A
Project Report
On

E-commerce Book Recommendation

Developed by

Yami Naik Department of IT, DD University
Yash Naik Department of IT, DD University

Guided By

Internal Guide:
Prof. Mukesh M Goswami
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology Faculty of Technology,
Dharmsinh Desai University College Road, Nadiad-387001
April 2021

DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF TECHNOLOGY DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD – 387001

April, 2020

CANDIDATE'S DECLARATION

We declare that final semester report entitled E-commerce book recommendation is our own work conducted under the supervision of the guide Prof.(Dr.) Mukesh M Goswami.

We further declare that to the best of our knowledge the report for B. Tech. final semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Also we declare that following students also worked in this project:

Yami Naik. - Department of IT, DD University
Yash Naik. – Department of IT, DD University

Yami Naik
Branch : IT Student ID: 17ITUON029

Yash Naik
Branch : IT Student ID: 17ITUOS092

DHARMSINH DESAI UNIVERSITY
NADIAD – 387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "E-commerce Book Recommendation" and is a bonafide report of work carried out by Miss. Yami Naik. ID No: 17ITUON147 and Mr. Yash Naik. ID No: 17ITUOS092 Of Department of Information Technology, semester VIII, under the guidance and supervision for the award of the degree of Bachelor of technology at Dharmsinh Desai University, Nadiad(Gujarat).They were involved in Project work during the academic year 2020 - 2021.

Yami Naik. - Department of IT, DD University

Yash Naik. – Department of IT, DD University

Prof. (Dr.)Mukesh M Goswami,
Associate Professor,
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad Date:

Prof. (Dr.) Vipul K. Dabhi,
Head ,Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad. Date:

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "E-commerce Book Recommendation" is a bonafied report of the work carried out by

1) Miss Yami Naik , Student ID No: 17ITUON029

2) Mr Yash Naik, Student ID No: 17ITUOS092

of Department of Information Technology, semester VIII, under the guidance and supervision for the award of the degree of Bachelor of Technology at Dharmsinh Desai University, Nadiad (Gujarat). They were involved in Project training during academic year 2017-2018.
Following student(s) was/were also involved in this project:

Yami Naik. - Department of IT, DD University

Yash Naik. – Department of IT, DD University

Prof.(Dr.) Mukesh M Goswami
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. (Dr.)V K Dabhi,
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to express our profound gratitude to our project guide, Prof.(Dr. Mukesh M Goswami, as well as our head of department, Prof. (Dr.) Vipul Dabhi, who gave us this resplendent opportunity to work on this project. Working on this project led us to research extensively in this field, which subsequently helped us grow our knowledge immensely. Furthermore, we also appreciate the assistance provided by numerous other faculty members of our department.

With Regards
Yami Naik
Yash Naik

TABLE OF CONTENTS

ABSTRACT

"BookEasy" is a E-commerce for book along with Book Recommendation System using Django framework. BookEasy is a recommender system built for book lovers. Using your profile, our system uses Machine Learning methods to provide you with highly personalized book recommendations. Recommender systems are at the forefront of the ways in which content-serving websites like Facebook, Amazon, Spotify, etc. interact with their users. Given this climate, it is paramount that websites aim to serve the best personalized content possible.

It helps to get book recommendation according to users shopping preference and able to buy book from our website and also able to write the review.We also have review sentiment analysis which can categorize positive netural and negative reviews.User can make online payment as we have Razorpay payement gateway integration.So here user can able to buy book according to his taste and check reviews before buying it.

1.0 INTRODUCTION

1.1 PROJECT DETAILS

This project is a collection of business logic and Machine Learning Algorithm and API which will provide the functionalities to the BookEasy-the book Recommendation System.This project provide user an ecommerce experience with book recommendation and rating analysis facility in it.

1.2 PURPOSE

In current times, most day to day tasks have been digitized in one form or the other. This application is meant to simplify the process of finding book of his choice.Moreover, it also helps customer find right book from others people reviews as we have sentiment analysis of reviews.We also provide full ecommerce functionality such as add to card the product and buying the product.

1.3 SCOPE

The backend of this application is developed in Django and also we have used artificial intelligence algorithm for book recommendation and sentiment analaysis which is trending nowadays.By this project we are saving time of book reader as they are getting suggestion of the books of their interest.

1.4 OBJECTIVE

We use a content recommender system to power our recommendations. Content-based filtering is a method of recommending items by the similarity of the said items. That is, if I like the first book of the Harry Potter, and if the second book is similar to the first, it can recommend me the second book.

After getting recommendation the user can buy the book as we have full ecommerce functionality and also user can analyse the reviews of others as we have use review sentiment analysis which categorize reviews into positive, negative and neutral.

1.5 TECHNOLOGY AND LITERATURE REVIEW

1.5.1 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Django was designed to help developers take applications from concept to completion as quickly as possible.Django takes security seriously and helps developers avoid many common security mistakes. Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

1.5.2 Python

Python is a widely used high-level programming language for general purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java.

The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

1.5.3 HTML / CSS / JavaScript

HTML stands for Hyper Text Markup Language. HTML describes the structure of Web pages using markup. HTML elements are the building blocks of HTML pages. HTML elements are represented by tags.
CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files
JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

2.0 PROJECT MANAGEMENT

2.1 FEASIBILITY STUDY

2.1.1 Technical feasibility

The technical feasibility is to understand if it is possible to complete the project with the current technologies. We are going to use Machine Learning method to provide book recommendations.We are going to use Content based filtering.The backend will be implemented using Django, its purpose is to provide recommendation on the based of above mention filtering with the help of dataset. The frontend should be also be completed using Django framework.

2.1.2 Time Schedule feasibility

Since, the requirement is very clear with all the required tools and business logic at our disposal, this project will be finished within the given time period with enough time left in our hands for testing before the final version release.

2.1.3 Operational feasibility

Creating applications which made using Django and AI algorithms had become very common these days. Thus, this project has a very high operational feasibility given the plethora of tools and technologies available that support this kind of software development.

2.1.4 Implementation feasibility

The challenges in the implementation of the project include creating a data set of books and also in gathering libraray for reviews sentiment analysis.

2.2 PROJECT PLANNING

2.2.1 Project Development Approach and Justification

We plan on taking a time based approach to deliver the product on time. To begin with, we started by creating a set of flexible deadlines by which we are supposed to finish certain tasks. Furthermore, these tasks contain several subdivisions with other deadlines as well. Thus, this system of PROJECT MANAGEMENT working will help us focus on individual tasks and finish the product on time.

AGILE Method:

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing crossfunctional teams. Agile methods or agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, selforganization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

2.2.2 Milestones and Deliverables

The tangible milestones includes the following –

- Software Installation and Understanding of Technology.
- Creating the actions for generating the appropriate response which will be according to the desire of the user.
- Integration and system testing which will have the great user experience to interact with the website.

2.2.3 Roles and Responsibility

The roles and responsibility for the development process is shown in the following table.

Name	Role					
	Analysis	Design	Coding	Testing	Documentation	Maintenance
Yami Naik	✓	✓	✓	✓	✓	✓
Yash Naik	✓	✓	✓	✓	✓	✓

Table 2.2.3 Roles and Responsibility

2.2.4 Project Scheduling

BookEasy	StartDate	EndDate	100%
BookEasy Development	20/12/2020	30/3/2021	100%
Requirement Gathering	20/12/2020	26/12/2020	100%
Designing	26/12/2020	16/1/2021	100%
Coding And Testing	16/1/2021	13/3/2021	100%
Changes in Current System	13/3/2021	30/3/2021	100%

3.0SYSTEM REQUIREMENTS STUDY

3.1 STUDY OF CURRENT SYSTEM

We have ecommerce website like amazon, flipkart etc. which have functionality of buying goods online and do delivery of the goods. They provide online payment facility and COD also. They also provide the cart facility in which we can put our goods to buy later and also they recommend the goods according to our behaviours .Also they provide rating and reviews section for product.

3.2 PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM

The current ecommerce does do reviews analsyis so they can hide the negative reviews or push down negative reviews in order to manipulate the customer mind to buy product.

3.3 USER CHARACTERISTICS

There are 2 types of users of this system.

Customer: The person who want to buy and recommendation.
Admin: The person who regulates the use of the application, checks user logs, approve payement etc.

3.4 HARDWARE AND SOFTWARE REQUIREMENTS

3.4.1 Hardware Requirements

Storage and processing power of these resources are dependent on the scaling of the application. The software can run on minimal hardware for minimal user traffic. However, with increase in the user traffic, the hardware will have to be scaled further.

3.4.2Software Requirements

- Front End: Web Browser
- Back End: Python, Django files
- Server: Action server
- IDE: Jupyter Notebook / Subline

3.5 CONSTRAINTS

3.5.1 Regulatory Policies

Some regulatory policies that should be followed by the developers include – abiding to certain coding standards, following best practices, logging various milestones and crashes, commenting on the code appropriately, using a version control system with its best practices.

3.5.2 Hardware Limitations

There are no specific hardware limitations, as long as the processing power and storage is available based on the application usage, the software is least likely to run into hardware issues.

3.5.3Interfaces to Other Applications

This application will have an integration with google assistant. This will enable the frontend users to use the application through voice commands directly from their Google Assistant application.

3.5.4Parallel Operations

Numerous parallel operations take place in the working of the software. When the client website is executed, it will go through the stories and will find out the intent which has high confidence among all the intents present. The the particular action is taken place and the appropriate response is shown to the user.

3.5.5Reliability Requirements

In order for the application to be reliable and accurate, hardware uptime is extremely important. If at any time the server is down, it may fail to collect the traffic data for the downtime. The application does demand much reliability and it is fully assured that the particular information requested the user will provide related information and flow is maintained.

3.5.6Criticality of the Consideration

The application deals with payment so the task and respective details should be in proper flow.

3.5.7 Safety and Security Consideration

Razorpay having inbuilt privacy policy there is no such problem and Django is very safe for privacy concern

3.6 ASSUMPTIONS AND DEPENDENCIES

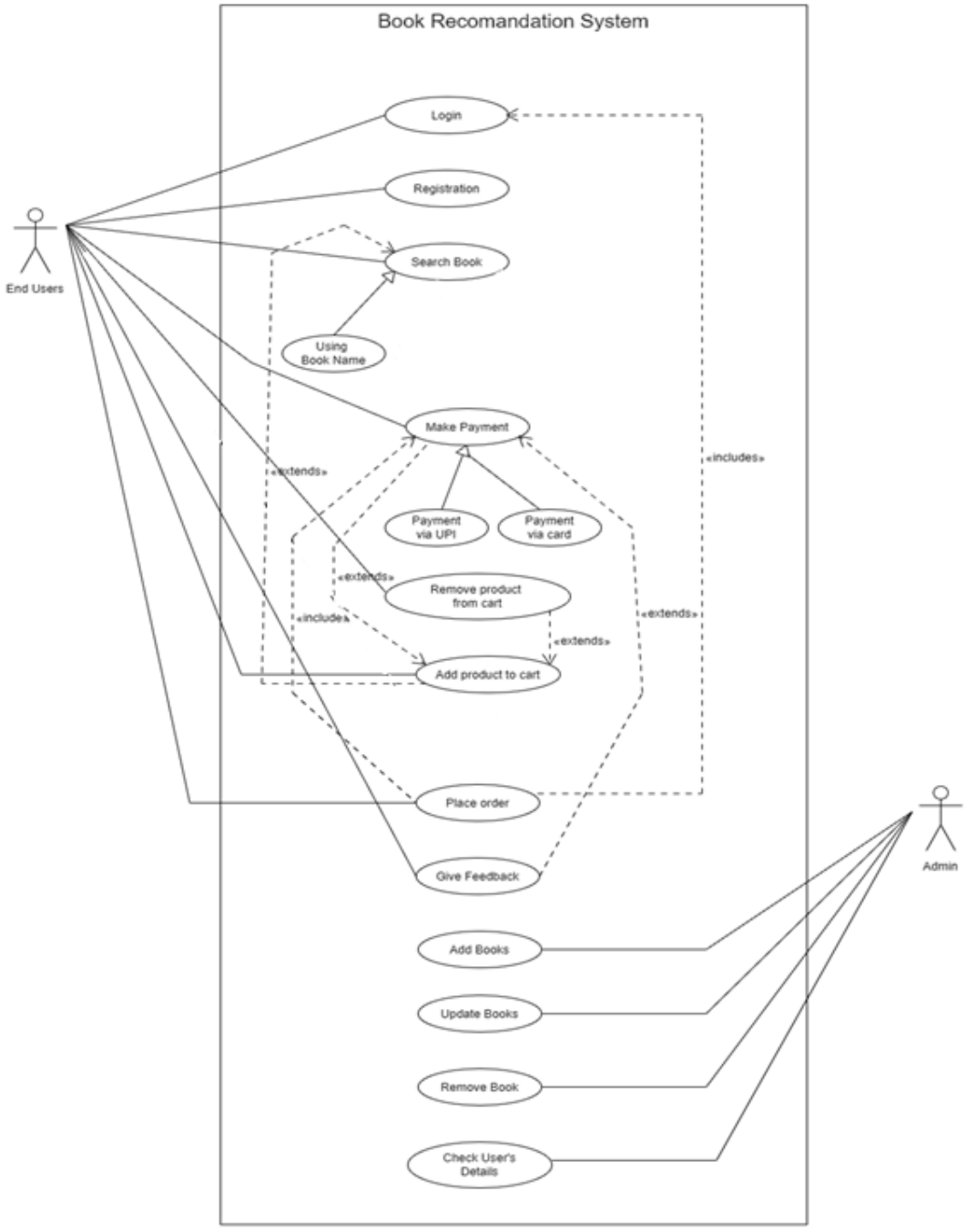
Some known assumptions and dependencies are mentioned below –

- The server works with perfect uptime.
- All the users must enter meaningful reviews to analyze sentiment

4.0 SYSTEM ANALYSIS

4.1 REQUIREMENTS OF NEW SYSTEM

4.1.1 Use-Case Diagram



4.1.2 System Requirements

R1: End user

R1.1: Create Account

Description:User can register for the account to get access to the E-commerce.
Input: Enter user details like email, password, name etc.
Output: Register successfully

R1.2: Login

Description:User can log in to the E-commerce.
Input: Enter login details
Output: Login successfully and redirect to home page

R1.3: Search book with book name.

Description:User can search for the book.
Input: Book name
Output: Book details of the given book

R1.4: Get suggested books on the basis of machine learning algorithm.

Description: Machine can automatic suggest book using different algorithms of machine learning.
Input: The Book Name
Output: All the suggested book

R1.5: Logout

Description: User can log out after finishing work.
Input: User selection.
Output: Redirection to Login page.

R1.6 Add to cart

Description:User add book to cart
Input:Click add to cart button
Output:Cart pop shows number of item added

R1.7 Remove from cart

Description:User can remove the product from the cart.
Input:Click remove from cart button.
Output:Cart pop item removed

R1.8 Add your reviews
Description: User add their reviews in textbox
Input: Write reviews in text box below books
Output: Review shows down the book

R1.9 Rate the book
Description : User give rating of books
Input:There are star to click for reviews.
Output:Rating shows down the book

R1.10 Check out
Description:User checkout to buy book
Input: Click on checkout option
Output:Shows the total amount of purchase and address form

R1.11 Payment option
Description: Payment gateway for buying book
Input:Click on pay option.
Output:Redirect to razorpay payment gateway

R1.12 Sentiment Analysis
Description:When User Reviews the book a sentiment analysis ids done which will classify the review as positive or negative.
Input:Write Done the Review
Output:It will be classified successfully as positive or negative

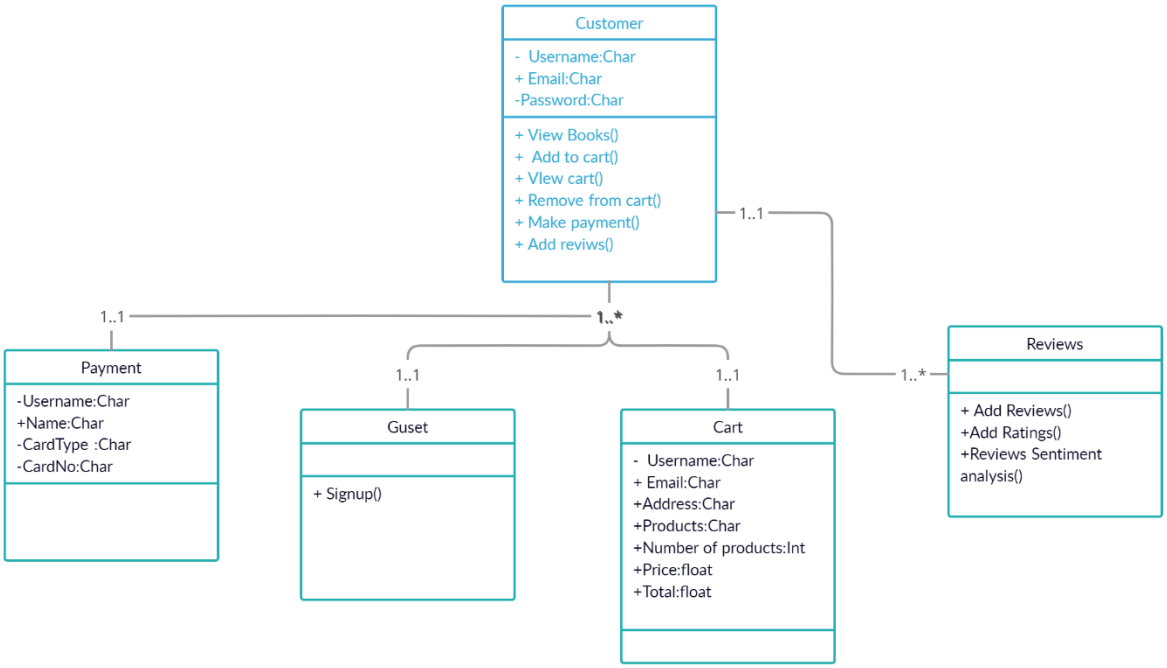
4.1.3 Non Functional Requirements

Security/privacy for all users Consistent and reliable data storage and management
Efficient and scalable web servers
Maintainability
Fast and efficient system

4.2 FEATURES OF NEW SYSTEM

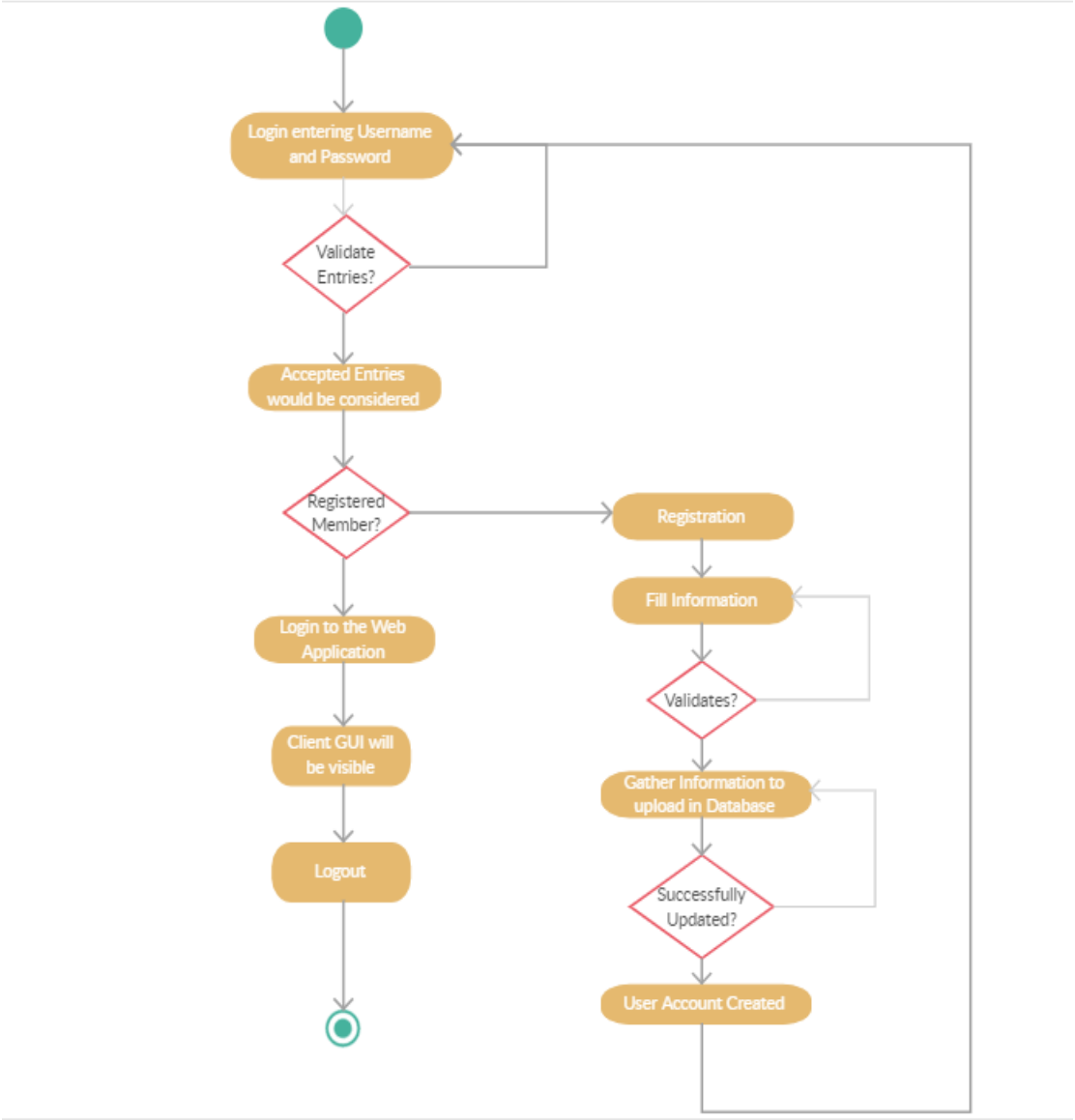
Find Books based on choice.

4.3 Class Diagram

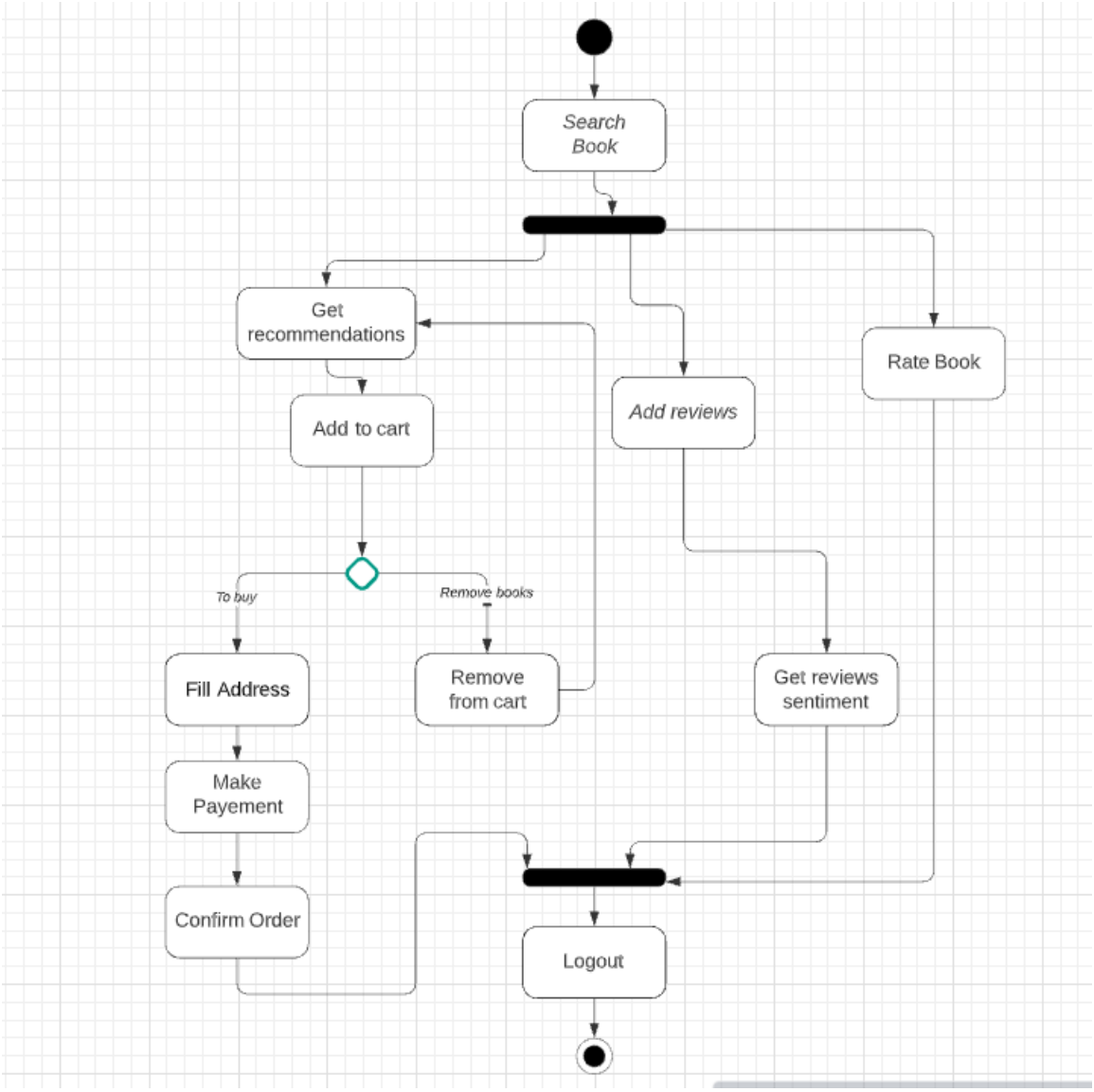


4.4 System Acticity

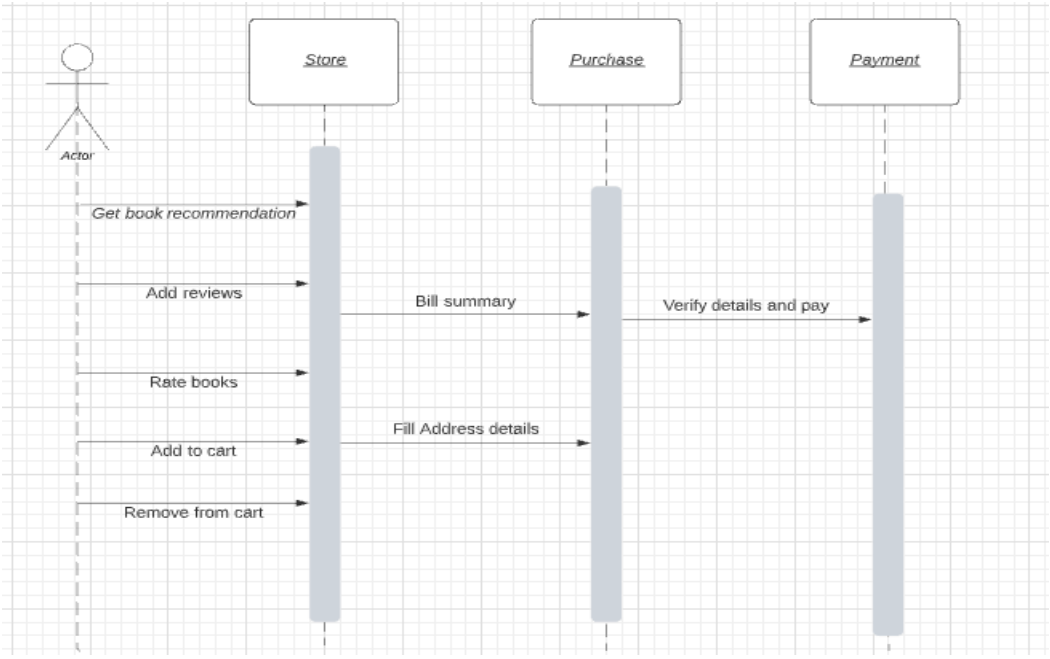
Activity Diagram for login and registration



Activity for purchasing book



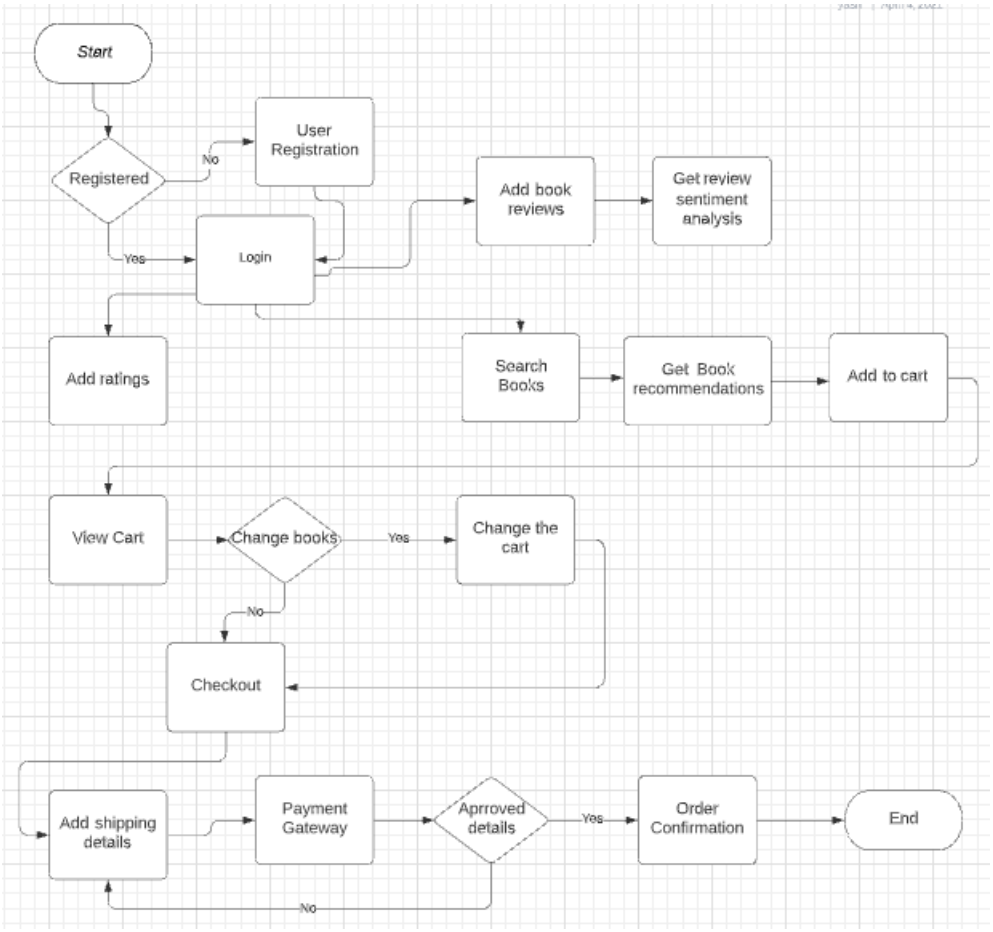
4.5 Sequence Diagram



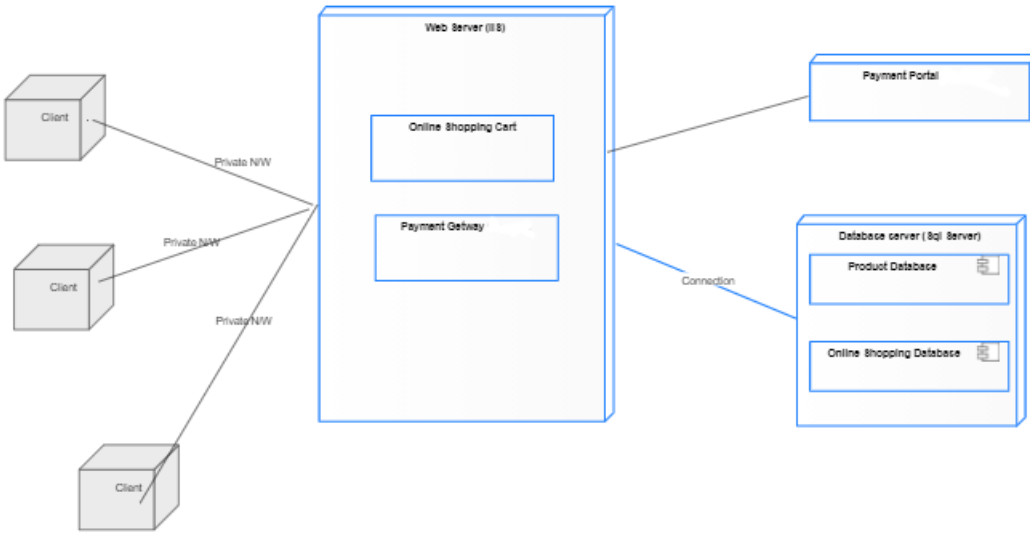
5.0 SYSTEM DESIGN

5.1 System Architecture Design

5.1.1 State Transition Diagram

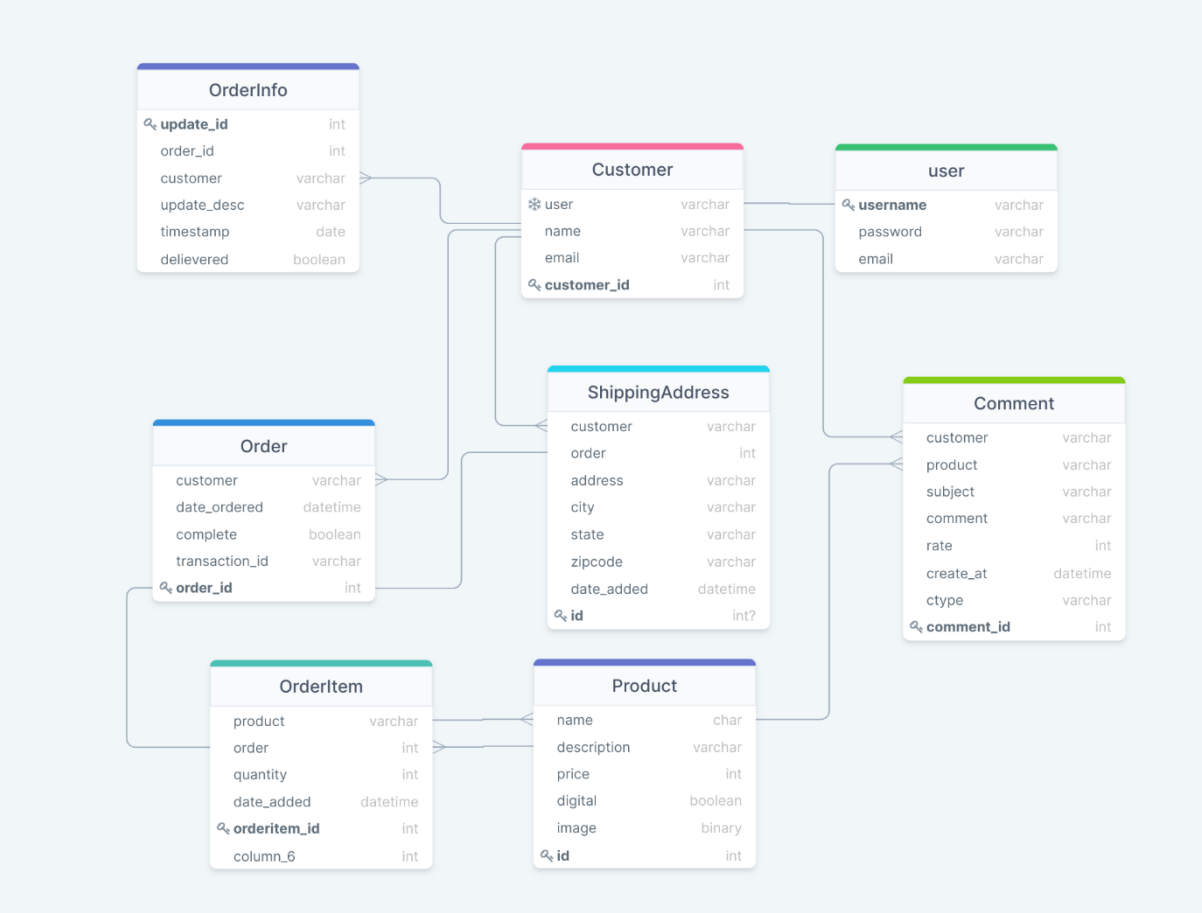


5.1.2 Deployment Diagram



5.2 Database Design

5.2.1 Schema Diagram



5.2.2 Data Dictionary

Customer

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	auto_increment
name	varchar(200)	YES		*NULL*	
email	varchar(200)	YES		*NULL*	
user_id	int(11)	YES	UNI	*NULL*	

Product

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	auto_increment
name	varchar(200)	YES		*NULL*	
price	int(11)	NO		*NULL*	
digital	tinyint(1)	YES		*NULL*	
image	varchar(100)	YES		*NULL*	
description	varchar(1000)	YES		*NULL*	

Order_item

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	auto_increment
quantity	int(11)	YES		*NULL*	
date_added	datetime(6)	NO		*NULL*	
order_id	int(11)	YES	MUL	*NULL*	
product_id	int(11)	YES	MUL	*NULL*	

Shipping_address

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	auto_increment
address	varchar(200)	NO		*NULL*	
city	varchar(200)	NO		*NULL*	
state	varchar(200)	NO		*NULL*	
zipcode	varchar(200)	NO		*NULL*	
date_added	datetime(6)	NO		*NULL*	
customer_id	int(11)	YES	MUL	*NULL*	
order_id	int(11)	YES	MUL	*NULL*	

Order_info

Field	Type	Null	Key	Default	Extra
update_id	int(11)	NO	PRI	*NULL*	auto_increment
order_id	int(11)	NO		*NULL*	
update_desc	varchar(5000)	NO		*NULL*	
timestamp	date	NO		*NULL*	
delievered	tinyint(1)	YES		*NULL*	
customer_id	int(11)	YES	MUL	*NULL*	

Order

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	auto_increment
date_ordered	datetime(6)	NO		*NULL*	
complete	tinyint(1)	YES		*NULL*	
transaction_id	varchar(100)	YES		*NULL*	
customer_id	int(11)	YES	MUL	*NULL*	

Comment

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	*NULL*	auto_increment
comment	varchar(200)	YES		*NULL*	
rate	int(11)	NO		*NULL*	
customer_id	int(11)	YES	MUL	*NULL*	
product_id	int(11)	YES	MUL	*NULL*	
create_at	datetime(6)	NO		*NULL*	
ctype	varchar(200)	YES		*NULL*	
subject	varchar(200)	YES		*NULL*	

6.0 IMPLEMENTATION PLANNING

6.1 Implementation Environment

For implementation, software like Visual Studio Code is used and Google Collabrator is used.

The Visual Studio IDE contains several features like git integration, ESlint, JavaScript debugger, integrated console, intellisense, etc. Git in VSCode can be used for version control and manage conflicts and version from the IDE GUI itself. Furthermore, intellisense and eslint helps the coder to write efficient and fast code.

The Google Collabrator is used to check the machine learning algorithm and then integrate them in Django.

6.2 Coding Standards

Coding Standards contribute to an improved comprehension of source code. Perhaps one of the most influential aids to understanding the logical flow of an application is how the various elements of the avoiding names that expose the underlying implementation. Naming Conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier etc. which can be helpful in understanding the code.

Reasons for using the coding standards are:

- Uniform Distribution
- Sound Understanding
- Encourages good programming skills

All code should be well commented. All procedures and functions should begin with a comment to explain what the function/procedure performs.

- Good and meaningful comments make code more maintainable.
- Do not write comments for every line of code and every variable defined.
- Write comments wherever required. But good readable code will require very less comments. If all the variables and methods names are meaningful, that would make the code very readable and will not need more comments.

6.3 Coding Conventions

- Use tab for indentation that contains 4 spaces.
- Never use spaces to indent nor mix spaces and tabs.
- Lines should not have trailing white spaces.
- Semicolons should be used everywhere even though JavaScript allows code without semicolons.
- As Django would give even an indentation error.

6.4 Example Code

6.4.1

In this filtering,the book is recommended on the basis of the genre of the book you have search for. Here we will make the use of the book description column as for each book we have to create vector of matrix.The book description (book_desc) contains sentence combining with word and my model cannot directly understand the language so we have used nlp concept-Natural language processing Tfidf whoch will help us to create document matrix.So now all the common English words are removed and its filled with blank.We have applied fit transform to it.It will convert into sparsematrix which have lot of zeros and non zeros values.Now applying sigmoid it will show how one book is related to another via description.

```
import pandas as pd
import numpy as np
df=pd.read_csv("book_data2.csv")
df.head()
req=df.drop(columns=['book_authors','book_edition','book_format'])
req.head()
req.info()
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
books_tfidf = TfidfVectorizer(stop_words='english')
req['book_desc'] = req['book_desc'].fillna('')
tvf_matrix = books_tfidf.fit_transform(req['book_desc'])
tvf_matrix
tvf_matrix.shape
from sklearn.metrics.pairwise import sigmoid_kernel
sig=sigmoid_kernel(tvf_matrix, tvf_matrix)
indices=pd.Series(req.index,index=req['book_title']).drop_duplicates()
indices

def give_rec(title):
    idx =indices[title]
    sig_scores =list(enumerate(sig[idx]))
    sig_scores =sorted(sig_scores, key=lambda x:x[1], reverse=True)
    sig_scores=sig_scores[1:6]
    book_indices= [i[0] for i in sig_scores]
    return req['book_title'].iloc[book_indices]
print(give_rec('The Book Thief'))
```

Output

```
959     Maus I: A Survivor's Tale: My Father Bleeds Hi...
225     The Hiding Place: The Triumphant True Story of...
931                                       The Winds of War
236                             The Silence of the Lambs
1902                                 Winter of the World
Name: book_title, dtype: object
```

6.4.2 Review Classifier

Using nltk sentiment analyzer we will classify the sentiment as positive,negative or neutral.

```
import string
from collections import Counter
import matplotlib.pyplot as plt
import nltk

import pandas as pd
nltk.download('wordnet')
nltk.download('vader_lexicon')
from nltk.corpus import stopwords
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
# reading text file
final_words = []
lemma_words=[]
emotion_list = []
text = request.POST.get('comment')
lower_case = text.lower()
cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))
tokenized_words = cleaned_text.split()
stop_words = ["i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours", "yourself",
              "yourselves", "he", "him", "his", "himself", "she", "her", "hers", "herself", "it", "its", "itself",
              "they", "them", "their", "theirs", "themselves", "what", "which", "who", "whom", "this", "that", "these",
              "those", "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had", "having", "do",
              "does", "did", "doing", "a", "an", "the", "and", "but", "if", "or", "because", "as", "until", "while",
              "of", "at", "by", "for", "with", "about", "against", "between", "into", "through", "during", "before",
              "after", "above", "below", "to", "from", "up", "down", "in", "out", "on", "off", "over", "under", "again",
              "further", "then", "once", "here", "there", "when", "where", "why", "how", "all", "any", "both", "each",
              "few", "more", "most", "other", "some", "such", "no", "nor", "not", "only", "own", "same", "so", "than",
              "too", "very", "s", "t", "can", "will", "just", "don", "should", "now"]
```

```
for word in tokenized_words:
    if word not in stop_words:
        final_words.append(word)

for word in final_words:
    word = WordNetLemmatizer().lemmatize(word)
    lemma_words.append(word)


with open('F://sem8//book_store//emotions.txt', 'r') as file:
    for line in file:
        clear_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()
        word, emotion = clear_line.split(':')

        if word in final_words:
            emotion_list.append(emotion)
print(emotion_list)
```

```
def sentiment_analyse(sentiment_text):
    score = SentimentIntensityAnalyzer().polarity_scores(sentiment_text)
    if request.method == 'POST':
        form = CommentForm(request.POST)
        if score['neg'] > score['pos']:
            type="Negative"
            if form.is_valid():
                Comment.objects.create(
                customer=customer,
                product=product,
                subject = form.cleaned_data['subject'],
                comment = form.cleaned_data['comment'],
                rate = form.cleaned_data['rate'],
                ctype=type,
                )
        elif score['neg'] < score['pos']:
            type="Positive"
            if form.is_valid():
                Comment.objects.create(
                customer=customer,
                product=product,
                subject = form.cleaned_data['subject'],
                comment = form.cleaned_data['comment'],
                rate = form.cleaned_data['rate'],
                ctype=type,
                )
        else:
            type="Neutral"
            if form.is_valid():
                Comment.objects.create(
                customer=customer,
                product=product,
                subject = form.cleaned_data['subject'],
                comment = form.cleaned_data['comment'],
                rate = form.cleaned_data['rate'],
                ctype=type,
                )
sentiment_analyse(cleaned_text)
```

6.4.3 WordCloud Generator

Using the keywords that come in the comment which are helpful for classification appears in word cloud.

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
final_comment_words = ''
emotion_comment_list = ''
stopwords = set(STOPWORDS)
for val in pcomments,ncomments:
    val = str(val)
    lower_case = val.lower()
    cleaned_text = lower_case.translate(str.maketrans('', '', string.punctuation))
    tokenized_words = cleaned_text.split()
    for word in tokenized_words:
        if word not in stopwords:
            final_comment_words+=word
        print(final_comment_words)
    with open('F://sem8//book_store//emotions.txt', 'r') as file:
        for line in file:
            clear_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()
            word, emotion = clear_line.split(':')

            if word in final_comment_words:
                emotion_comment_list+=emotion
    print(emotion_comment_list)

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(emotion_comment_list)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
wordcloud.to_file("F:/sem8/book_store/store/static/images/first_review.png")
```

6.4.4 Razorpay payment gateway integration

Razorpay integration for payment and then redirection to “success” page where you can track your order.

```
<script>
    var options = {
        "key": "rzp_test_a6SQyxPY7IfF2e", // Enter the Key ID generated from the Dashboard
        "amount": "{{order.get_cart_total}}"* 100, // Amount is in currency subunits. Default currency is INR. Hence, 50000 refers to 50000 paise
        "currency": "INR",
        "name": "Books",
        "description": "Test Transaction",
        "image": "https://example.com/your_logo",
        "order_id": "{{ payment.id }}", //This is a sample Order ID. Pass the `id` obtained in the response of Step 1

        "handler": function (response){
          if (typeof response.razorpay_payment_id == 'undefined' ||  response.razorpay_payment_id < 1) {
            redirect_url = '/remain';
          } else {
            redirect_url = '/success';
          }
          location.href = redirect_url

        },
        "prefill": {
            "name": "Yami Naik",
            "email": "yaminaik9172yahoo.com",
            "contact": "9978314804"
        },
        "notes": {
            "address": "Razorpay Corporate Office"
        },
        "theme": {
            "color": "#3399cc"
        }
    };
    var rzp1 = new Razorpay(options);
    rzp1.on('payment.failed', function (response){
            alert(response.error.code);
            alert(response.error.description);
            alert(response.error.source);
            alert(response.error.step);
            alert(response.error.reason);
            alert(response.error.metadata.order_id);
            alert(response.error.metadata.payment_id);
    });
    document.getElementById('rzp-button1').onclick = function(e){
        rzp1.open();
        e.preventDefault();
    }
</script>
```

7.0 TESTING

7.1 TESTING STRATEGY

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery to customer. Your goal is to design a series of test cases that have a high likelihood of finding errors. Software testing techniques provide systematic guidance for designing tests that (1) exercise the internal logic of software components, and (2) exercise the inputs and outputs domains of the program to uncover errors in program function, behaviour and performance.

During early stages of testing, a software engineer performs all tests. However, as the testing process progresses, testing specialists may become involved. Reviews and other activities can and do uncover errors, but they are not sufficient. Every time the program is executed, the customer tests it! Therefore, you have to execute the program before it gets to the customer with the specific intent of finding and removing all errors. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

Testing Objective:-

- Testing is a process of executing a program with the intention of finding an error.
- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- A successful test is one that uncover an as-yet undiscovered error.

7.1.1 Unit Testing

Unit testing is a software development process in which the smallest testable part of an application, called units, are individually scrutinized for proper operation. Unit testing is often automated but it can also be done manually. This testing mode is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision. Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developer to modify the source code without immediate concerns about how such changes might affect the functioning of the units or the program as a whole. Once of whole of the unitsn a program have been found to be working in the most efficient and error free manner possible, larger components of the program can be evaluated by means of integration testing. I tested each single part of the entire application. I tested each and every module individually.

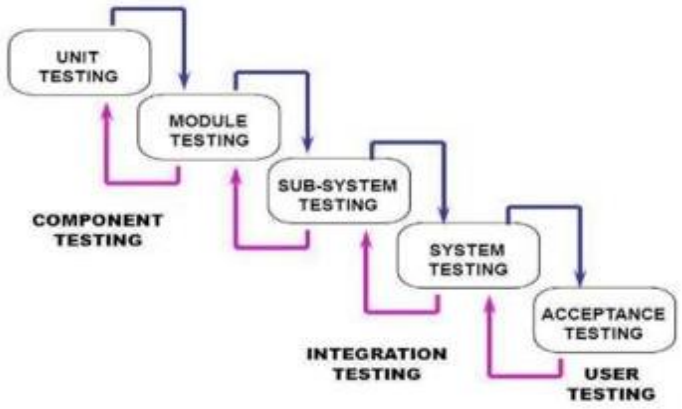


Fig. 7.1.1 Types of Testing

7.1.2 Sub System Testing

After testing each unit, we move on to larger units called sub system. In subsystem testing I tested the whole user side as one system. On the user side all the modules like dashboard, API, etc. were tested together to see if there was any error or bug found.

7.1.3 System Testing

After testing all the sub-system, it is time to test the whole system. System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.While testing the whole system I found many errors like the data mining delays leading to difficulties in inputs. I solved it by making appropriate changes in the duration of data mining as well as changed its properties. I worked on each error and exception that I got while testing and most of them are removed or made such correction that it will not happen again.

- **Recovery Testing**: It is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.
- **Security Testing**: It attempts to verify that protection mechanisms build into a system will, in fact, protect it from improper penetration.
- **Performance Testing:** It is designed to test the run-time performance of software within the context of an integrated system performance testing occurs throughout all step in the testing process.

7.1.4 Acceptance Testing

Acceptance testing can be connected by the end user, customer, or client to validate whether or not to accept the product. Acceptance testing may be performed as part of the hand-off process between any two phases of development. The acceptance test suite is run again the supplied input data or using an acceptance test script to direct the tester. Then the results obtained are compared with the expected results. If there is a correct match for every case, the test suite is said to pass.

7.2 TESTING METHODS

The verification activities fall into the category of static testing. During static testing, you have a checklist to check whether the work you are doing is going as per the set standards of the organization. These standards can be for coding, integrating and deployment. Reviews, Inspections and Walkthroughs are static testing methodology. Dynamic testing involves working with the software giving input values and checking if the output is as expected. These are the validation activities. Unit test, integration test, System and acceptance tests are few of the dynamic testing methodologies.

Alpha & beta testing: The alpha test is conducted at the developer's site by a customer. The software is used in a natural setting with the developer "looking over shoulder" of the user and recording errors and usage problems. Alpha test are conducted in a controlled environment. The beta testing is conducted at one or more customer site by the end-user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a "live" application of the software in an environment that cannot be controlled by the developer.

7.2.1 Black Box Testing

Also known as functional testing. A software testing techniques where by the internal working of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specification.

The advantages of this type of testing include:

- The test is unbiased as the designer and the tester are independent of each other
- The tester does not need knowledge of any specific programming languages
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

The disadvantages of this type of testing include:

- The test can be redundant if the software designer has already run a test case
- The test cases are difficult to design
- Testing every possible input stream is unrealistic because it would take an inordinate amount of time: hence many program paths will go untested.

7.2.2 White Box Testing

Also known as glass box, structural, clear box and open box testing. A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can than see if the program diverges from its intended goal.

7.2.3 Design of Test Cases

To minimize the number of errors in software, a rich variety of test design methods have evolved for software. These methods provide the developer with a systematic approach to testing. More important, methods provide a mechanism that can help to ensure the completeness of test and provide the highest likelihood for uncovering errors in software.

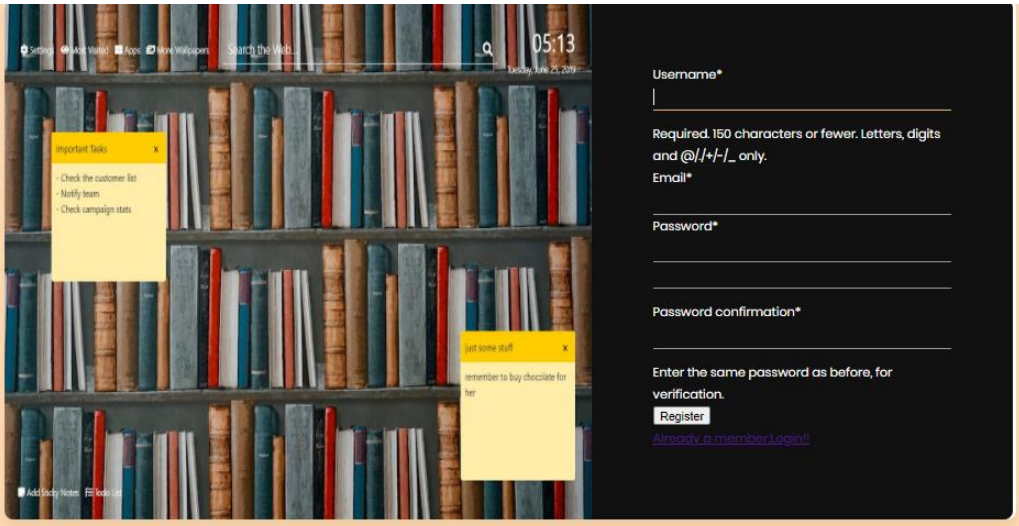
An engineering product can be tested in one of the two ways: (1) knowing the specified function that product has been designed to perform, tests can be conducted
demonstrate each function is fully operational while at the same time searching for errors in each function: (2) knowing the internal workings of a product, tests can be conducted to ensure that “all gear mesh”,that is,internal oppression are performed that according to specifications and all internal components have been adequately exercised. Here are the test cases that we had made for our application.

7.3 Test Cases

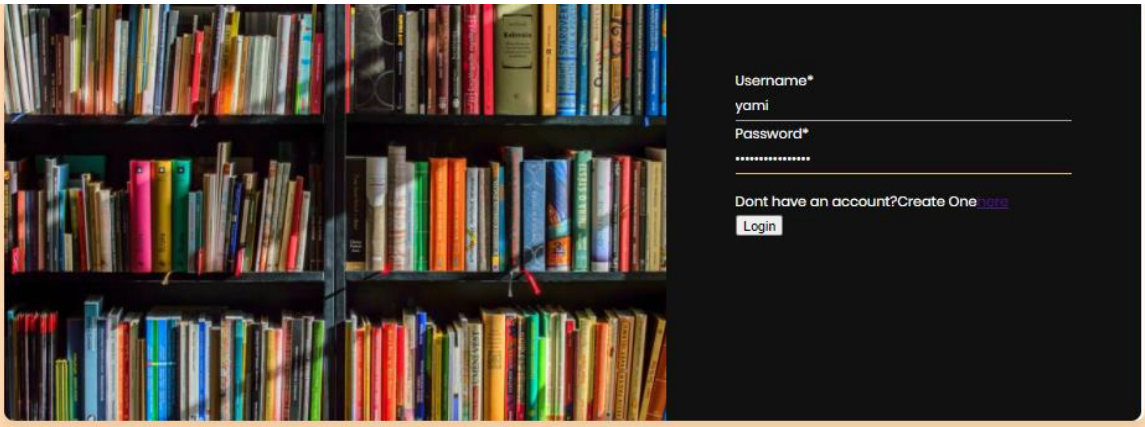
Sr.No	Module	Input	State	Ex. Output	Actual Output	Test Result
1	User	New User tries to Login	LoginPage	User doesn't exist	Invalid username or password	Pass
2	User	Valid Email,Password	LoginPage	Success	Success	Pass
3	User	Invalid Email,Password	LoginPage	User doesn't exist	Invalid username or password	Pass
4	User	When User tries to view the book.	HomePage	Book viewed successfully with details	Book viewed successfully with details	Pass
5	User	Search Book	HomePage	Search Book displayed	Search Book displayed	Pass
6	User	Book Recommendation	Product Page	Similar Books Recommended	Similar Books Recommended	Pass
7	User	Add to Cart	IndexPage	Book added Succesful	Book added Successful	Pass
8	User	Total Rs. of Books	CheckoutPage	Correct Total Displayed	Correct Total Displayed	Pass
9	User	Add Checkout Details	CheckoutPage	Add Details Successfully	Add Details Successfull	Pass
10	User	Payment	PaymentPage	Payment done Successful	Payment done Successful	Pass
11	User	Mail to your email regarding OrderId	PaymentPage	When Payment Successful,mail is send.	When Payment Successful,mail is send.	Pass
12	User	Track your Order	TrackPage	Order tracked successfully.	Order Tracked Successfully.	Pass
13	User	Review the Book	Product Page	Book Reviewed Successful	Book Reviewed Successful	Pass
14	User	Classify the Review on the book	Product Page	Review classified Successfully.	Review classified a Successfully.	Pass

8.0User Manual

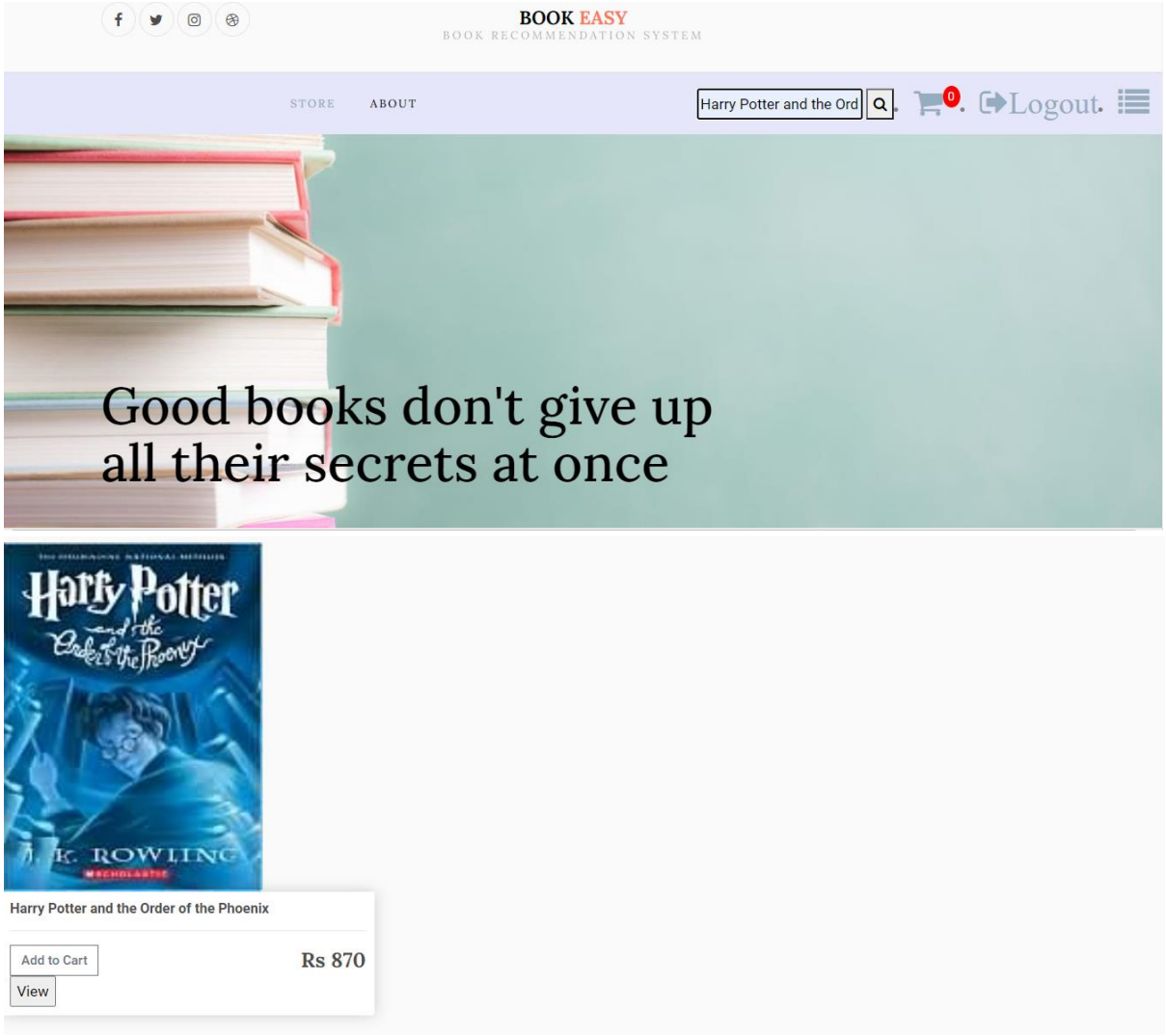
8.1. Signup



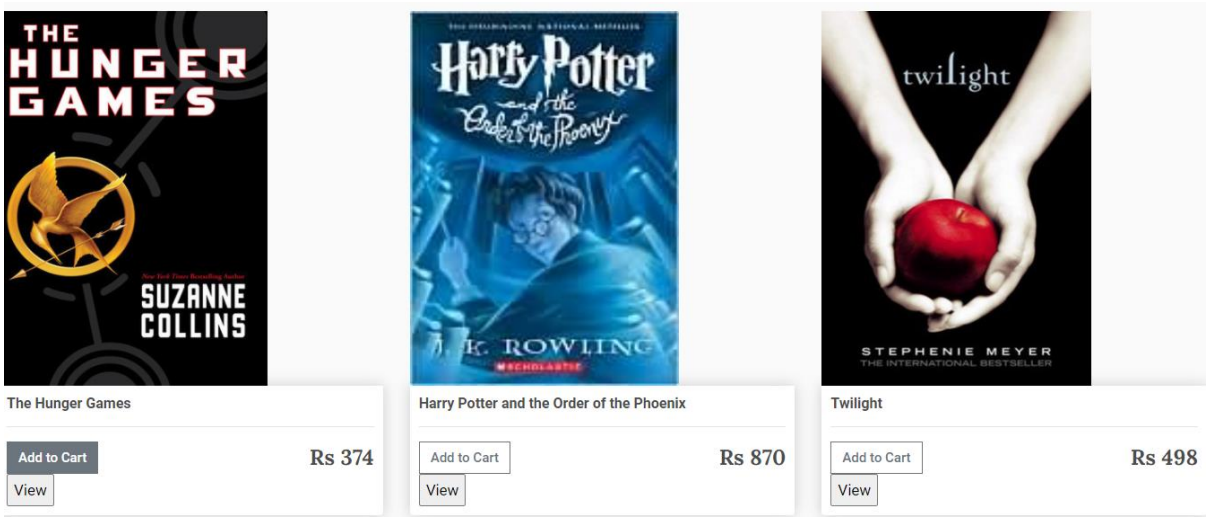
8.2 Login



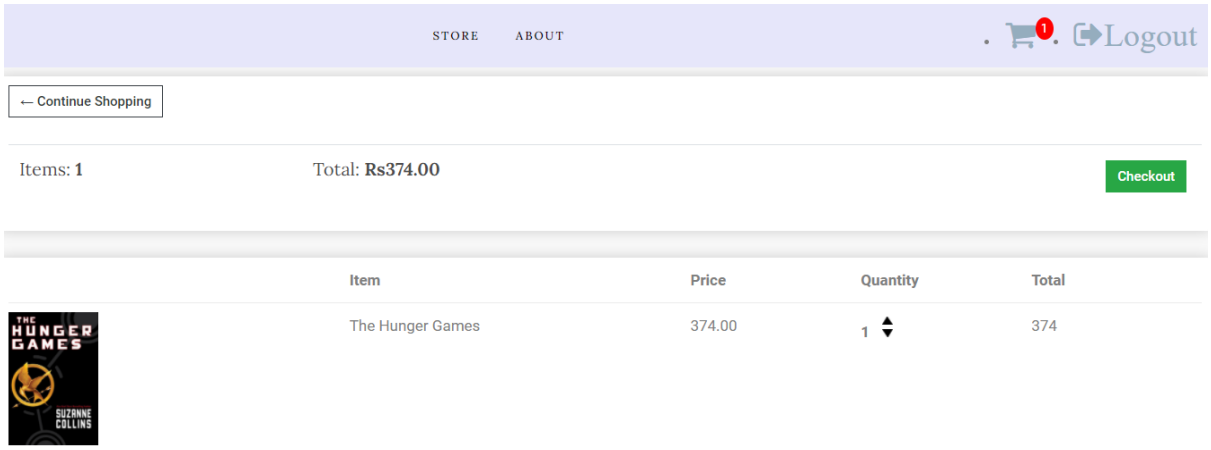
8.3 Search for the book



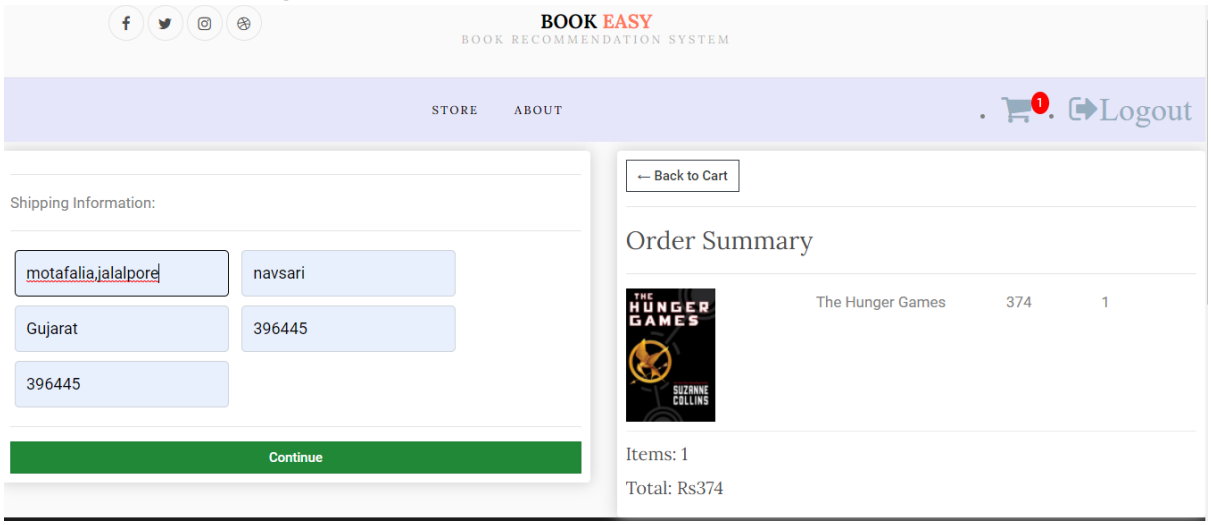
8.4 Add to cart



Order Overview Page

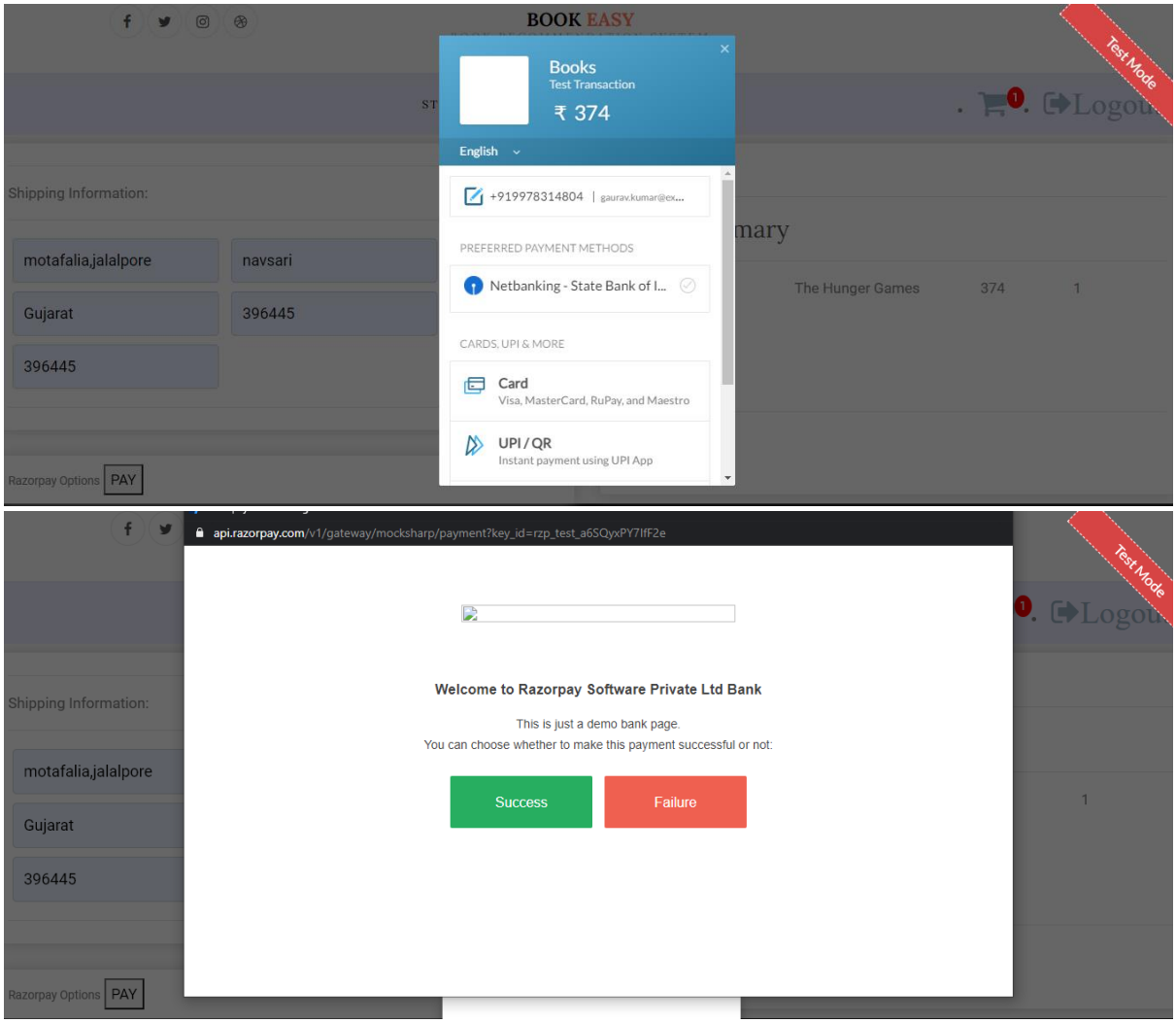


CheckOut Page



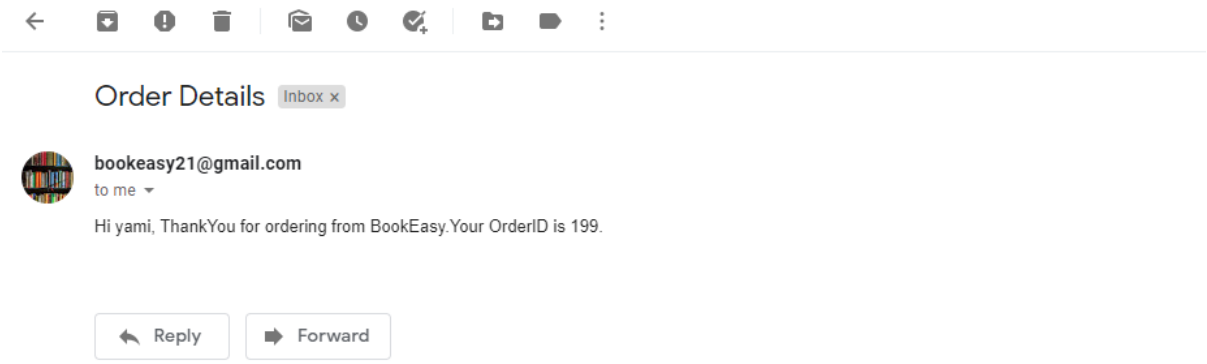
8.5 Payment Page

After successful page redirect to Tracking page.

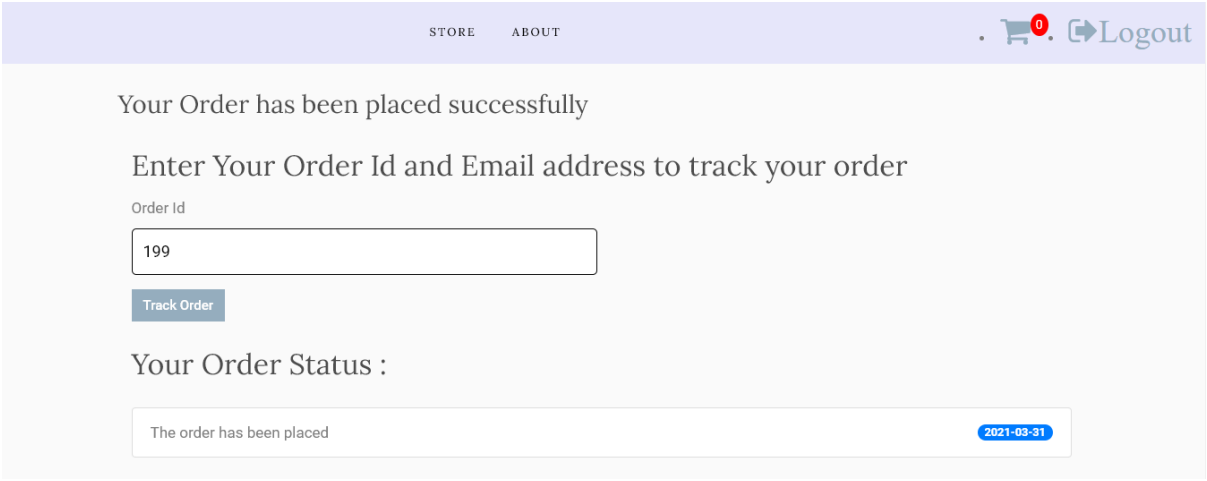


8.6 TrackOrder

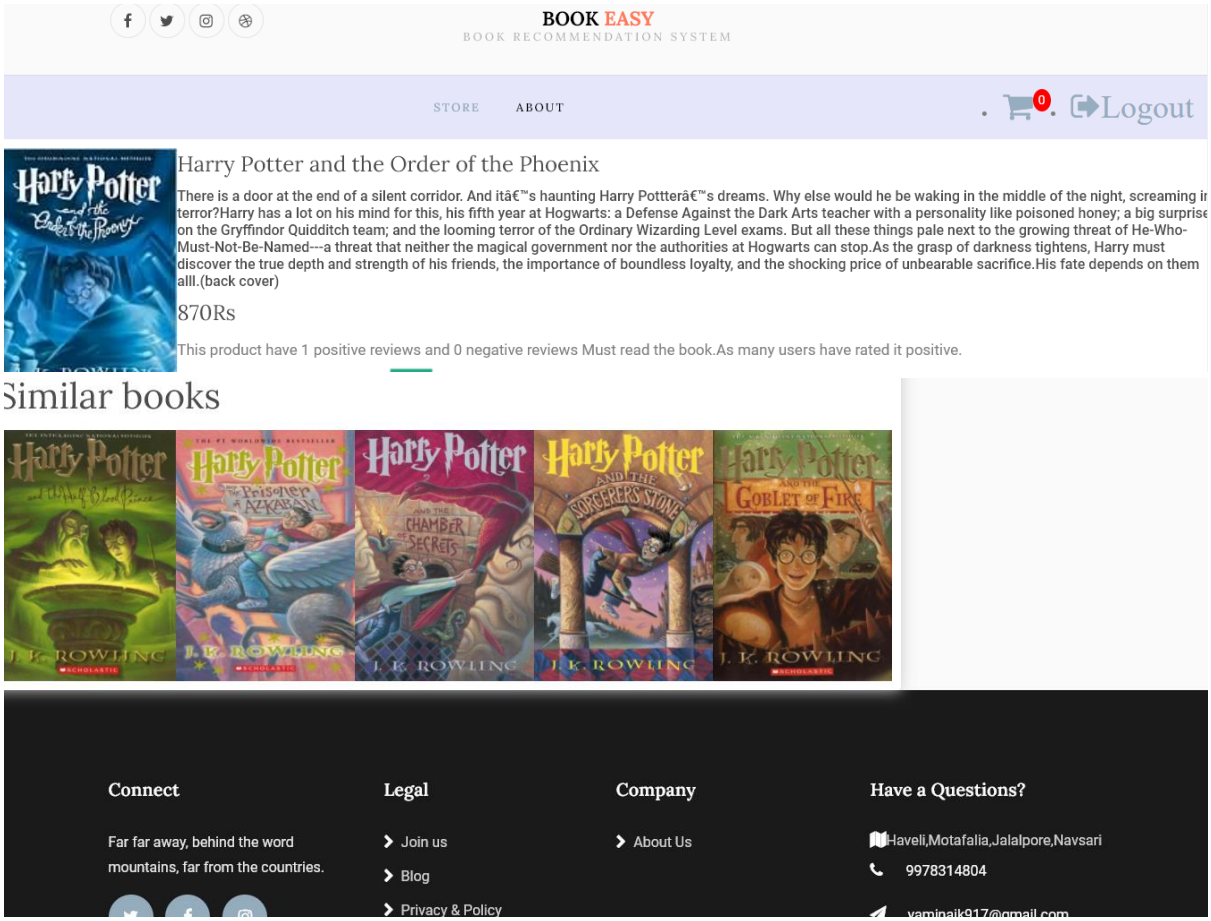
Email sent to user.



Track your order.

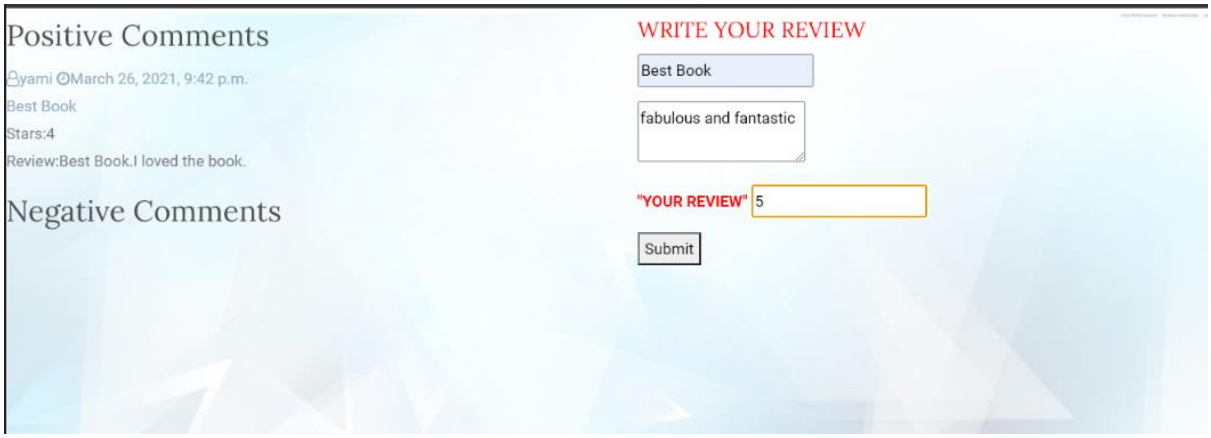


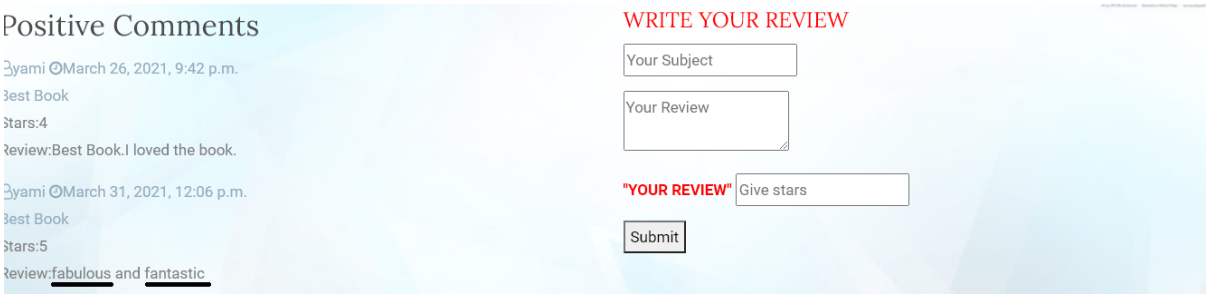
8.7 View Product Details and Show Similar Books.



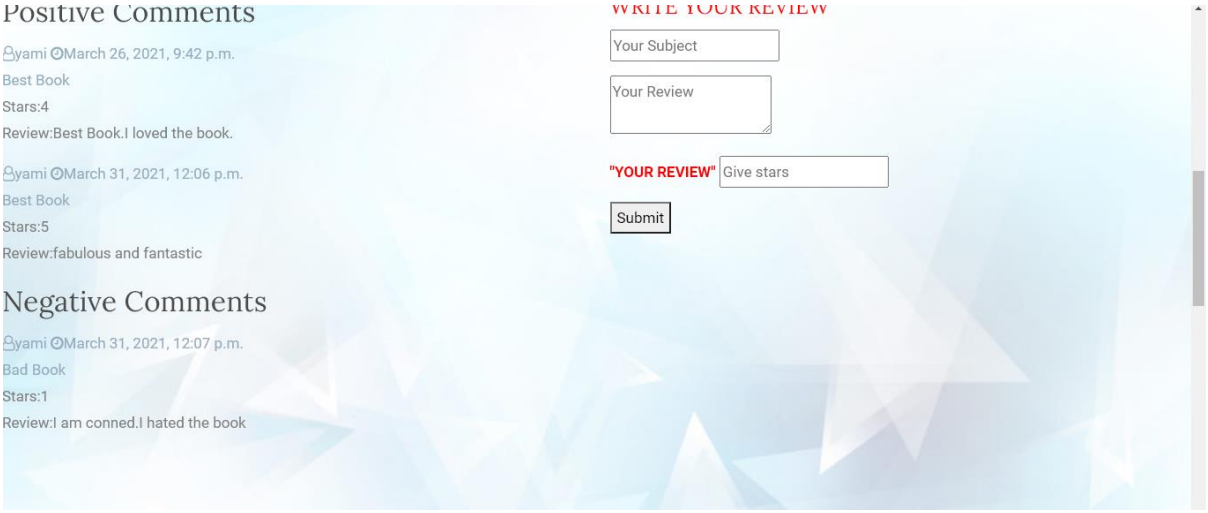
8.8 Review the Book,Classify the review and Create Word Cloud.

I have written “Fabulous and Fantastic” in the comment which says automatically by seeing that it’s a Positive Review.Both the words-fantastic and faboulous have same emotion-“ecstatic” so the word is added in wordcloud and the comment is classified as “Positive”.

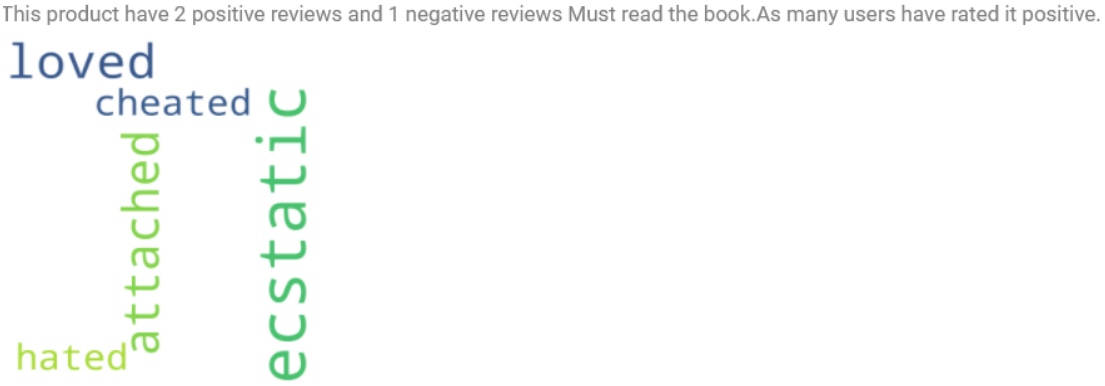




Similarly I have a Negative Comment which says “I am conned,I hated the book”. Which have emotion-“hated and cheated” so it will be classified as Negative Comment.



And the word cloud generated is:



9.0 LIMITATION AND FUTURE ENHANCEMENT

9.1 Limitations

The app is desktop app which constantly nedds to be managed.
The admin can add the books only from the backend.

9.2 Future Enhancement

Development and launching of Mobile app and refining existing services and adding more service, System security, data security and reliability are the main feature which can be done in future.

We can add features like, web app access, google assistant integration for more features, efficient use of queries, efficient database structure. Furthermore it can be made scalable to handle requests faster and more efficiently

10.0 Conclusion And Discussion

10.1 Conclusion

The functionalities implemented in system after understanding all the system modules according to the requirements.

Functionalities that are successfully implemented in the system are:

- Book recommendation
- E-Commerce Site

After the implementation and coding of system comprehensive testing was performed on the system to determine the errors and possible flaws in the system. We were able to train efficient models that had high accuracy. The model performed well on random inputs outside the dataset as well. We tried pursuing different approaches to solve the same problem.

10.2 DISCUSSION

10.2.1 Self-Analysis of Project Viabilities

In my opinion, the project has achieved the goal and requirements that we started with. Besides, there are a number of improvements and changes that we went through during the development of the application. Moreover, there is still a scope of a lot of improvements in the current system.

10.2.2 Problem Encountered and Possible Solution

During the development of this software, notable problems encountered include, programming the application with industry coding standards, figuring out how to integrate technologies that are designed and run on different platform from each other.

10.2.3 Summary of Project Work

To conclude, I have successfully complemented my proposed work for my industrial training. This was only possible with a proper schedule, planning, focus and guidance from the internal and external guides.

Reference/Biblography

Following links and websites were referred during the development of this project.

https://github.com/zygmuntz/goodbooks-10k
https://towardsdatascience.com
https://docs.djangoproject.com/en/3.0/
http://stackoverflow.com/
https://scikit-learn.org/stable/