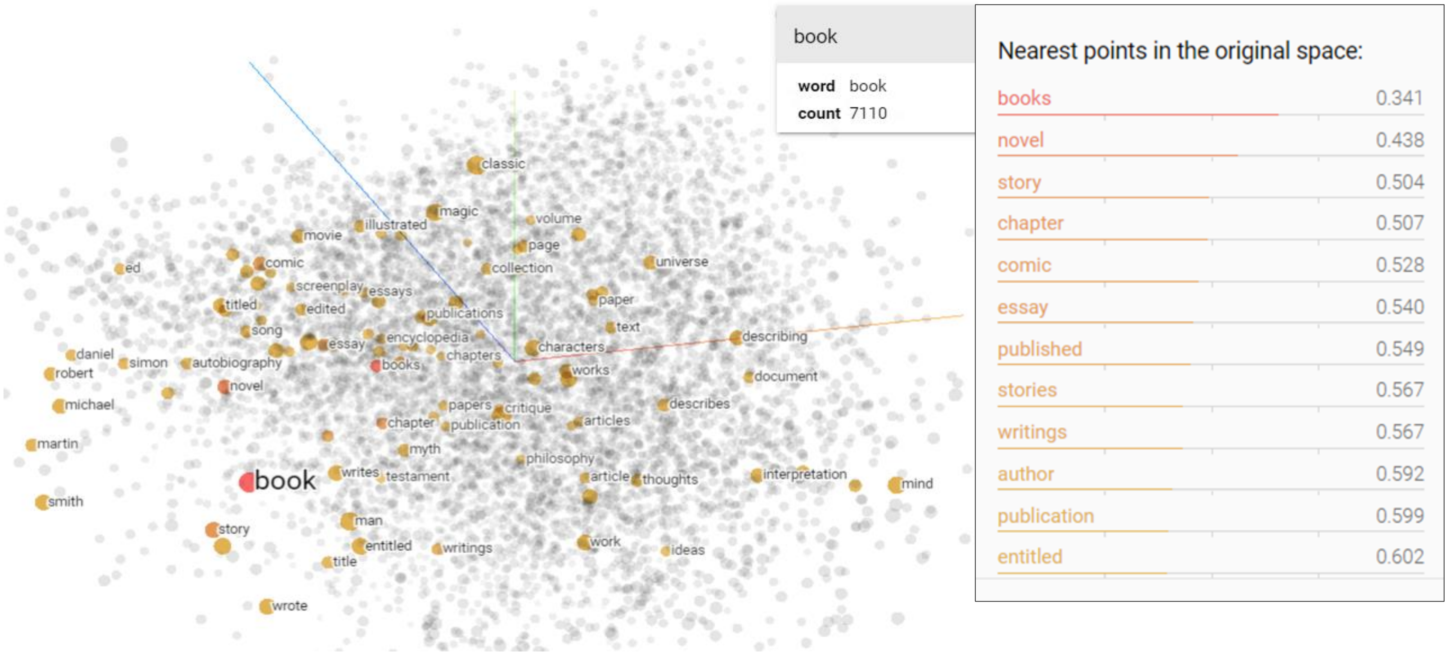
Word2Vec is a classical method that creates word embeddings in the field of Natural Language Processing **(NLP).** It was developed by Tomas Mikolov and his team at Google in 2013. Word2vec takes in words from a large corpus of texts as input and learns to give out their vector representation.

In the same way CNNs extract features from images, the word2vec algorithm extracts features from the text for particular words. Using those features, word2vec creates vectors that represent a word in the vector space. These vectors are chosen using the cosine similarity function, which indicates the semantic similarity between words.

Cosine similarity of 1 would mean that the angle between two words is 0; and would denote that the words are similar. Two similar words will occupy locations close to each other in that vector space, whereas words that are very different will occupy far away spaces. In that way, using its ability with linear algebra, the algorithm can recognize context and words that have similar meanings.

For example, the words “intelligent” and “smart” would appear closer together in this vector space, whereas the words “engine” and “car” will be far from “intelligent” and “smart”. This is because these words have that contextual understanding within a vector space.

5.4 How word2vec works?



book

word book
count 7110

Nearest points in the original space:

books	0.341
novel	0.438
story	0.504
chapter	0.507
comic	0.528
essay	0.540
published	0.549
stories	0.567
writings	0.567
author	0.592
publication	0.599
entitled	0.602

Fig 5.4: Word2vec 3d Model

Word2Vec creates vectors of the words that are distributed numerical representations of word features – these word features could comprise of words that represent the context of the individual words present in our vocabulary. Word embeddings eventually help in establishing the association of a word with another similar meaning word through the created vectors. Words are represented in the form of vectors and placement is done in such a way that similar meaning words appear together and dissimilar words are located far away.

Word2vec learns word by predicting its surrounding context. For example, let us take the word “He read books”

We want to calculate the Word2vec for the word: read.

Suppose

read = V_{in}. $P(V_{out} / V_{in})$ is calculated

where,

V_{in} is the input word.

P is the probability of likelihood.

V_{out} is the output word.

Word read moves over each word in the corpus. Syntactic as well as the Semantic relationship between words is encoded. This helps in finding similar and analogies words.

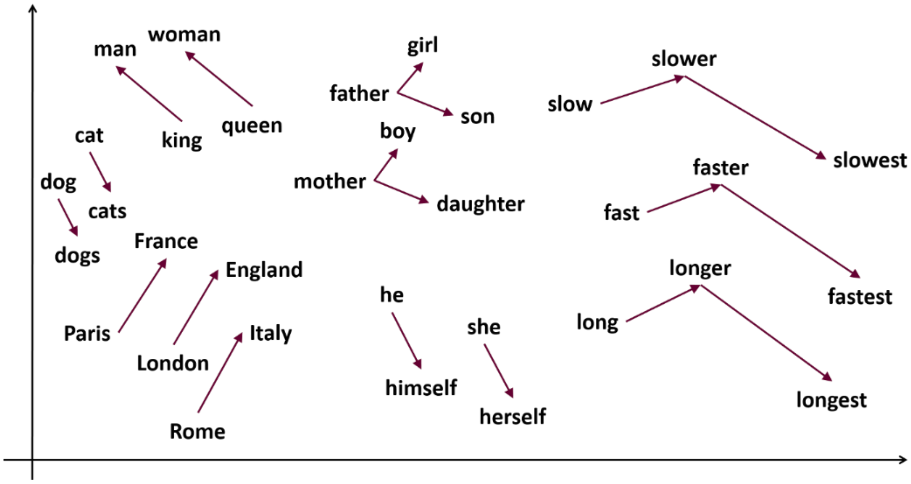


Fig 5.5: Word2vec model vector relation

5.5 Recommendation Flowchart

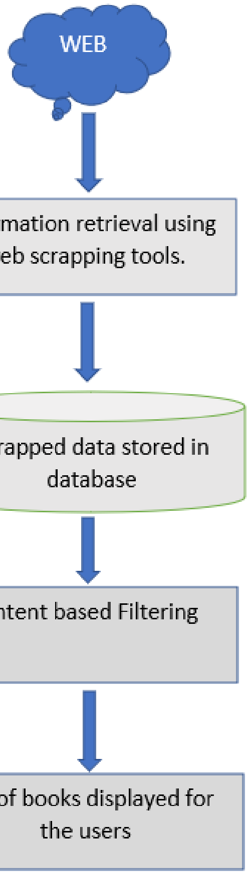

WEB
Information retrieval using web scrapping tools.
Scrapped data stored in database
Content based Filtering
List of books displayed for the users

Fig 5.6: Recommendation Flowchart

5.6 Recommendation Model Making

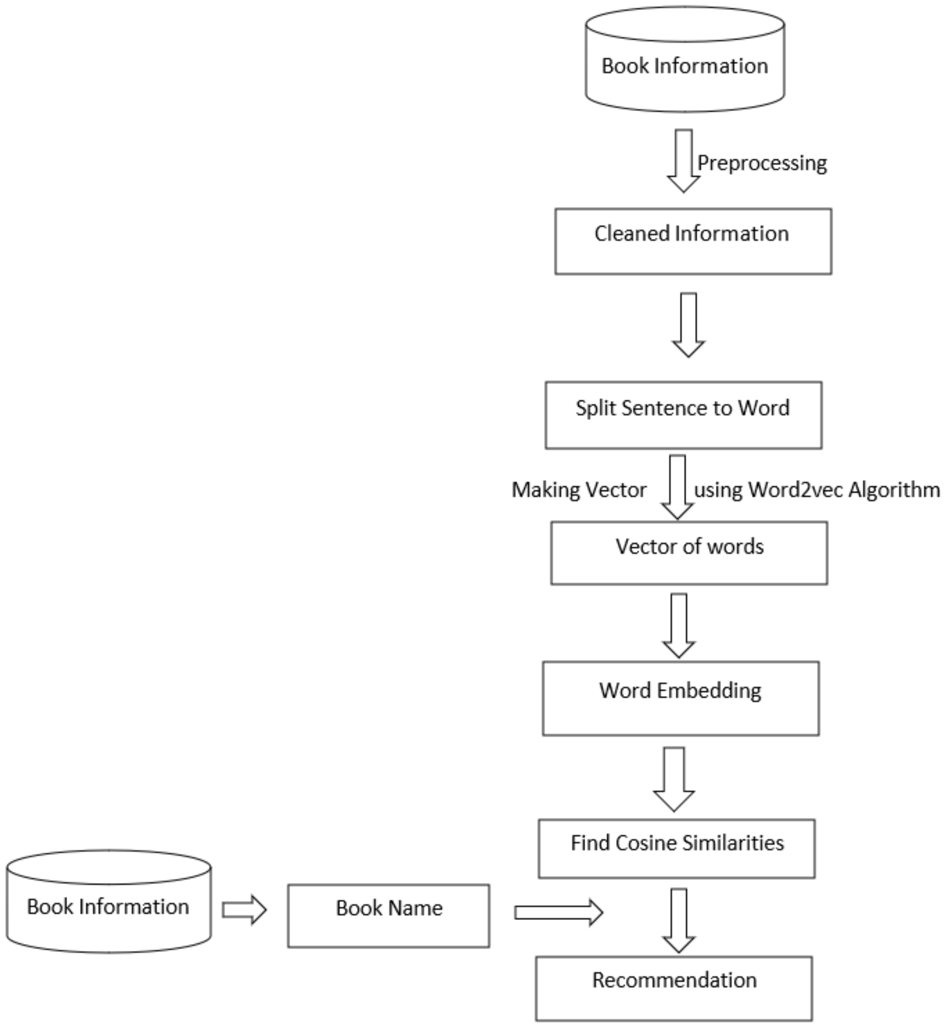
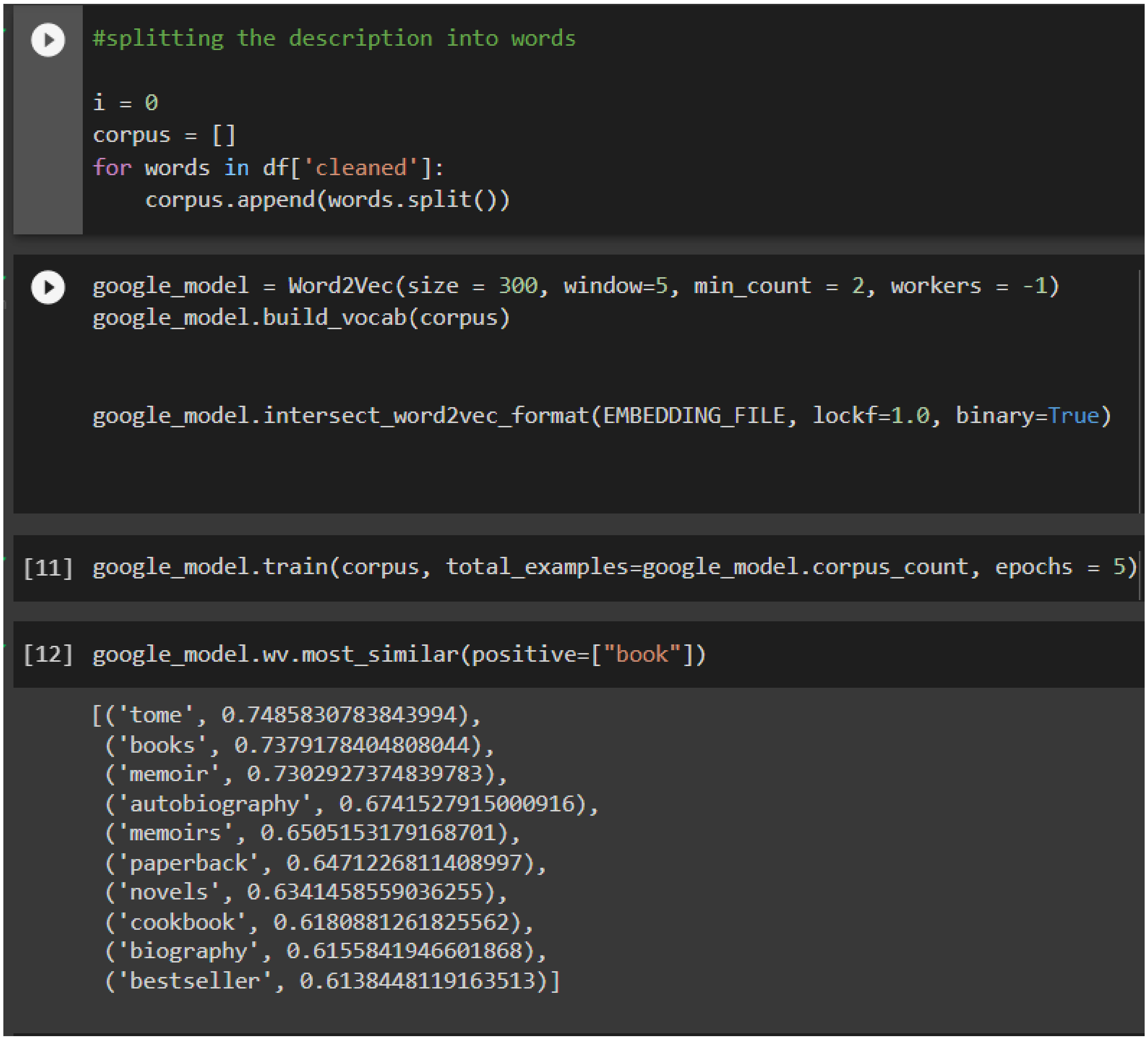


Fig 5.7: Recommendation Model Making

5.7 Word2vec Model Training with our Dataset



```
#splitting the description into words

i = 0
corpus = []
for words in df['cleaned']:
    corpus.append(words.split())
```

```
google_model = Word2Vec(size = 300, window=5, min_count = 2, workers = -1)
google_model.build_vocab(corpus)


google_model.intersect_word2vec_format(EMBEDDING_FILE, lockf=1.0, binary=True)
```

```
[11] google_model.train(corpus, total_examples=google_model.corpus_count, epochs = 5)
```

```
[12] google_model.wv.most_similar(positive=["book"])

     [('tome', 0.7485830783843994),
      ('books', 0.7379178404808044),
      ('memoir', 0.7302927374839783),
      ('autobiography', 0.6741527915000916),
      ('memoirs', 0.6505153179168701),
      ('paperback', 0.6471226811408997),
      ('novels', 0.6341458559036255),
      ('cookbook', 0.6180881261825562),
      ('biography', 0.6155841946601868),
      ('bestseller', 0.6138448119163513)]
```

5.8 Word Embedding

Word Embedding is a word representation type that allows machine learning algorithms to understand words with similar meanings. It is a language modeling and feature learning technique to map words into vectors of real numbers using neural networks, probabilistic models, or dimension reduction on the word co-occurrence matrix. Some word embedding models are Word2vec (Google), Glove (Stanford), and fastest (Facebook).

Word Embedding is also called as distributed semantic model or distributed represented or semantic vector space or vector space model. The word semantic which means categorizing similar words together. For example fruits like apple, mango, banana should be placed close whereas books will be far away from these words. In a broader sense, word embedding will create the vector of fruits which will be placed far away from vector representation of books.

5.9 Where is Word Embedding used?

Word embedding helps in feature generation, document clustering, text classification, and natural language processing tasks. Let us list them and have some discussion on each of these applications.

Compute similar words: Word embedding is used to suggest similar words to the word being subjected to the prediction model. Along with that it also suggests dissimilar words, as well as most common words.

Create a group of related words: It is used for semantic grouping which will group things of similar characteristic together and dissimilar far away.

Feature for text classification: Text is mapped into arrays of vectors which is fed to the model for training as well as prediction. Text-based classifier models cannot be trained on the string, so this will convert the text into machine trainable form. Further its features of building semantic help in text-based classification.

Document clustering: is another application where Word Embedding Word2vec is widely used

Natural language processing: There are many applications where word embedding is useful and wins over feature extraction phases such as parts of speech tagging, sentimental analysis, and syntactic analysis. Now we have got some knowledge of word embedding. Some light is also thrown on different models to implement word embedding. This whole Word Embedding tutorial is focused on one of the models (Word2vec)[10].

5.10 Word Embedding working Procedure

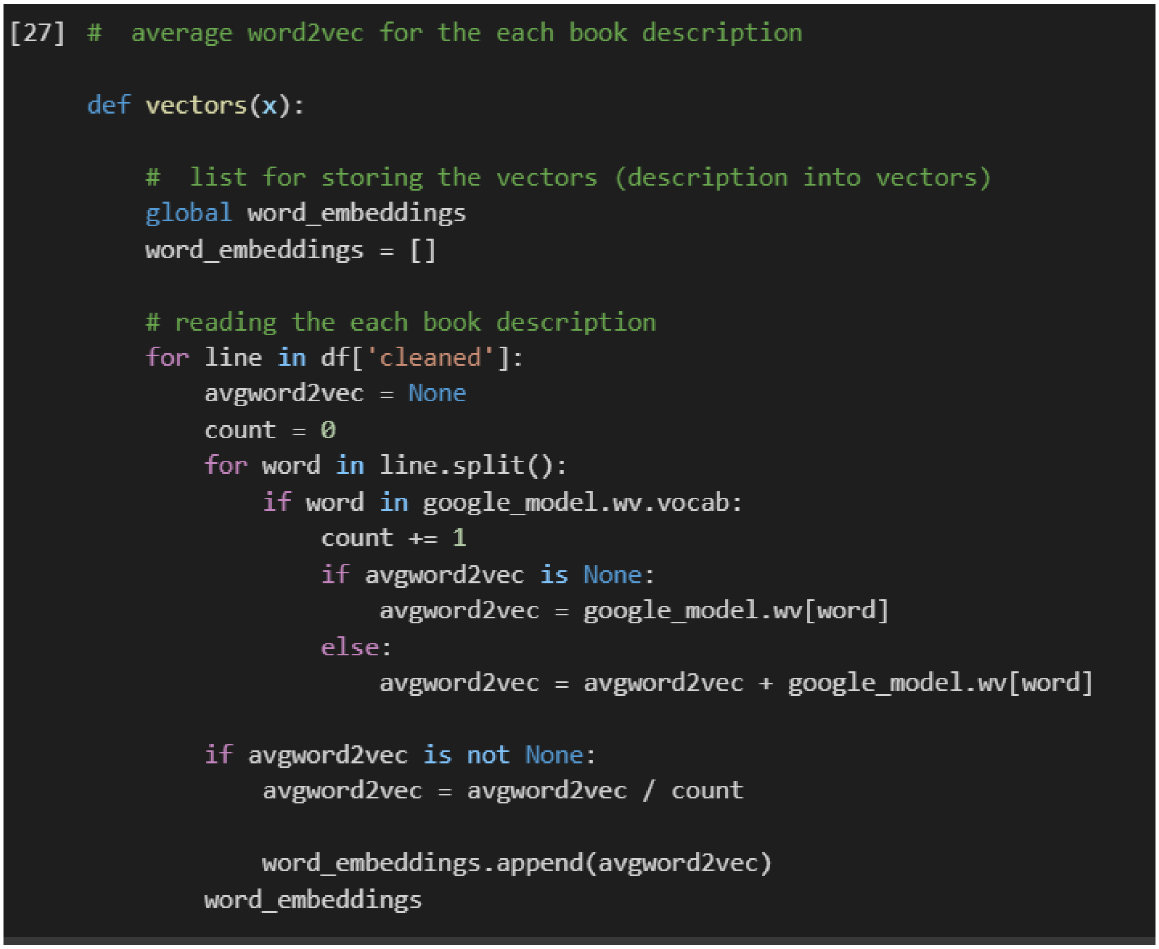
For example:

our president is a good leader he will not fail

our president is not a good leader he will fail

```
from sentence_transformers import SentenceTransformer
import numpy as np

def cosine_similarity(sentence_embeddings, ind_a, ind_b):
    s = sentence_embeddings
    return np.dot(s[ind_a], s[ind_b]) /
(np.linalg.norm(s[ind_a]) * np.linalg.norm(s[ind_b]))

model = SentenceTransformer('bert-base-nli-mean-tokens')

s0 = "our president is a good leader he will not fail"
s1 = "our president is not a good leader he will fail"
s2 = "our president is a good leader"
s3 = "our president will succeed"

sentences = [s0, s1, s2, s3]

sentence_embeddings = model.encode(sentences)

s = sentence_embeddings

print(f"{s0} <--> {s1}:
{cosine_similarity(sentence_embeddings, 0, 1)}")
print(f"{s0} <--> {s2}:
{cosine_similarity(sentence_embeddings, 0, 2)}")
print(f"{s0} <--> {s3}:
{cosine_similarity(sentence_embeddings, 0, 3)}")
```

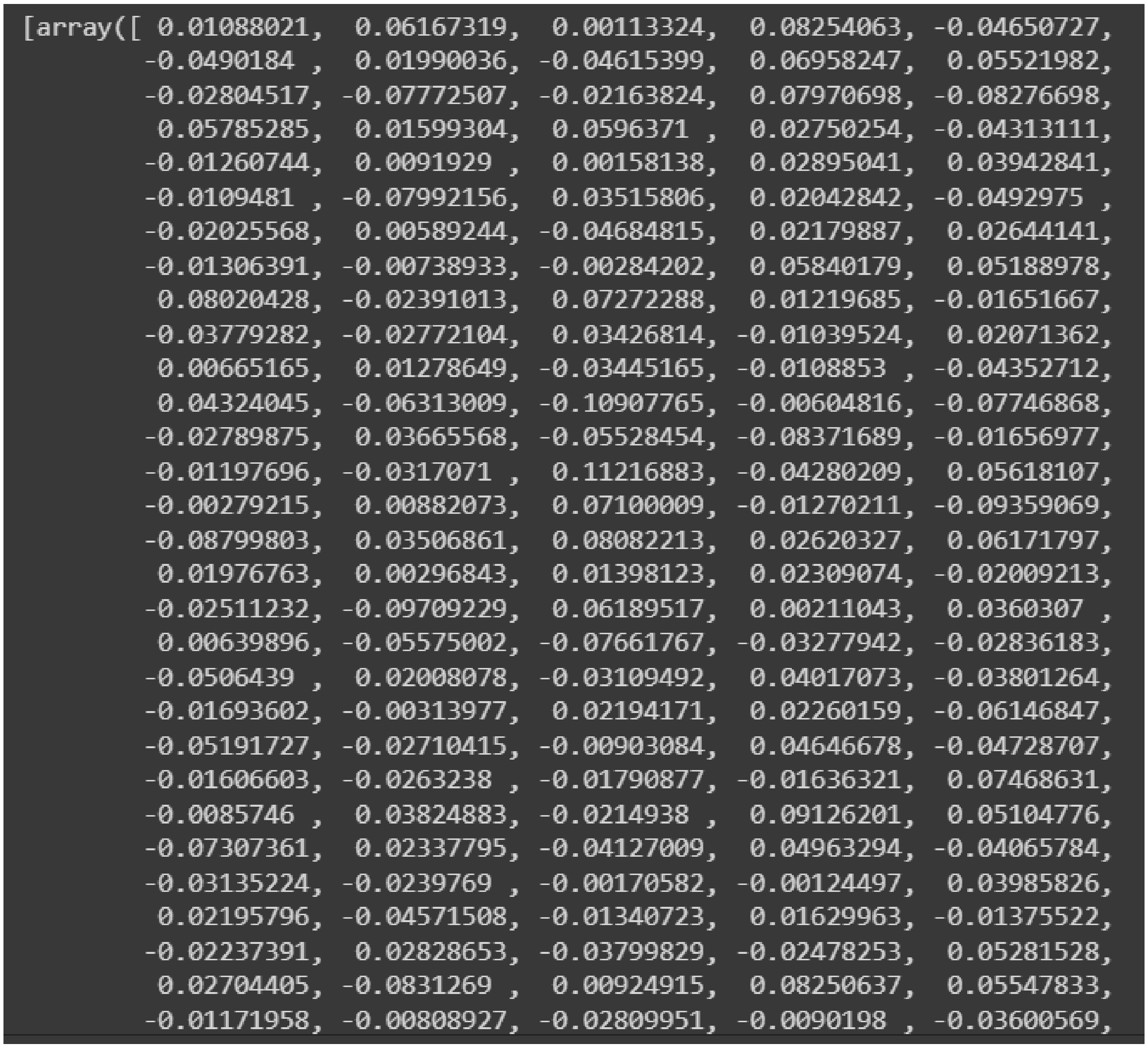
5.11 Generating WordEmbeddings



```
[27] #  average word2vec for the each book description

     def vectors(x):

         #  list for storing the vectors (description into vectors)
         global word_embeddings
         word_embeddings = []

         # reading the each book description
         for line in df['cleaned']:
             avgword2vec = None
             count = 0
             for word in line.split():
                 if word in google_model.wv.vocab:
                     count += 1
                     if avgword2vec is None:
                         avgword2vec = google_model.wv[word]
                     else:
                         avgword2vec = avgword2vec + google_model.wv[word]

             if avgword2vec is not None:
                 avgword2vec = avgword2vec / count

                 word_embeddings.append(avgword2vec)
             word_embeddings
```

Fig 5.9: Generating Word Embeddings

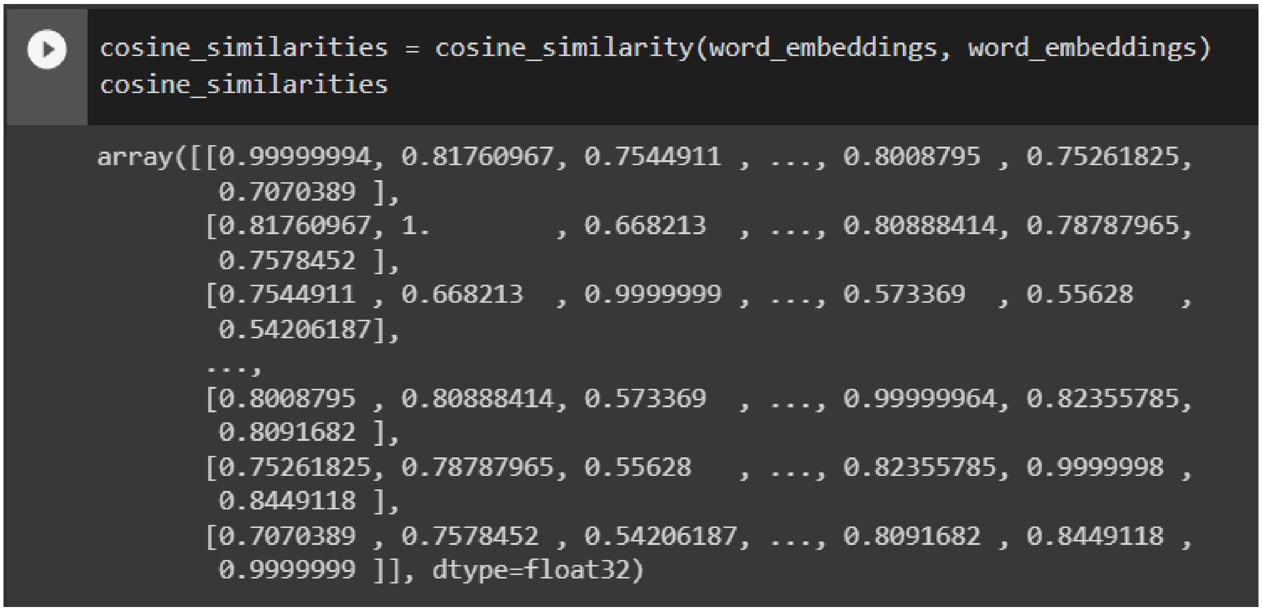
5.12 Distance Vector Between Books



```
[array([ 0.01088021,  0.06167319,  0.00113324,  0.08254063, -0.04650727,
        -0.0490184 ,  0.01990036, -0.04615399,  0.06958247,  0.05521982,
        -0.02804517, -0.07772507, -0.02163824,  0.07970698, -0.08276698,
         0.05785285,  0.01599304,  0.0596371 ,  0.02750254, -0.04313111,
        -0.01260744,  0.0091929 ,  0.00158138,  0.02895041,  0.03942841,
        -0.0109481 , -0.07992156,  0.03515806,  0.02042842, -0.0492975 ,
        -0.02025568,  0.00589244, -0.04684815,  0.02179887,  0.02644141,
        -0.01306391, -0.00738933, -0.00284202,  0.05840179,  0.05188978,
         0.08020428, -0.02391013,  0.07272288,  0.01219685, -0.01651667,
        -0.03779282, -0.02772104,  0.03426814, -0.01039524,  0.02071362,
         0.00665165,  0.01278649, -0.03445165, -0.0108853 , -0.04352712,
         0.04324045, -0.06313009, -0.10907765, -0.00604816, -0.07746868,
        -0.02789875,  0.03665568, -0.05528454, -0.08371689, -0.01656977,
        -0.01197696, -0.0317071 ,  0.11216883, -0.04280209,  0.05618107,
        -0.00279215,  0.00882073,  0.07100009, -0.01270211, -0.09359069,
        -0.08799803,  0.03506861,  0.08082213,  0.02620327,  0.06171797,
         0.01976763,  0.00296843,  0.01398123,  0.02309074, -0.02009213,
        -0.02511232, -0.09709229,  0.06189517,  0.00211043,  0.0360307 ,
         0.00639896, -0.05575002, -0.07661767, -0.03277942, -0.02836183,
        -0.0506439 ,  0.02008078, -0.03109492,  0.04017073, -0.03801264,
        -0.01693602, -0.00313977,  0.02194171,  0.02260159, -0.06146847,
        -0.05191727, -0.02710415, -0.00903084,  0.04646678, -0.04728707,
        -0.01606603, -0.0263238 , -0.01790877, -0.01636321,  0.07468631,
        -0.0085746 ,  0.03824883, -0.0214938 ,  0.09126201,  0.05104776,
        -0.07307361,  0.02337795, -0.04127009,  0.04963294, -0.04065784,
        -0.03135224, -0.0239769 , -0.00170582, -0.00124497,  0.03985826,
         0.02195796, -0.04571508, -0.01340723,  0.01629963, -0.01375522,
        -0.02237391,  0.02828653, -0.03799829, -0.02478253,  0.05281528,
         0.02704405, -0.0831269 ,  0.00924915,  0.08250637,  0.05547833,
        -0.01171958, -0.00808927, -0.02809951, -0.0090198 , -0.03600569,
```

Fig 5.10: Distance Vector Between

5.13 Cosine Similarities between books

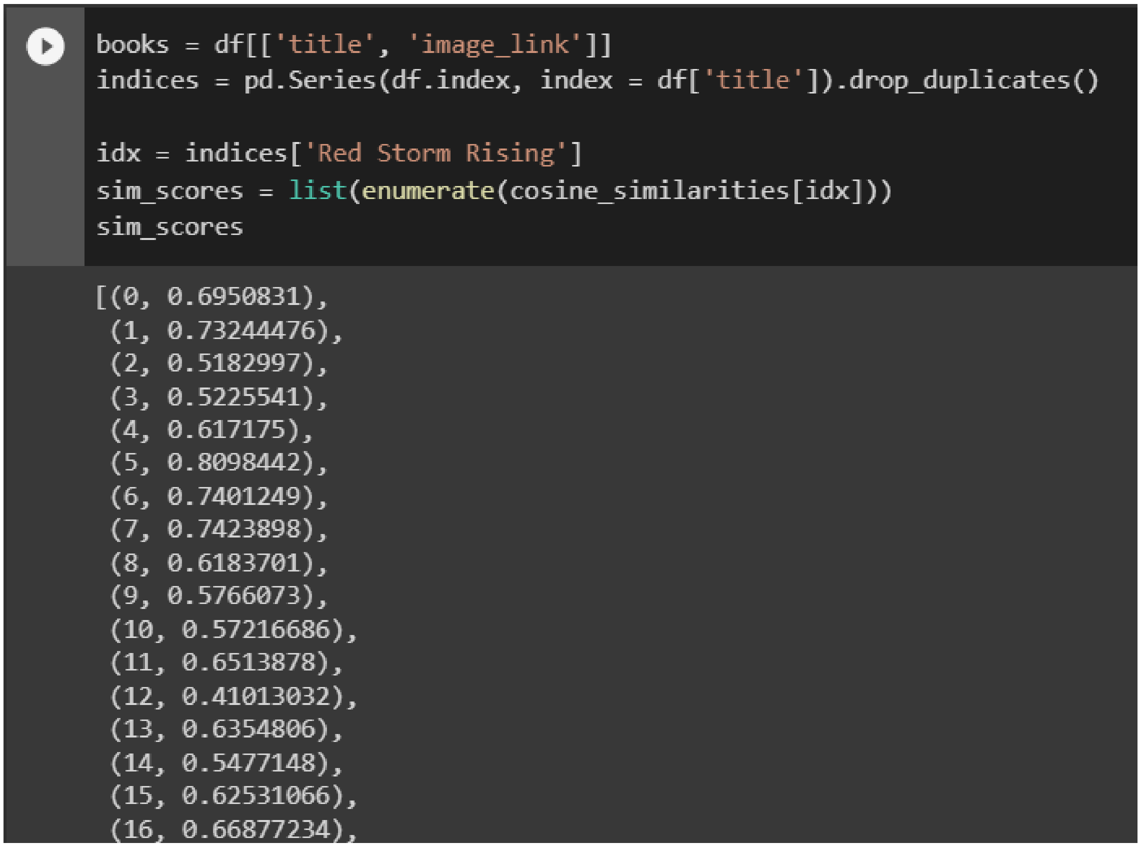


```
cosine_similarities = cosine_similarity(word_embeddings, word_embeddings)
cosine_similarities
```

```
array([[0.99999994, 0.81760967, 0.7544911 , ..., 0.8008795 , 0.75261825,
        0.7070389 ],
       [0.81760967, 1.        , 0.668213  , ..., 0.80888414, 0.78787965,
        0.7578452 ],
       [0.7544911 , 0.668213  , 0.9999999 , ..., 0.573369  , 0.55628   ,
        0.54206187],
       ...,
       [0.8008795 , 0.80888414, 0.573369  , ..., 0.99999964, 0.82355785,
        0.8091682 ],
       [0.75261825, 0.78787965, 0.55628   , ..., 0.82355785, 0.9999998 ,
        0.8449118 ],
       [0.7070389 , 0.7578452 , 0.54206187, ..., 0.8091682 , 0.8449118 ,
        0.9999999 ]], dtype=float32)
```

Fig 5.11: Cosine Similarities between books Books

5.14 Book Id and similarities with other books with the book “Red Strom Rising

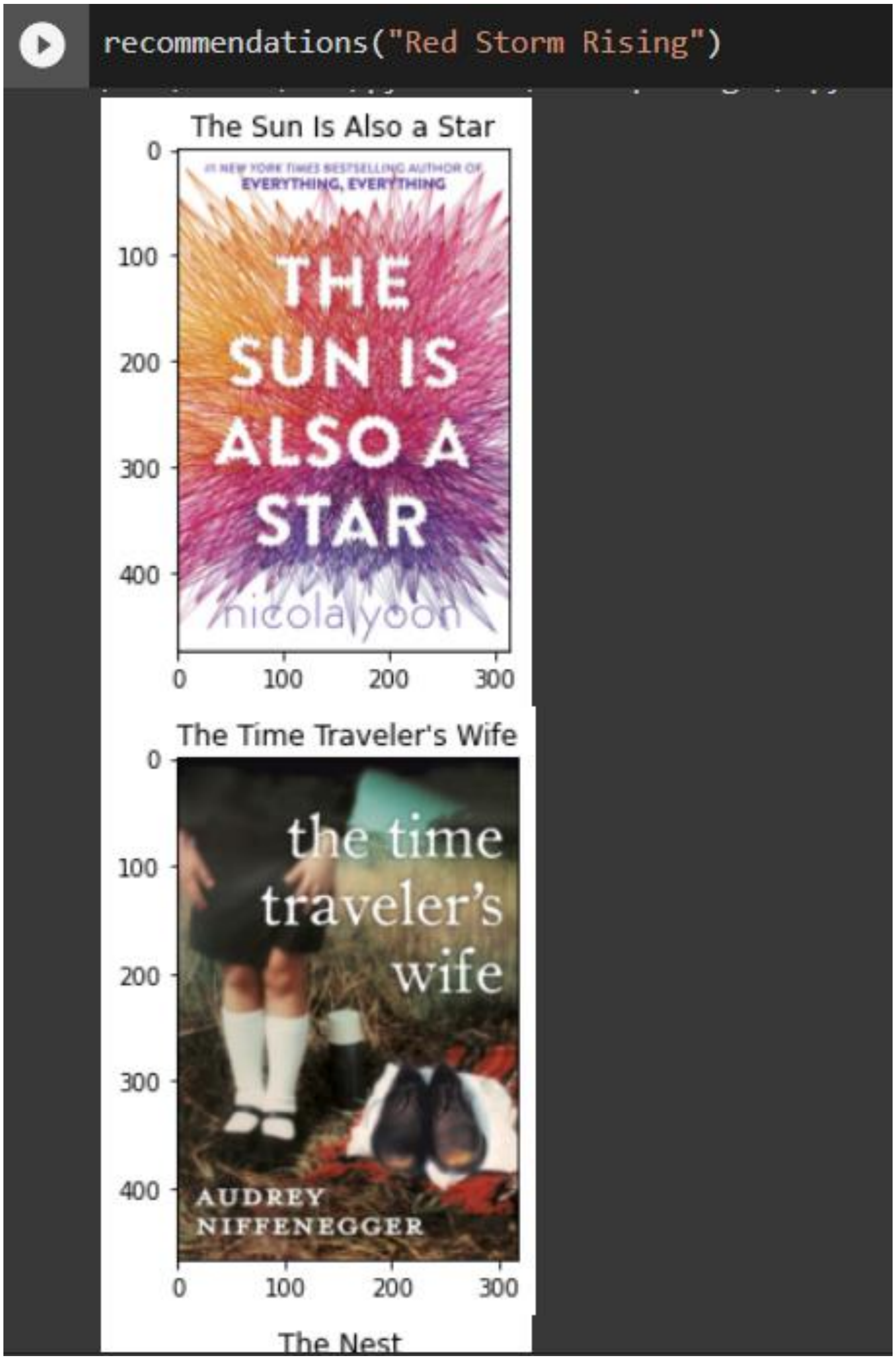


```
books = df[['title', 'image_link']]
indices = pd.Series(df.index, index = df['title']).drop_duplicates()

idx = indices['Red Storm Rising']
sim_scores = list(enumerate(cosine_similarities[idx]))
sim_scores
```

```
[(0, 0.6950831),
 (1, 0.73244476),
 (2, 0.5182997),
 (3, 0.5225541),
 (4, 0.617175),
 (5, 0.8098442),
 (6, 0.7401249),
 (7, 0.7423898),
 (8, 0.6183701),
 (9, 0.5766073),
 (10, 0.57216686),
 (11, 0.6513878),
 (12, 0.41013032),
 (13, 0.6354806),
 (14, 0.5477148),
 (15, 0.62531066),
 (16, 0.66877234),
```

Fig 5.12: Book Id and similarities with other books with the book “Red Strom Rising”

5.15 Recommendation



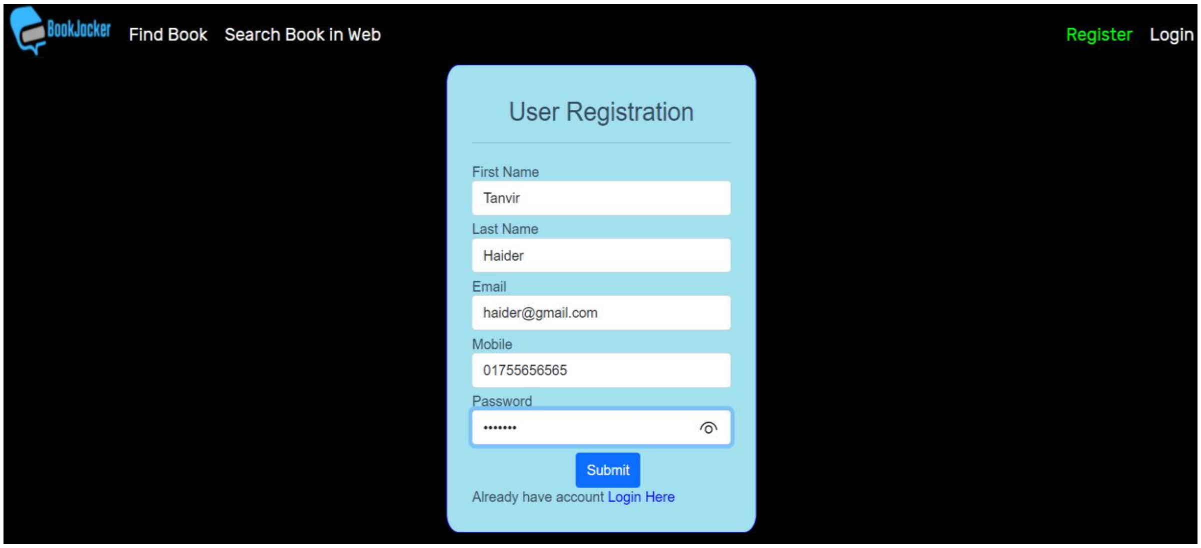
```
recommendations("Red Storm Rising")
```

Fig 5.13: Getting Recommendation For “Red Strom Rising

CHAPTER 6: IMPLEMENTATION

6.1 Implementation

6.1.1 Website Registration page



First Name: Tanvir

Last Name: Haider

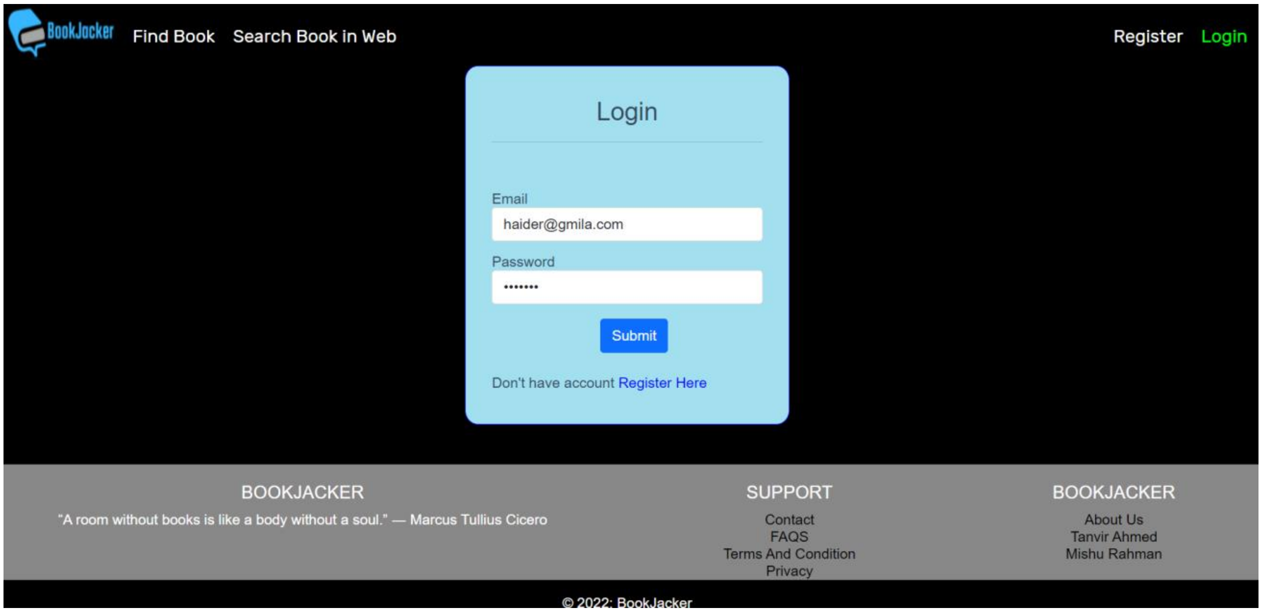
Email: haider@gmail.com

Mobile: 01755656565

Password: •••••••

Fig 6.1: Registration Page

6.1.2 Website Login page



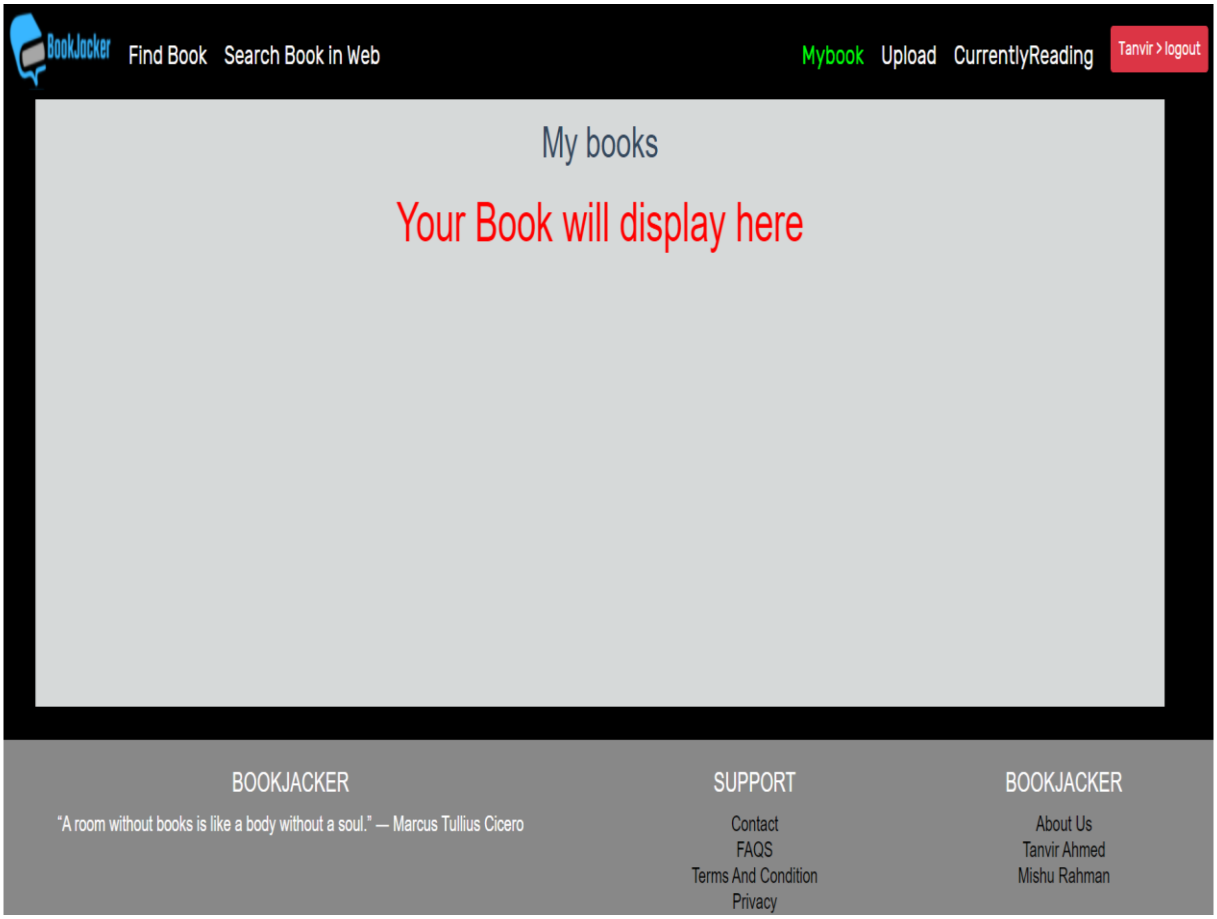
Email: haider@gmila.com

Password: •••••••

"A room without books is like a body without a soul." — Marcus Tullius Cicero

Fig 6.2: Registration Page

6.1.3 Website My book page with no books



My books

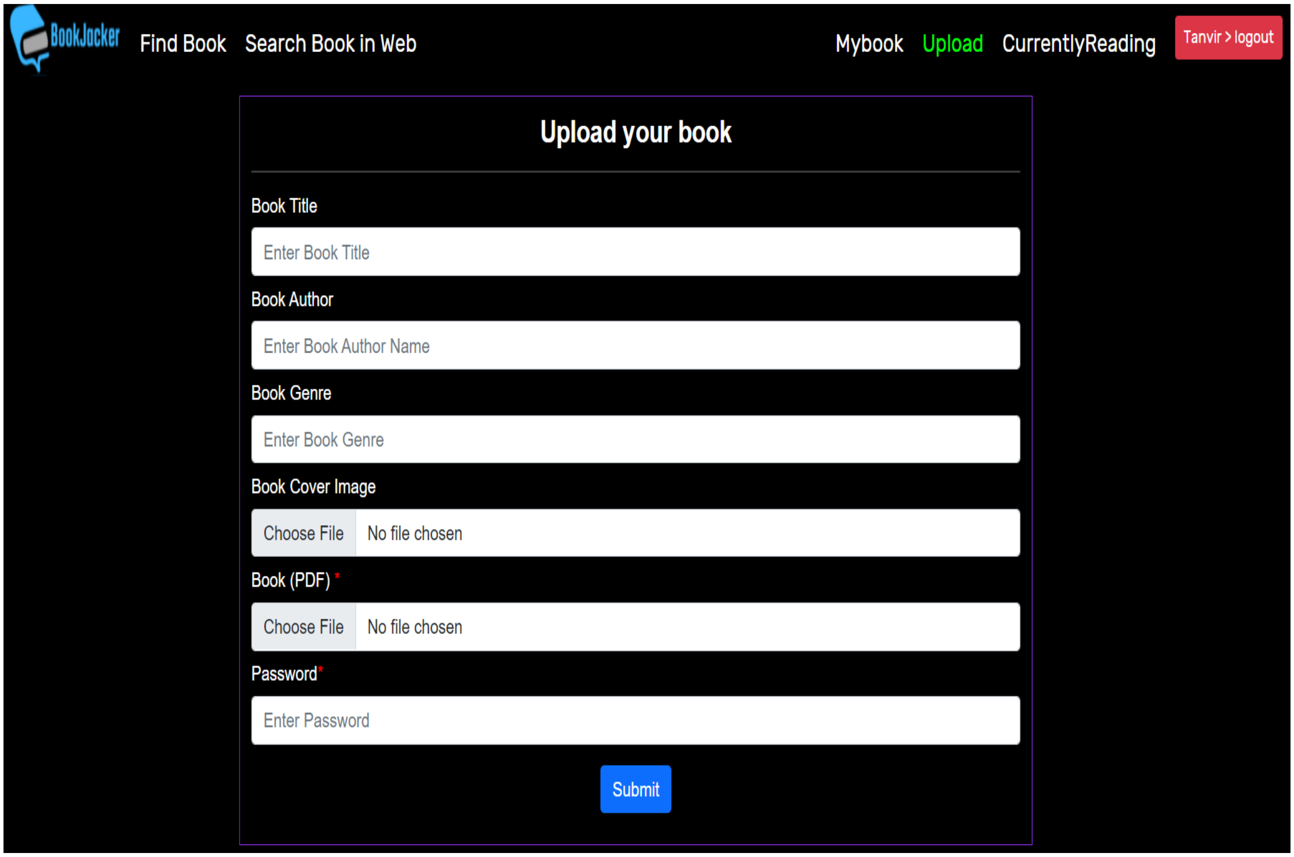
Your Book will display here

BOOKJACKER

“A room without books is like a body without a soul.” — Marcus Tullius Cicero

Fig 6.3: My book Page with no book

6.1.4 Website Upload page



Upload your book

Book Title

Enter Book Title

Book Author

Enter Book Author Name

Book Genre

Enter Book Genre

Book Cover Image

Choose File No file chosen

Book (PDF) *

Choose File No file chosen

Password*

Enter Password

Fig 6.4: Book Upload Page

6.1.5 Website My Book Page with books

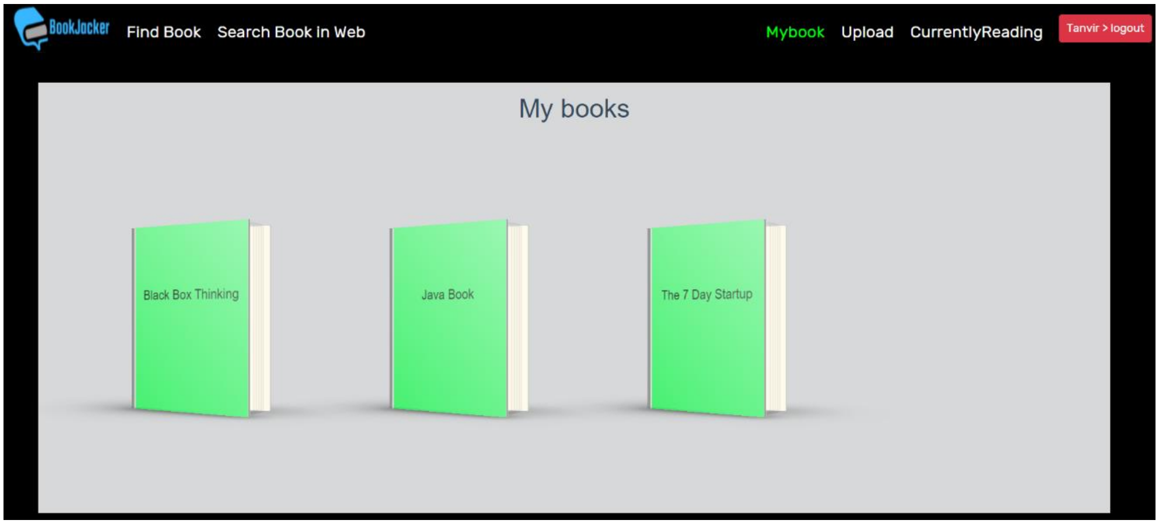
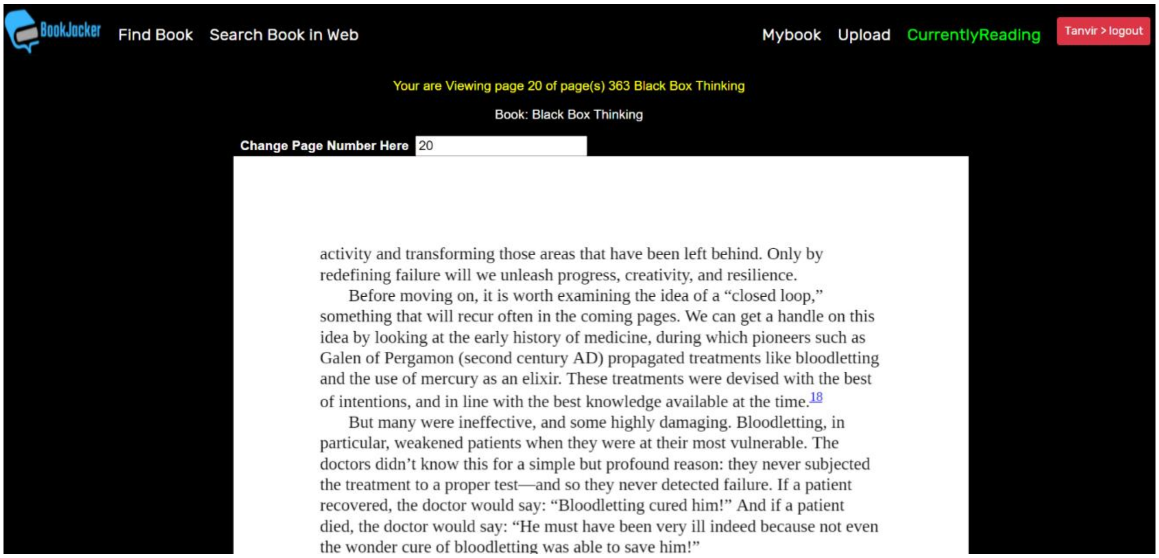


Fig 6.5: My Book Page with books

6.1.6 Website Currently Reading Book



Your are Viewing page 20 of page(s) 363 Black Box Thinking

Book: Black Box Thinking

Change Page Number Here 20

activity and transforming those areas that have been left behind. Only by redefining failure will we unleash progress, creativity, and resilience.

Before moving on, it is worth examining the idea of a "closed loop," something that will recur often in the coming pages. We can get a handle on this idea by looking at the early history of medicine, during which pioneers such as Galen of Pergamon (second century AD) propagated treatments like bloodletting and the use of mercury as an elixir. These treatments were devised with the best of intentions, and in line with the best knowledge available at the time.[18]

But many were ineffective, and some highly damaging. Bloodletting, in particular, weakened patients when they were at their most vulnerable. The doctors didn't know this for a simple but profound reason: they never subjected the treatment to a proper test—and so they never detected failure. If a patient recovered, the doctor would say: "Bloodletting cured him!" And if a patient died, the doctor would say: "He must have been very ill indeed because not even the wonder cure of bloodletting was able to save him!"

Fig 6.6: Currently Reading Book

6.1.7 Website Home Page

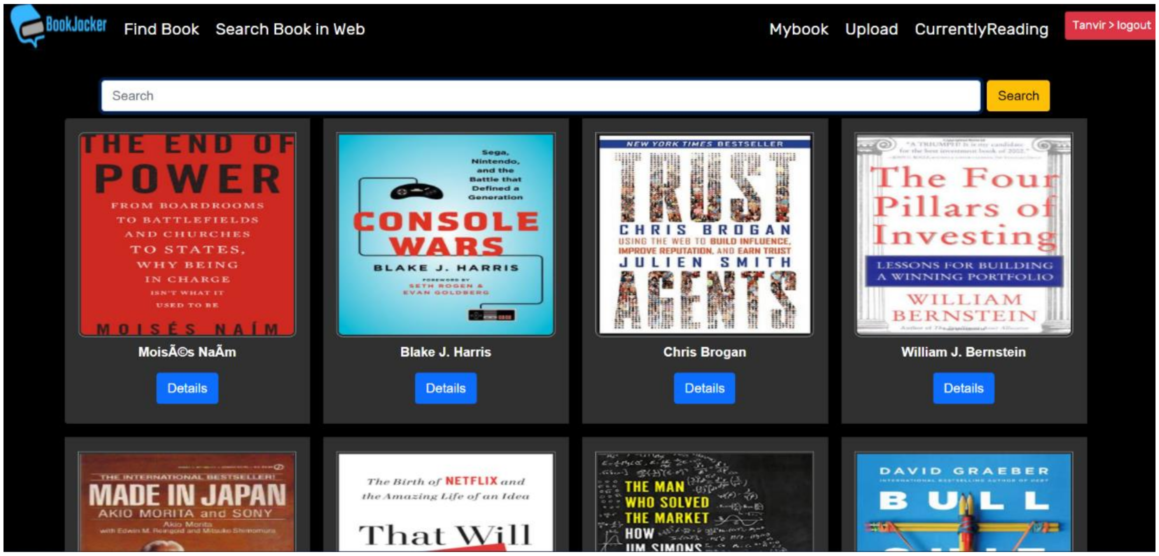
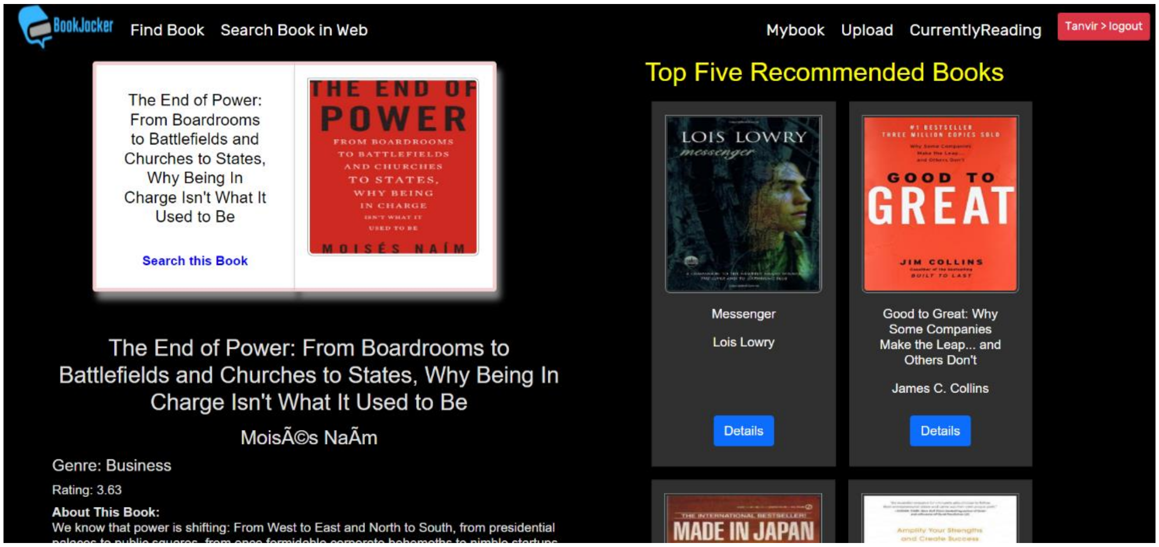


Fig 6.7: Home Page

6.1.8 Website Recommendation Page



The End of Power: From Boardrooms to Battlefields and Churches to States, Why Being In Charge Isn't What It Used to Be

Top Five Recommended Books

Messenger

Lois Lowry

Good to Great: Why Some Companies Make the Leap... and Others Don't

James C. Collins

The End of Power: From Boardrooms to Battlefields and Churches to States, Why Being In Charge Isn't What It Used to Be

MoisÃ©s NaÃm

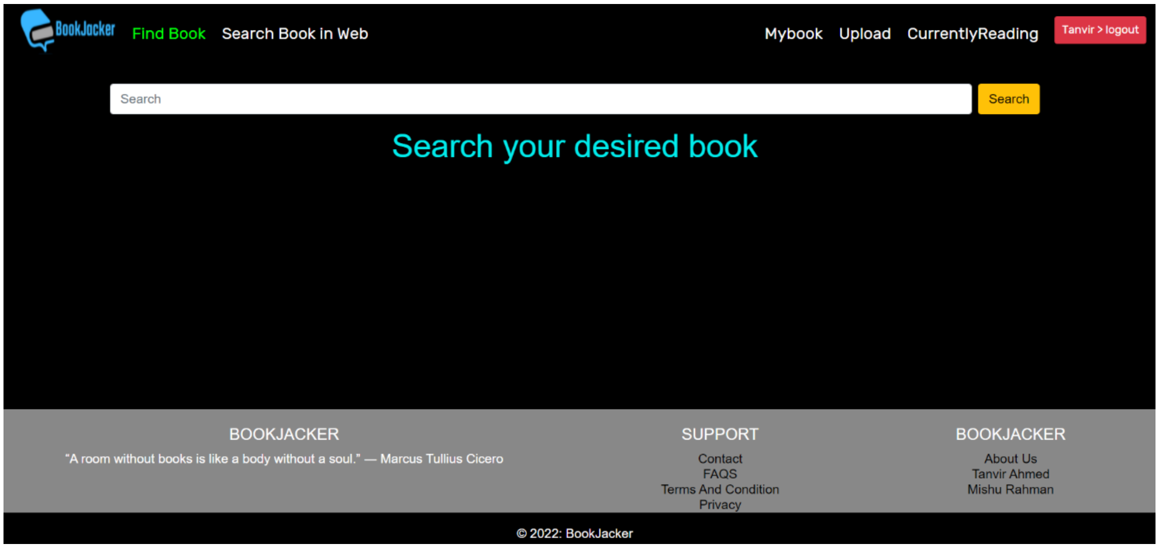
Genre: Business

Rating: 3.63

About This Book:
We know that power is shifting: From West to East and North to South, from presidential

Fig 6.8: Recommendation Page

6.1.9 Website Find Page



Search your desired book

BOOKJACKER

"A room without books is like a body without a soul." — Marcus Tullius Cicero

SUPPORT

Contact
FAQS
Terms And Condition
Privacy

BOOKJACKER

About Us
Tanvir Ahmed
Mishu Rahman

© 2022: BookJacker

Fig 6.9: Find Page

6.1.10 Website Book searching in web

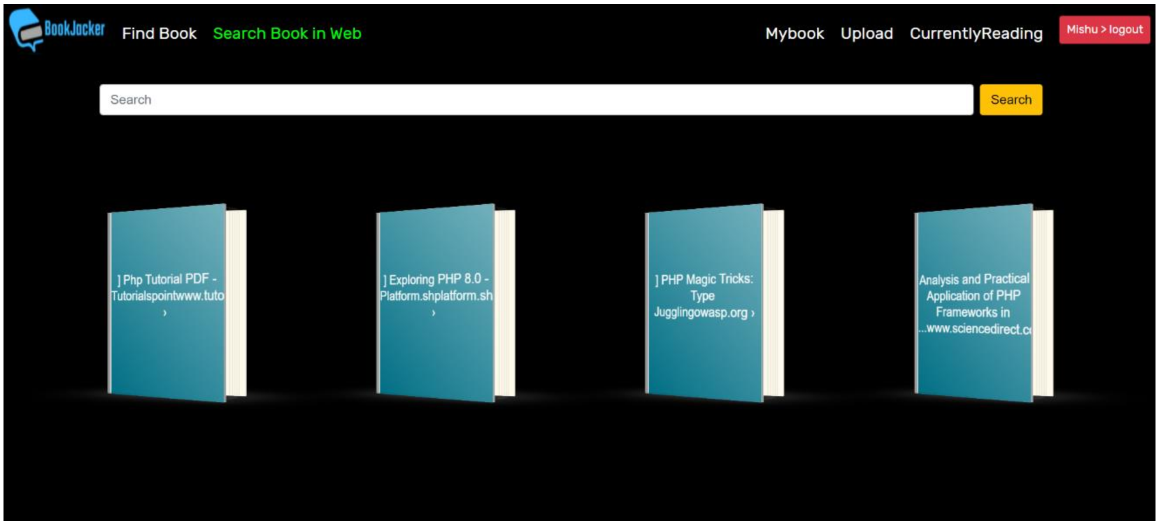

BookJacker
Find Book
Search Book in Web
Mybook
Upload
CurrentlyReading
Mishu > logout
Search
Search
] Php Tutorial PDF - Tutorialspointwww.tuto ›
] Exploring PHP 8.0 - Platform.shplatform.sh ›
] PHP Magic Tricks: Type Jugglingowasp.org ›
Analysis and Practical Application of PHP Frameworks in ...www.sciencedirect.c

Fig 6.10: Book searching in web

6.1.11 Website Book Download Page

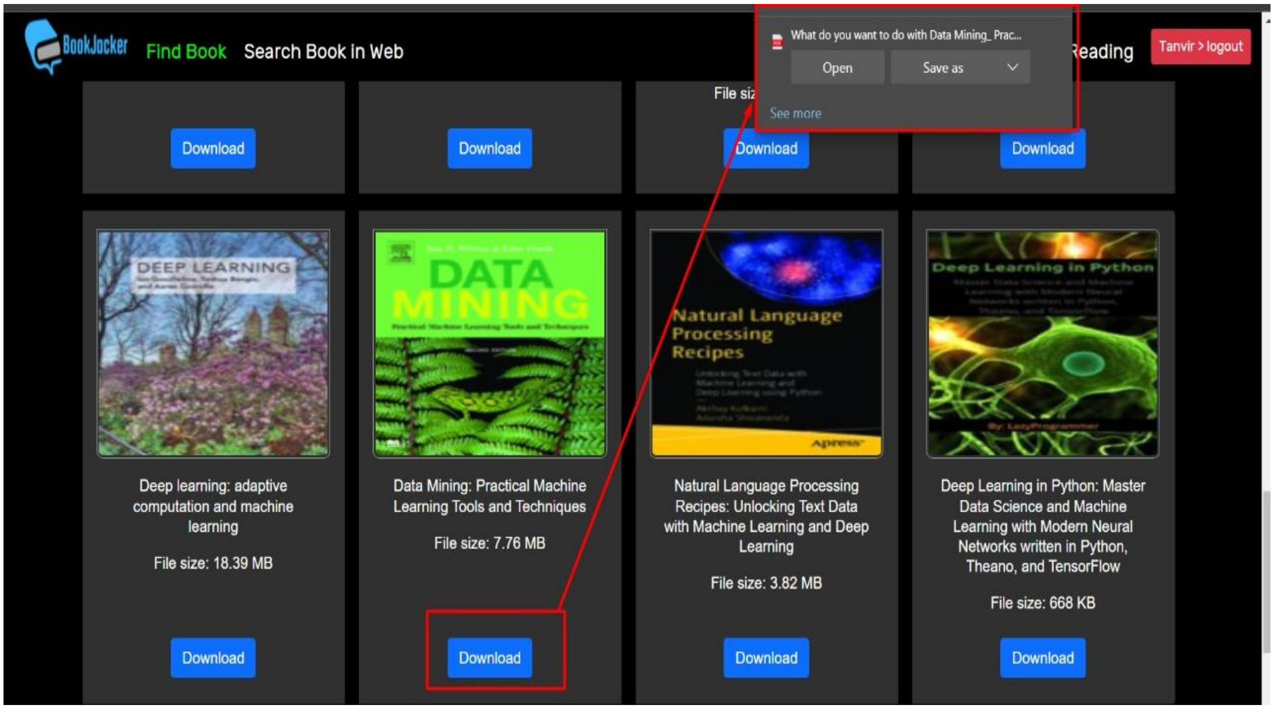

BookJacker
Find Book
Search Book in Web
What do you want to do with Data Mining_Prac...
Open
Save as
See more
File siz
Reading
Tanvir > logout
Download
Download
Download
Download
Deep learning: adaptive computation and machine learning
File size: 18.39 MB
Download
Data Mining: Practical Machine Learning Tools and Techniques
File size: 7.76 MB
Download
Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning
File size: 3.82 MB
Download
Deep Learning in Python: Master Data Science and Machine Learning with Modern Neural Networks written in Python, Theano, and TensorFlow
File size: 668 KB
Download

Fig 6.11: Book Download

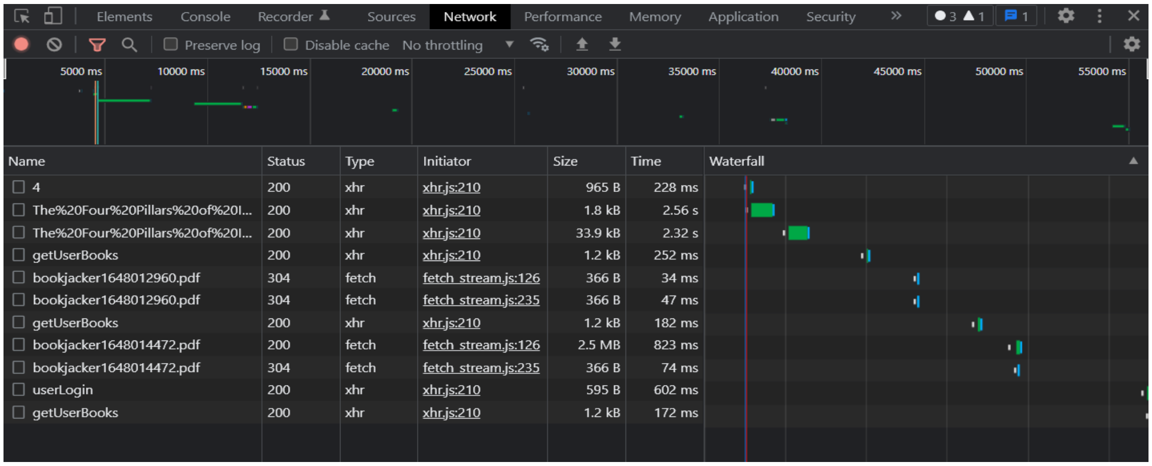
6.2 Testing

6.2.1 Unit Testing

Separately each and every functional module is tested and the result of that module is analyzed. The module interface is tested to ensure that the information flows in and out of the program under unit test. Each unit of the software is tested to verify that the detailed design for the unit has been correctly implemented.

6.2.2 Network Testing

Testing report of bookjacker view result in a table below. It shows that all the components are loading without error.

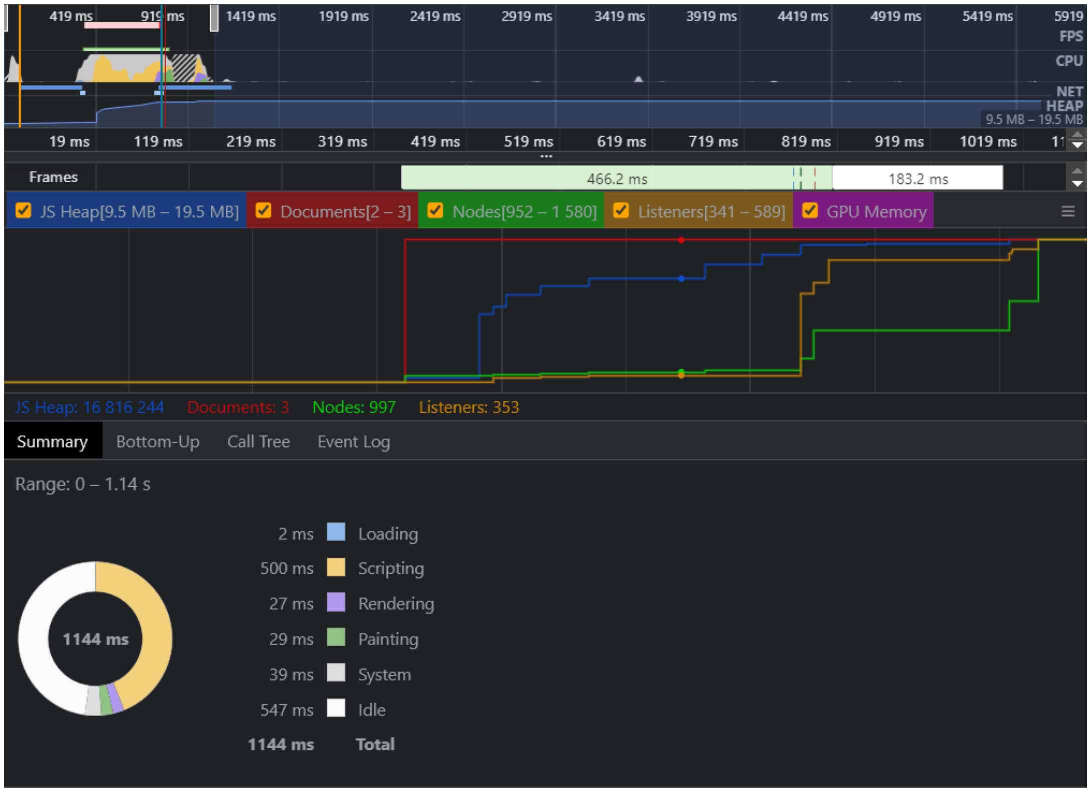


Name	Status	Type	Initiator	Size	Time	Waterfall
☐ 4	200	xhr	xhr.js:210	965 B	228 ms	
☐ The%20Four%20Pillars%20of%20I...	200	xhr	xhr.js:210	1.8 kB	2.56 s	
☐ The%20Four%20Pillars%20of%20I...	200	xhr	xhr.js:210	33.9 kB	2.32 s	
☐ getUserBooks	200	xhr	xhr.js:210	1.2 kB	252 ms	
☐ bookjacker1648012960.pdf	304	fetch	fetch_stream.js:126	366 B	34 ms	
☐ bookjacker1648012960.pdf	304	fetch	fetch_stream.js:235	366 B	47 ms	
☐ getUserBooks	200	xhr	xhr.js:210	1.2 kB	182 ms	
☐ bookjacker1648014472.pdf	200	fetch	fetch_stream.js:126	2.5 MB	823 ms	
☐ bookjacker1648014472.pdf	304	fetch	fetch_stream.js:235	366 B	74 ms	
☐ userLogin	200	xhr	xhr.js:210	595 B	602 ms	
☐ getUserBooks	200	xhr	xhr.js:210	1.2 kB	172 ms	

Fig 6.2.2: Network Testing

6.2.3 My Book Page Performance Testing

Testing report of bookjacker My Book Page View result in a table below. It shows that to load all the components it took only 1144ms. And it is in loading state only for 2 ms.



JS Heap: 16 816 244 Documents: 3 Nodes: 997 Listeners: 353

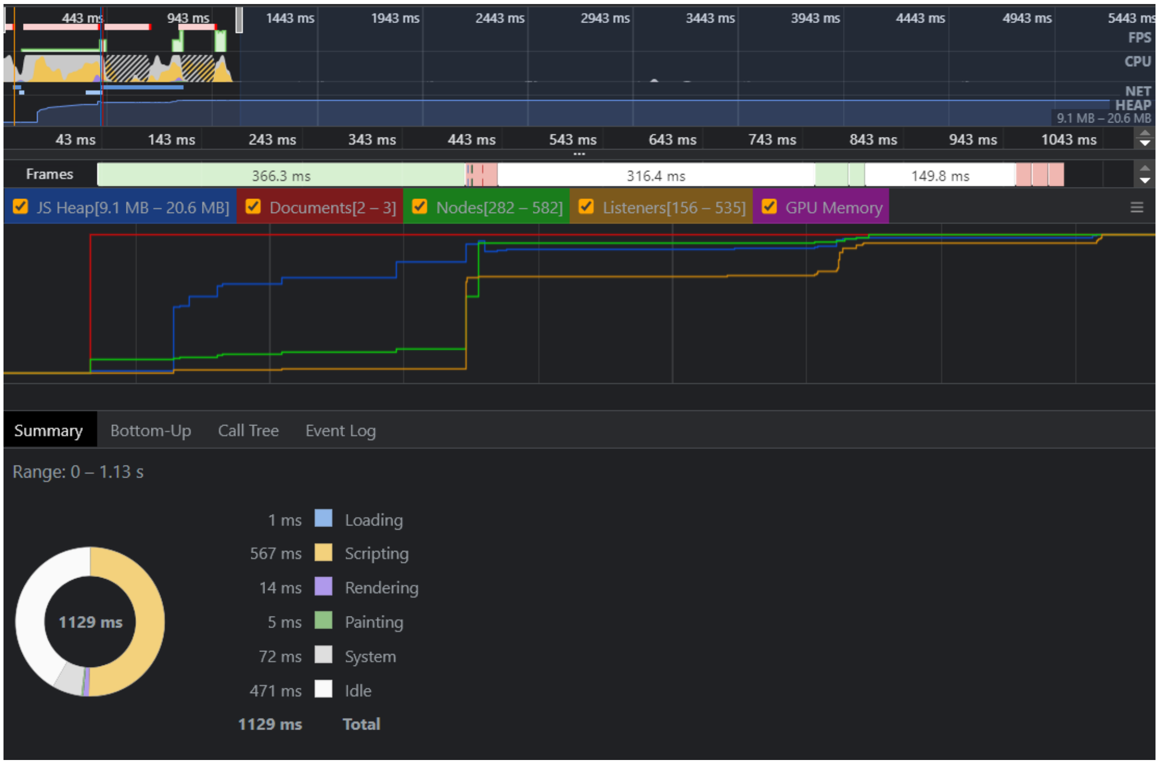
Range: 0 – 1.14 s

2 ms	🟦	Loading
500 ms	🟨	Scripting
27 ms	🟪	Rendering
29 ms	🟩	Painting
39 ms	⬜	System
547 ms	⬜	Idle
1144 ms		**Total**

Fig 6.2.3: My Book Page Performance Testing

6.2.4 Upload Page Performance Testing

Testing report of bookjacker Upload Page View result in a table below. It shows that to load all the components it took only 1129ms. And it is in loading state only for 1 ms.

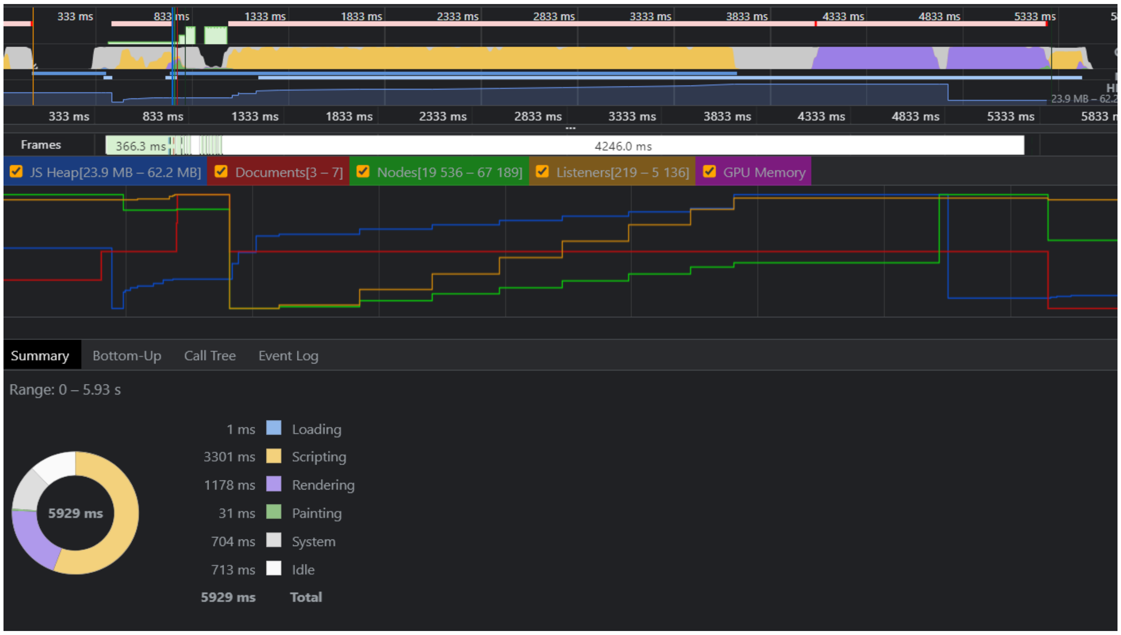


```
Range: 0 – 1.13 s

   1 ms  Loading
 567 ms  Scripting
  14 ms  Rendering
   5 ms  Painting
  72 ms  System
 471 ms  Idle
1129 ms  Total
```

Fig 6.2.4: Upload Page Performance Testing

6.2.5 Recommendation Page Performance Testing

Testing report of bookjacker Recommendation Page View result in a table below. It shows that to load all the components it took only 5929ms. And it is in loading state only for 1 ms. It took longer time because it load about 2300 book data at a time.



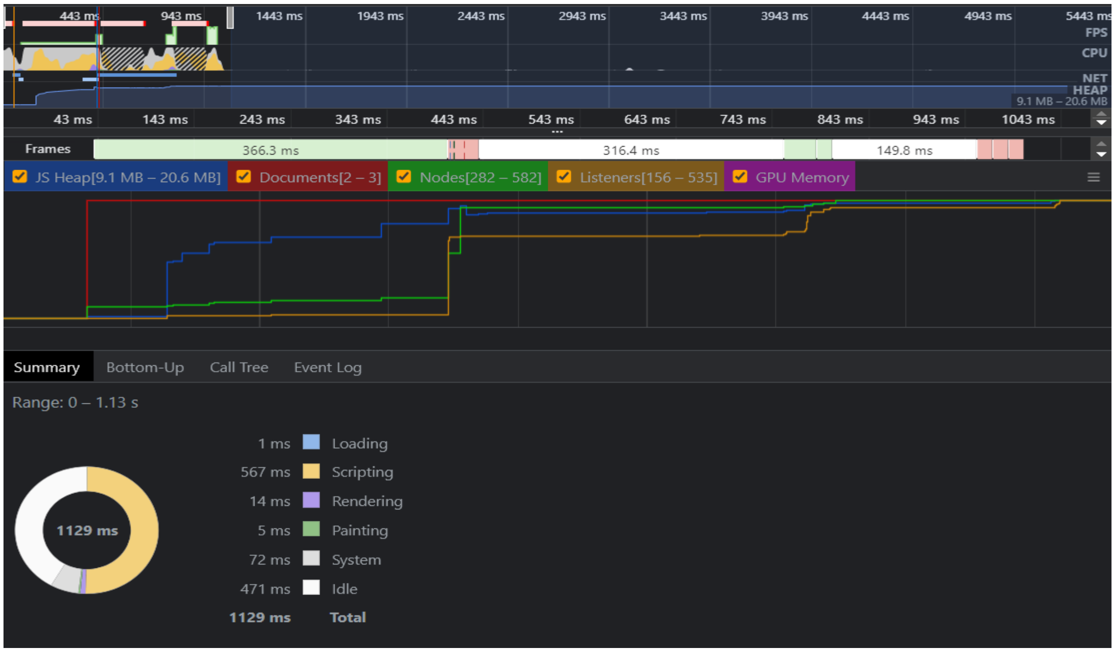
Range: 0 – 5.93 s

1 ms	Loading
3301 ms	Scripting
1178 ms	Rendering
31 ms	Painting
704 ms	System
713 ms	Idle
5929 ms	**Total**

Fig 6.2.5: Recommendation Page Performance Testing

6.2.6 Search Page Performance Testing

Testing report of bookjacker Search Page View result in a table below. It shows that to load all the components it took only 1129ms. And it is in loading state only for 1 ms.



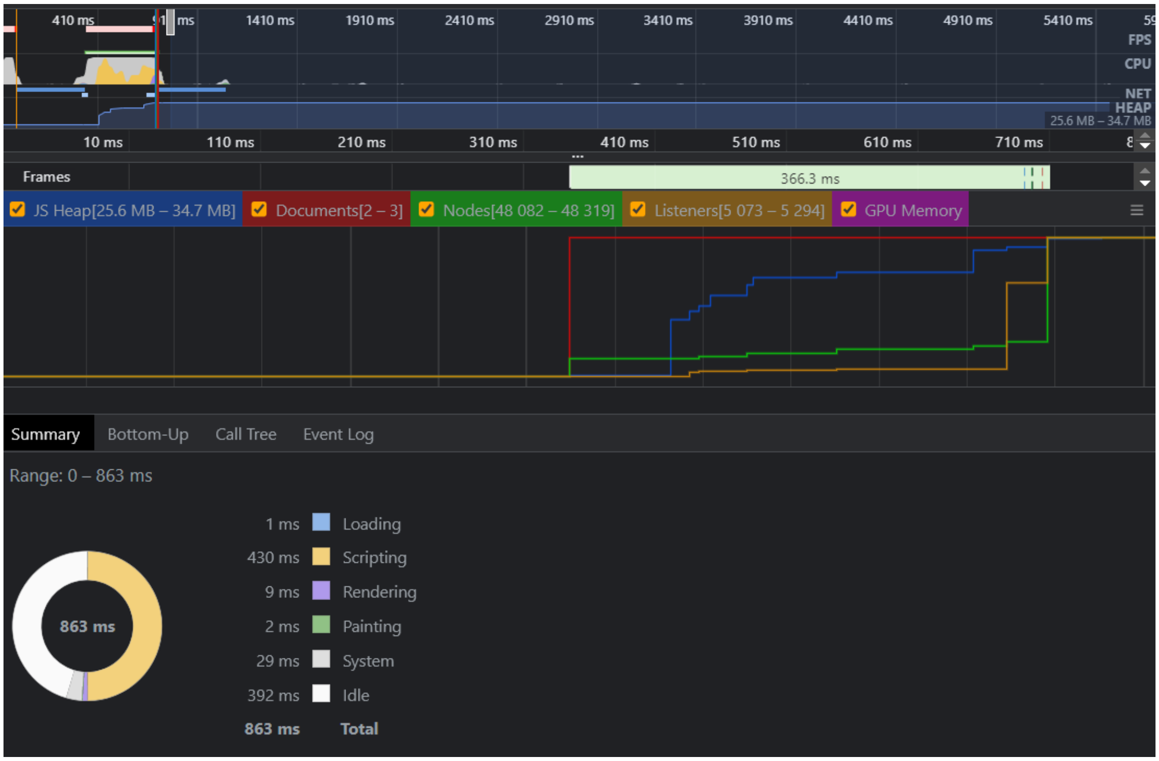
Range: 0 – 1.13 s

1 ms	Loading
567 ms	Scripting
14 ms	Rendering
5 ms	Painting
72 ms	System
471 ms	Idle
1129 ms	**Total**

Fig 6.2.6: Search Page Performance Testing

6.2.7 Currently Reading Page Performance Testing

Testing report of bookjacker Search Page View result in a table below. It shows that to load all the components it took only 863ms. And it is in loading state only for 1 ms.
That is much faster.



Range: 0 – 863 ms

```
  1 ms  🟦 Loading
430 ms  🟨 Scripting
  9 ms  🟪 Rendering
  2 ms  🟩 Painting
 29 ms  ⬜ System
392 ms  ⬜ Idle
863 ms     Total
```

Fig 6.2.6: Currently Reading Page Performance Testing

CHAPTER 7: CONCLUSION

7.1 Conclusion

The system provides platform to search and read the books online. The books are recommended to the users using word2vec model. This system recommends based on book description. Content based recommendation is one of the most familiar, most widely implemented and most mature of the technologies. This recommender system uses word2vec model which works using distance vector of words.

7.2 Future Work

In future in this system book sharing feature can be added. Book share can be done privately or publicly or in a group. This system recommends based on book description. In the future, behavior-based collaborative filtering can be used to suggest the most relevant books to users. In this process of recommendation, data of book rating and number of downloads can be used. To give better experience to user mobile application version of this project can be made. To search pdf a better crawler may use.

Reference

[1] S. S. Sohail, J. Siddiqui and R. Ali, "Book recommendation system using opinion mining technique," 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, 2013, pp. 1609-1614, doi: 10.1109/ICACCI.2013.6637421.

[2] P. Mathew, B. Kuriakose and V. Hegde, "Book Recommendation System through content based and collaborative filtering method," 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, 2016, pp. 47-52, doi: 10.1109/SAPIENCE.2016.7684166.

[3] S. Kanetkar, A. Nayak, S. Swamy and G. Bhatia, "Web-based personalized hybrid book recommendation system," 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), Unnao, 2014, pp. 1-5, doi: 10.1109/ICAETR.2014.7012952.

[4] M. Kommineni, P. Alekhya, T. M. Vyshnavi, V. Aparna, K. Swetha and V. Mounika, "Machine Learning based Efficient Recommendation System for Book Selection using User based Collaborative Filtering Algorithm," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 2020, pp. 66-71, doi: 10.1109/ICISC47916.2020.9171222.

[5] R. Rahutomo, A. S. Perbangsa, H. Soeparno and B. Pardamean, "Embedding Model Design for Producing Book Recommendation," 2019 International Conference on Information Management and Technology (ICIMTech), Jakarta/Bali, Indonesia, 2019, pp. 537-541, doi: 10.1109/ICIMTech.2019.8843769.

[6] Sarma, Dhiman & Mittra, Tanni & Shahadat, Mohammad. (2021). Personalized Book Recommendation System using Machine Learning Algorithm. International Journal of Advanced Computer Science and Applications. 12. 10.14569/IJACSA.2021.0120126.

[7] “ReactJS Tutorial - Javatpoint.” www.javatpoint.com/reactjs-tutorial. Accessed 12 Nov. 2021.

[8] ARYAL, DEEPAK, and MADAN BHANDARI. “BOOK BUZZ- an ONLINE BOOK STORE with RECOMMENDATION SYSTEM USING COLLABORATIVE-FILTERING.” www.coursehero.com/file/68468083/FINAL-YEAR-PROJECT-ON-ONLINE-BOOK-STOREpdf/. Accessed 20 Nov. 2021.

[9] Jevtic, Goran. “What Is SDLC? How the Software Development Life Cycle Works.” PhoenixNAP Global IT Services, phoenixnap.com/blog/software-development-lifecycle#:~:text=What%20is%20the%20Software%20 Development. Accessed 22 Nov. 2021.

[10] Johnson, Daniel. “Word Embedding Tutorial: Word2vec with Genism [EXAMPLE].” Www.guru99.com, 8 Mar. 2022, www.guru99.com/word-embedding-word2vec.html. Accessed 3 Jan. 2022.