

**SYNOPSIS**

**ON**

**“THE BOOK THINGS”**

Submitted in  
Partial Fulfillment of requirements for the Award of Degree  
*of*  
Bachelor of Technology  
*In*  
Computer Science and Engineering  
By  
**(Project Id: 2A09)**

**Anurag Singh Bhaduria (1901640100066)  
Gautam Yadav (1901640100121)**

Under the supervision of  
**Neha Singh**  
**(Assistant Professor)**



**Pranveer Singh Institute of Technology.**  
Kanpur - Agra - Delhi National Highway - 2  
Bhauti -Kanpur - 209305.  
(Dr. A.P.J. Abdul Kalam Technical University)

## **Table of Contents**

<u>Title-</u>	<u>Page Number-</u>
Introduction.....	3
Project Objective.....	4
Feasibility Study.....	5
Methodology/ Planning of work.....	8
Facilities required for proposed work.....	19
References.....	20

## **SYNOPSIS**

### **1. Introduction**

The business-to-consumer aspect of electronic commerce (e-commerce) is the most visible business use of the World Wide Web. The primary goal of an e-commerce site is to sell goods and services online. It is an E-commerce website for books through which one can get the desired books online. This is mainly developed to enhance the online book shopping experience of the users. The interactive interface and easy menu makes it easy for new users to surf the website. Not only this, one can also donate the books to the needy ones just by following a few simple steps on the website. The website functions to buy books easily from anywhere and let users donate books from anywhere just by following a few simple steps.

It provides the user with a catalogue of different books available for purchase in the store. The books that we show have been divided on the basis of genre. So, one can search the desired genre of their choice which makes it easier for the user to find the book quickly.

In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client side scripting techniques, implementation technologies such as Python, programming language (Python, Html, and CSS), and relational databases (such as Sqlite3).

The system is implemented using a 3-tier approach, with a backend database, a middle tier, and a web browser as the front end client.

## **2. Project Objective**

When users land on an e-commerce site, they expect to find what they are looking for quickly and easily. The main aim of the website is to make online book shopping easier and give users an easier way to donate the books to the needy ones. One of the main objectives behind creating this website is to donate the books to the needy users who want to read and have an opportunity to get the knowledge that is expensive for them to have.

In today's world people have everything but not have time to rush to the stores and buy things, most of the people prefer online shopping and it's actually helpful to many people. So, this is definitely time-saving. The books that we show will actually not be too expensive to buy, many people have a misconception that online shopping of books cost more than the other ways but we don't believe so and here we have feasible prices for the users to buy books. When you go to a store to buy any book, it is possible that you might have to wait in a queue for billing or might not find your desired book and come back disappointed but here we have abolished these kinds of things as online shopping has this advantage of fast purchasing and save time of the users.

Sometimes you might have to look for people whom you want to donate books which are no more relevant to you and the needy one might have to find you but we give our users a platform where they can directly donate the books by filling in their details.

The software keeps track of all information about the books and their complete details. It provides a User Interface for User and Admin login. The system contains the database where all the information will be stored safely. Specially designed for all types of book store related work.

### **3. Feasibility Study:**

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements.

#### **Different Type of Feasibility Study:-**

In the conduct of the feasibility study, the analyst will usually consider seven distinct, but inter-related types of feasibility. They are:

Technical Feasibility

Operational Feasibility

Economic Feasibility

Social Feasibility

Management Feasibility

Legal Feasibility

Time Feasibility

#### **(1) Technical Feasibility:-**

This is concerned with specifying equipment and software that will successfully satisfy the user requirement; the technical needs of the system may vary considerably, but might include:

The facility to produce outputs in a given time:

- Response time under certain conditions.
- Ability to process a certain volume of transaction at a Particular speed.
- Facility to communicate data to distant location.

In examining technical feasibility; configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirement: How many workstations are required, how these units are interconnected so that they could operate and communicate smoothly. What speeds of input and output should be achieved at particular quality of printing. This can be used as a basis for the tender document.

## **(2) Operational Feasibility:-**

It is mainly related to human organization and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organizational structures are distributed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational infallibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

## **(3) Economical Feasibility:-**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More frequently known as cost / benefit analysis; the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy a teach phase of the system life cycle.

**(4) Social Feasibility:-**

Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

**(5) Management Feasibility:-**

It is a determination of whether a proposed project will be project or gives a negligible support to it; the acceptable to management .if does not accept a project or gives a negligible support to it.

## **4. Methodology/ Planning of work**

Model view controller or MVC as it popularly called is a software design pattern for developing web application. A model view controller pattern is made up of the following three parts-

- **Model**- The lowest level of the pattern which is responsible for maintaining data.
- **View**- This is responsible for displaying all or a portion of the data to the user.
- **Controller**-Software code that controls the interactions between the model and view

The website has following modules:

1. Home
2. Contact
3. Login
4. Register
5. Feedback
6. Collectio

- **Home:**

This is the first page of the website where user gets the first sight of the website. This contains a navigation bar that leads to the other pages and it also shows what website is all about and makes an impact on the user.

- **Contact:**

This is the page where user can contact us regarding their queries and doubts. Here the user has to fill up a few details and then submit. Further the request will be received Email to the admin.

- **Login:**

This is the page where anyone who wants to use the website has to login like any other website with their credentials. If there is a new user then he or she will have to register first and that can be done through the register button given on the same page.

- **Register:**

This page is meant for the new users who want to use the website to make any purchase or donate a book. Without getting registered one cannot login and hence cannot make do any shopping except for surfing.

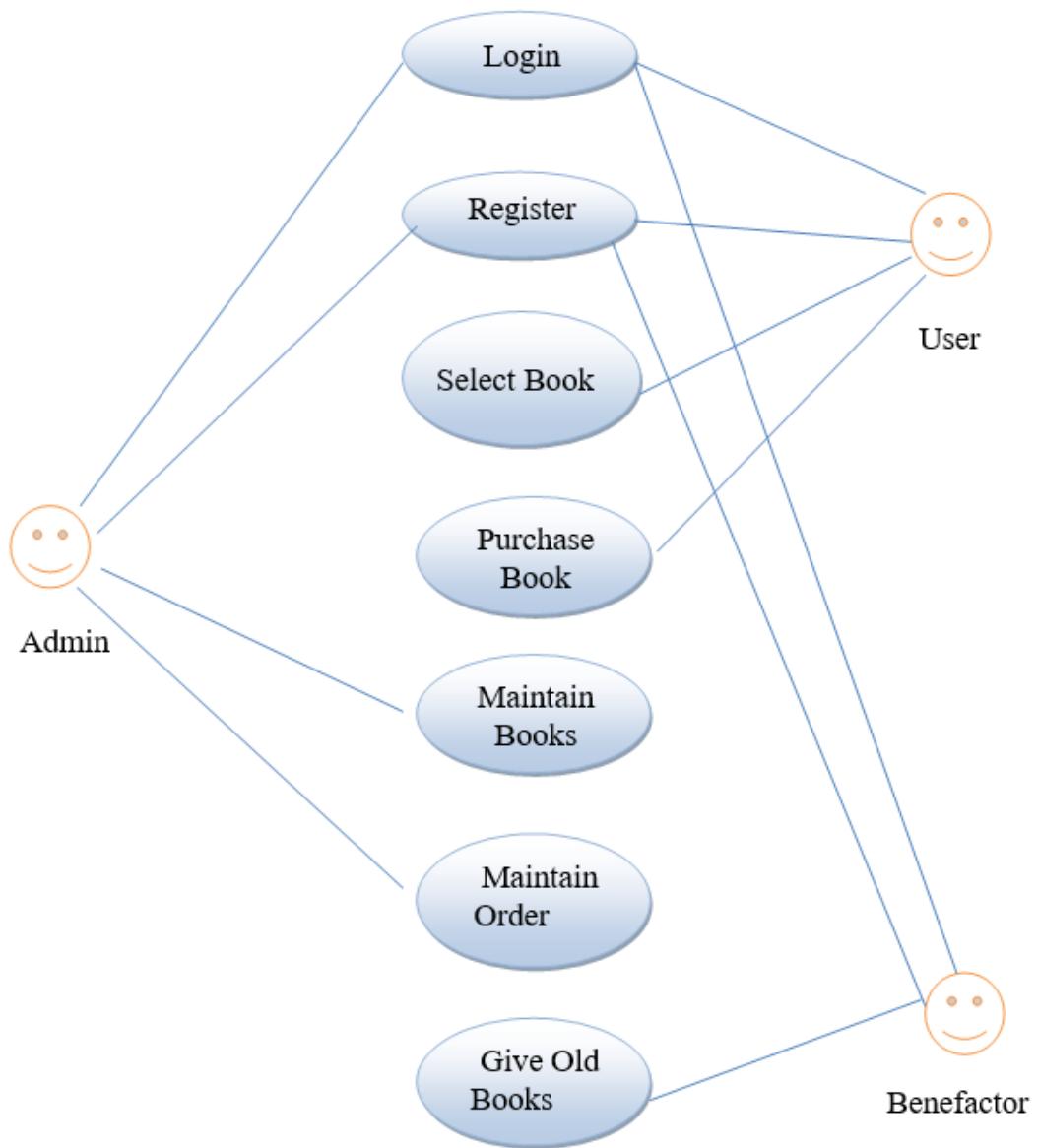
- **Feedback:**

This page is meant to improve our website to its full potential by the experience of users and by analyzing what can we improve more. This also lets user to tell the problems that they face while using the website and helpful to us to rectify it as soon as possible.

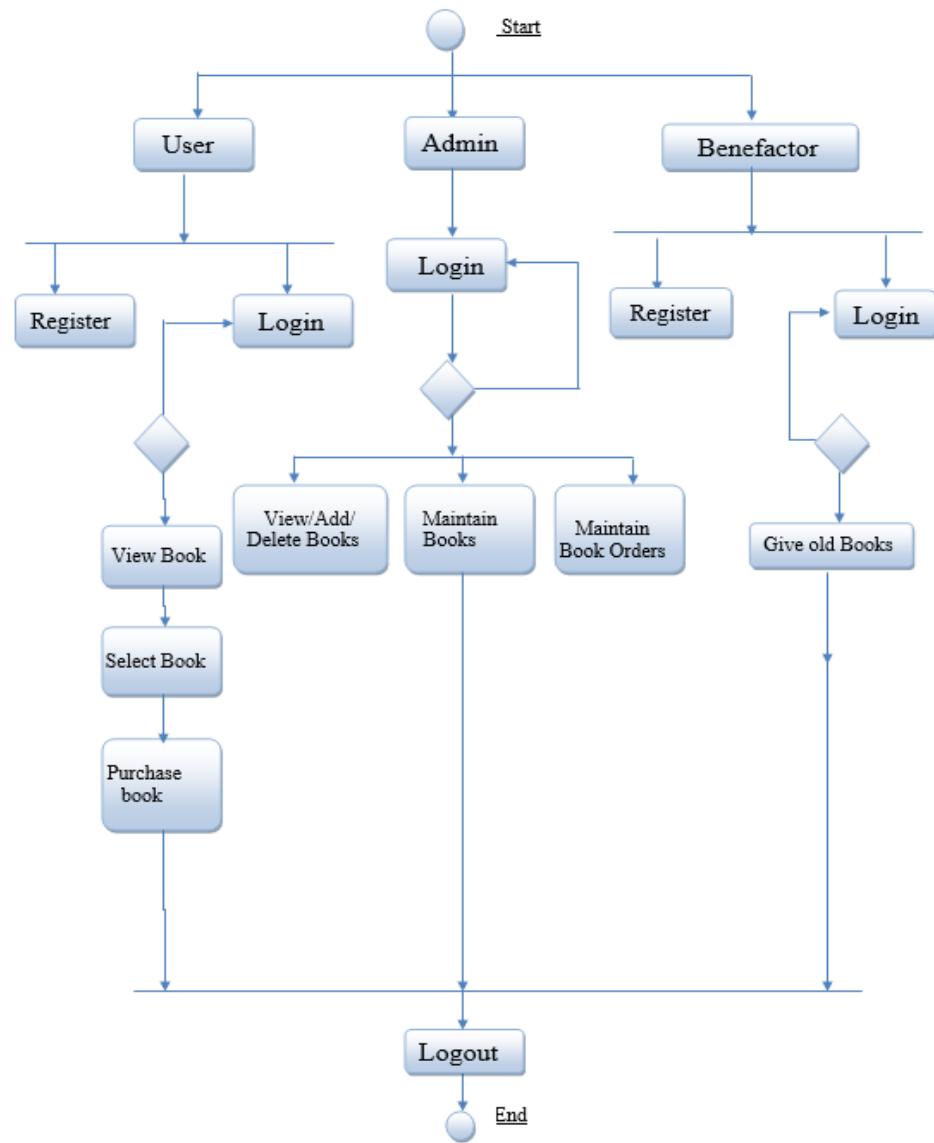
- **Collection:**

This is the page where we have the collections of all the books for the users. This page is a kind of showcase of the books that we have on our website and which redirects the interested buyer to purchase it and do further actions. This page can show books based on the genres you want. The website has many genres and you can search accordingly.

## Use Case Diagram:

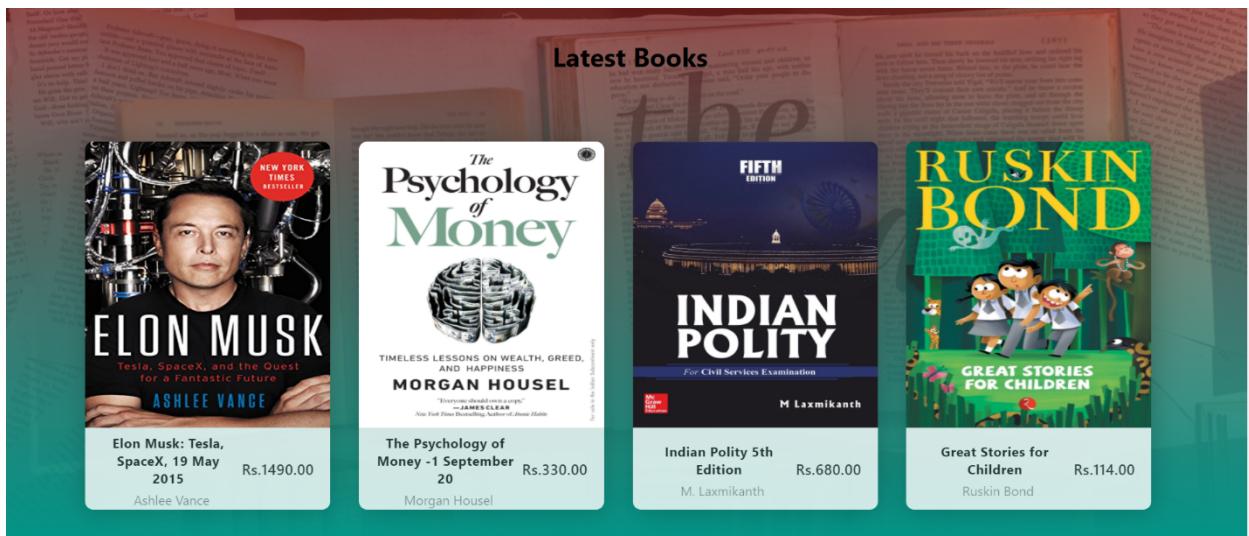
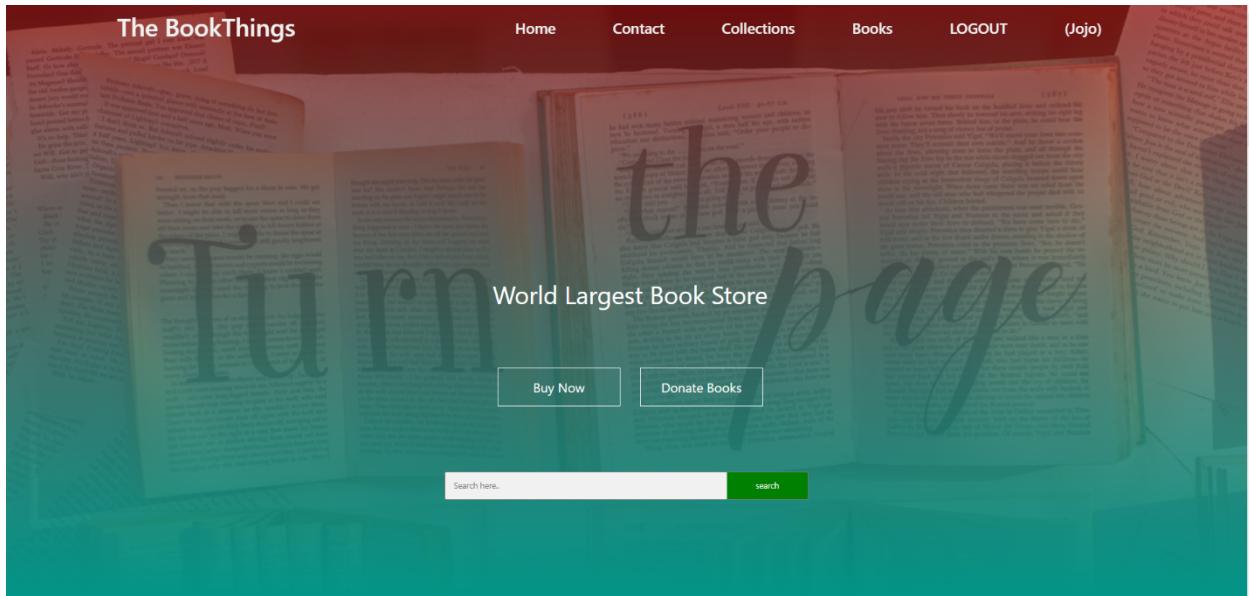


## Activity Diagram:



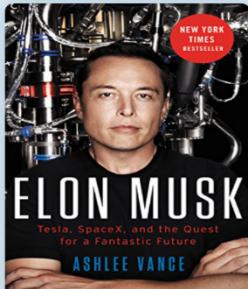
## Screenshots:

### Home Page:



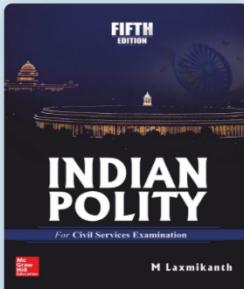
## Collection Page:

### BestSeller



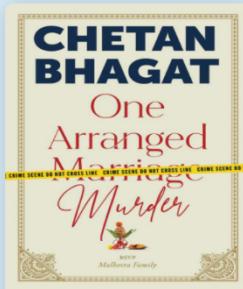
Elon Musk: Tesla,  
SpaceX, and the Quest  
for a Fantastic Future  
ASHLEE VANCE

Rs.1490.00



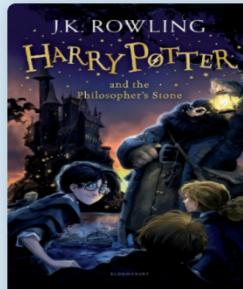
Indian Polity 5th  
Edition  
M. Laxmikanth

Rs.680.00



One Arranged  
Murder  
Chetan Bhagat

Rs.137.00



Harry Potter and the  
Philosopher's Stone  
J.K. Rowling

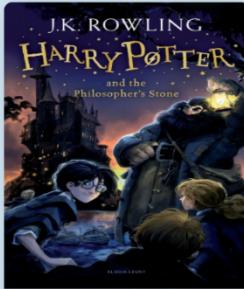
Rs.222.00

### Best for Children



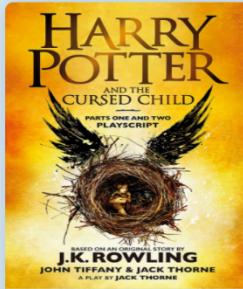
Great Stories for  
Children  
Ruskin Bond

Rs.114.00



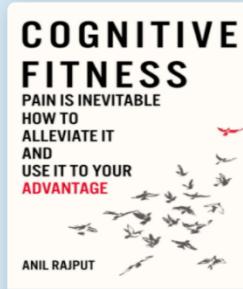
Harry Potter and the  
Philosopher's Stone  
J.K. Rowling

Rs.222.00



Harry Potter and the  
Cursed Child  
J.K. Rowling

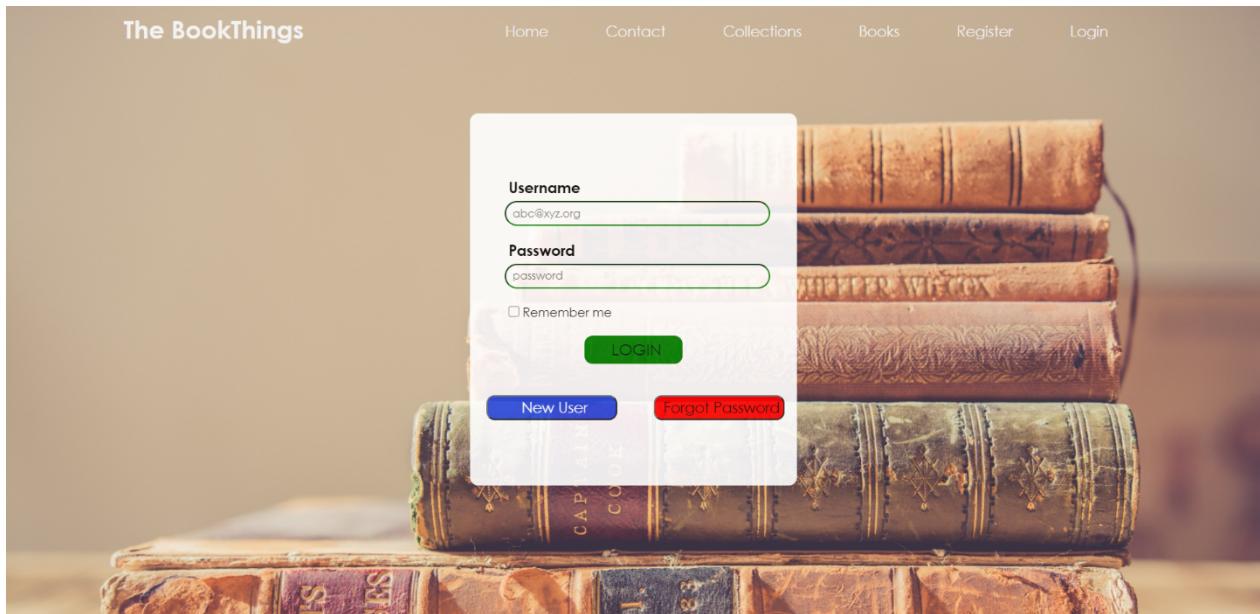
Rs.430.00



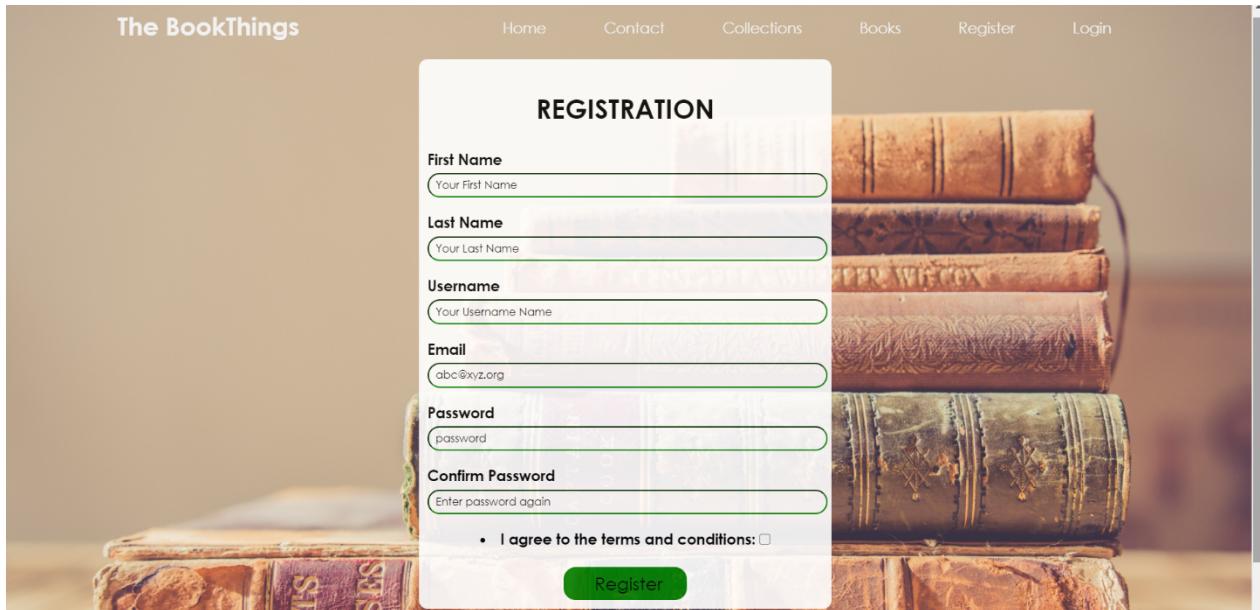
Cognitive Fitness:  
Pain is Inevitable  
Anil Rajput

Rs.309.00

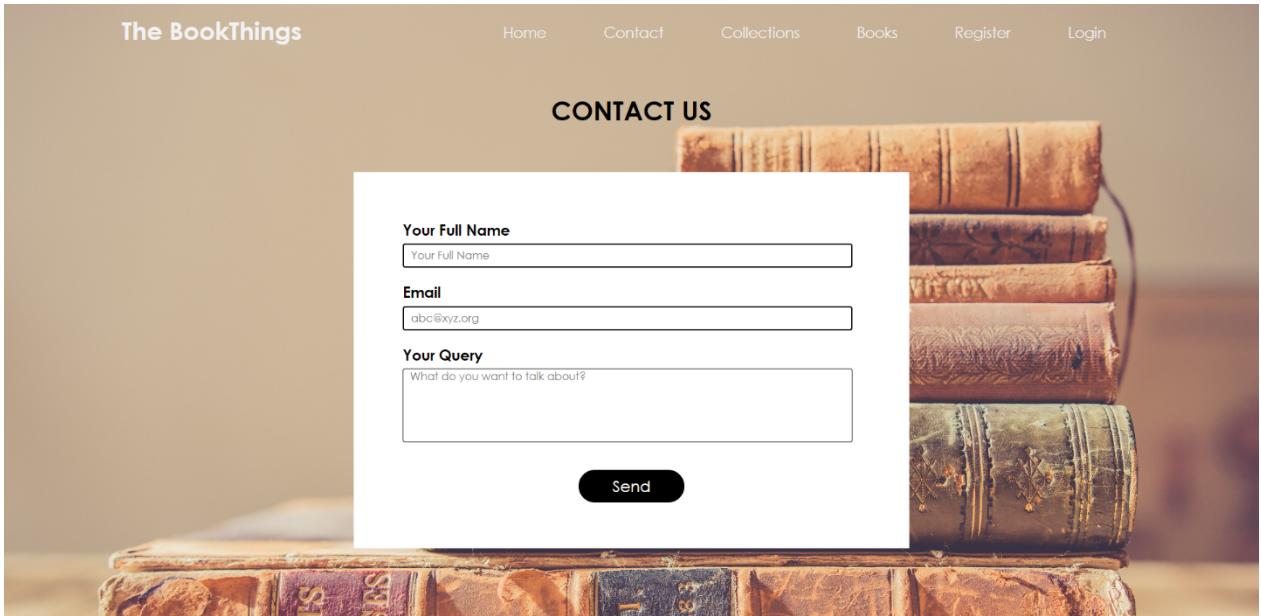
## Login Page:



## Register Page:



## Contact Page:



The BookThings

Home Contact Collections Books Register Login

### CONTACT US

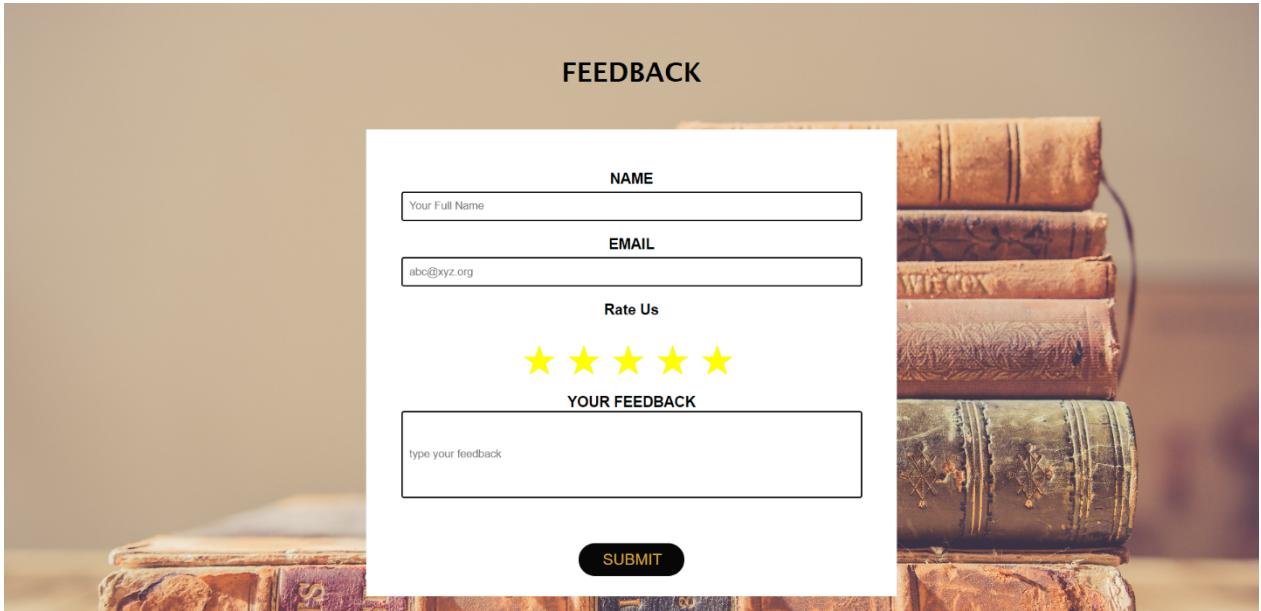
Your Full Name

Email

Your Query

**Send**

## Feedback Page:



### FEEDBACK

NAME

EMAIL

Rate Us  
★ ★ ★ ★ ★

YOUR FEEDBACK

**SUBMIT**

## Category Page:

The BookThings

Home Contact Collections Books Register Login

### Horror

The book cover features a dark, atmospheric illustration of a figure hanging from a tree branch against a cloudy sky. The title "POSSESSED & OTHER TALES FROM THE DARK" is at the top, followed by the author's name "AVISHEK KUMAR SINGH".

POSSESSED & Other Tales From The Dark  
AVISHEK KUMAR SINGH

Rs.250.00

The book cover shows a man in a striped pajama set standing in a dark hallway, looking down a long corridor. The title "THE SHINING" is prominently displayed in large letters.

The Shining  
Stephen King

Rs.339.00

The book cover features a photograph of a house with two red roses in the foreground. The title "THE HOUSE NEXT DOOR" is at the top, followed by the author's name "TOM COLEMAN".

The House Next Door: A Short Horror Story  
Tom Coleman

Rs.400.00

The BookThings

Home Contact Collections Books Register Login

### Cookbooks

The book cover displays various Indian spices and ingredients in small jars, with the title "MASALA LAB" and subtitle "THE SCIENCE OF INDIAN COOKING" below.

Masala Lab: The Science of Indian Cooking  
Krish Ashok

Rs.336.00

The book cover features a collection of small, round, white-coated food items, likely sweets or snacks. The title "THE FOOD BOOK" is visible.

Satvic Food Book : 45 Healing Recipes  
Subah Jain

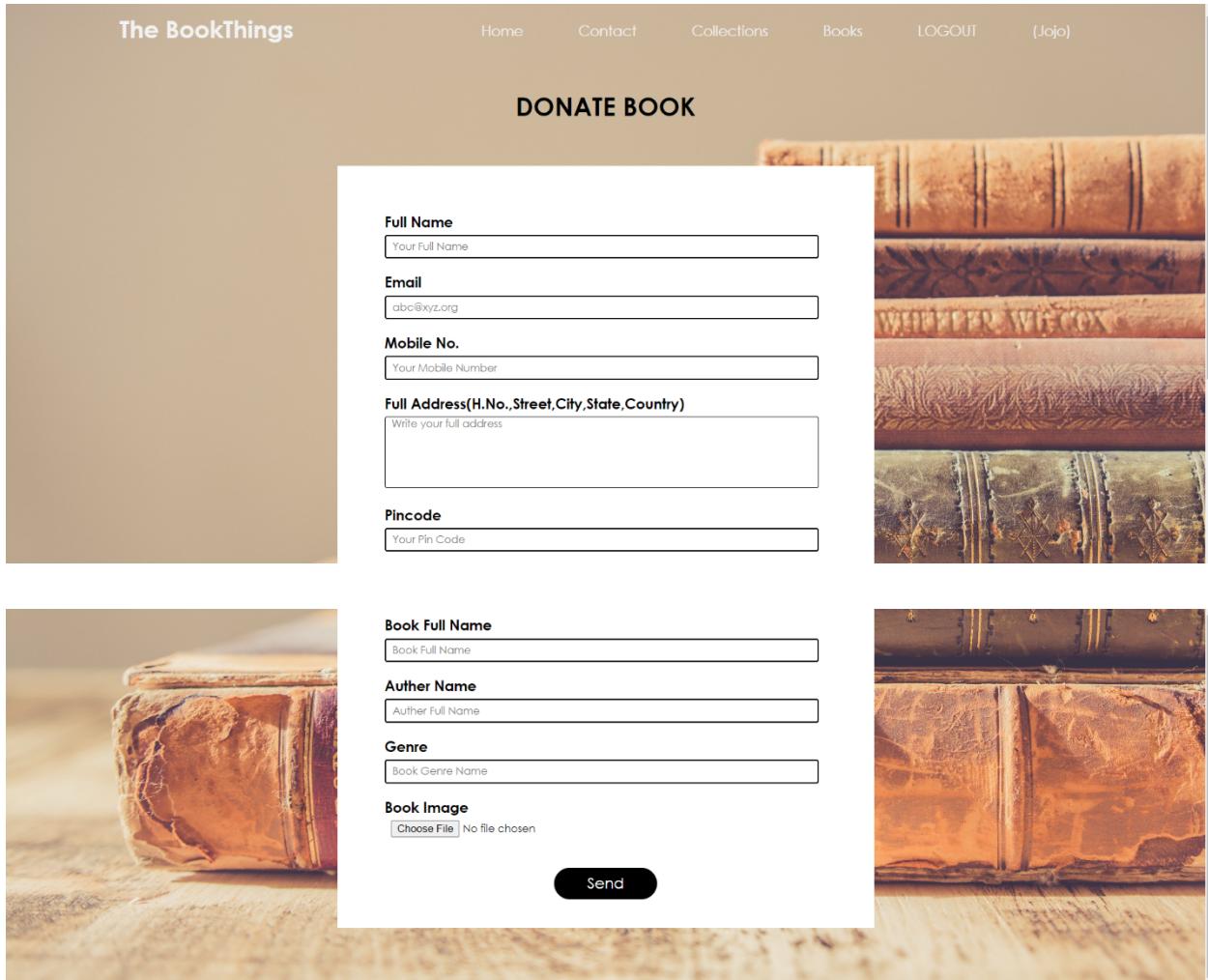
Rs.500.00

The book cover shows a hand holding a brass tiffin box filled with various Indian vegetarian dishes. The title "TIFFIN" is at the top, followed by the author's name "RUKMINI SRINIVAS".

Tiffin: Memories and Recipes of Indian Vegetarian Food  
Rukmini Srinivas

Rs.372.00

## Donation Page:

The background of the page features a large, slightly blurred photograph of several antique books standing upright. The spines of the books are visible, showing intricate gold-tooled decorations and some text embossed on them. The overall color palette is warm, with earthy tones and gold.

The BookThings

Home Contact Collections Books LOGOUT (Jojo)

### DONATE BOOK

**Full Name**

**Email**

**Mobile No.**

**Full Address(H.No.,Street,City,State,Country)**

**Pincode**

**Book Full Name**

**Auther Name**

**Genre**

**Book Image**

**Send**

## Single Book Page:

The BookThings

Home Contact Collections Books Register Login



Escape to Hogwarts with the unmissable series that has sparked a lifelong reading journey for children and families all over the world. The magic starts here. Harry Potter has never even heard of Hogwarts when the letters start dropping on the doormat at number four, Privet Drive. Addressed in green ink on yellowish parchment with a purple seal, they are swiftly confiscated by his grumpy aunt and uncle. Then, on Harry's eleventh birthday, a great beetle-eyed giant of a man called Rubeus Hagrid bursts in with some astonishing news: Harry Potter is a wizard, and he has a place at Hogwarts School of Witchcraft and Wizardry. The magic starts here!

### Give Rating To The Book

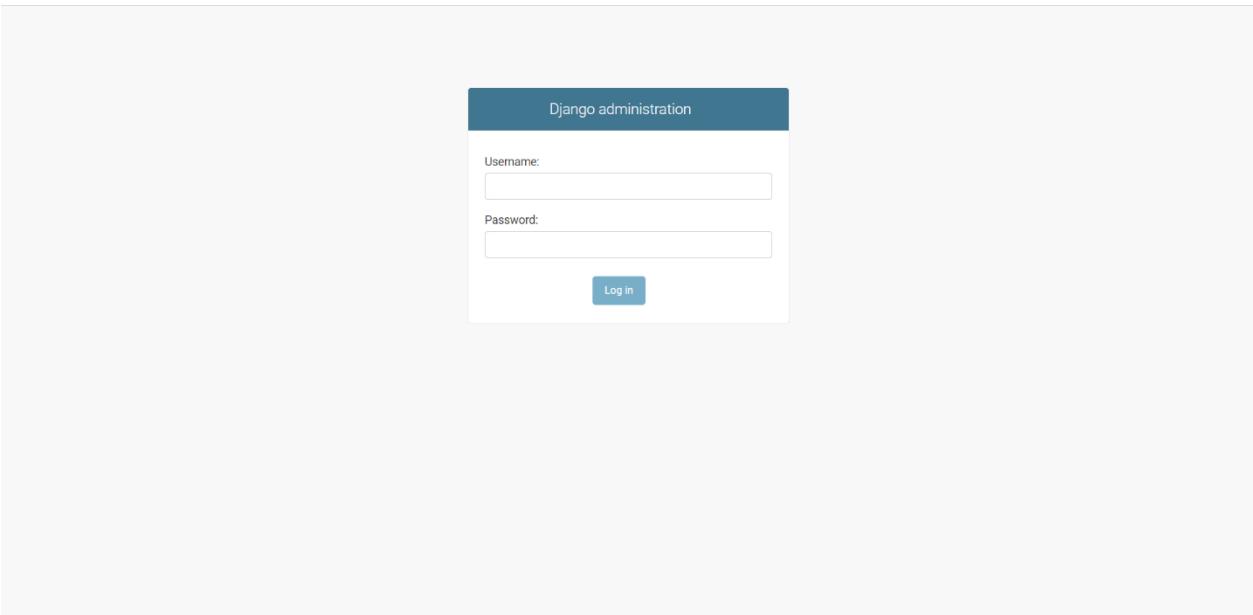
Rate Us

★ ★ ★ ★ ★

SUBMIT

## Database:

### Admin Login:



## Tables:

A screenshot of the Django administration site index. The top navigation bar shows "Django administration", "WELCOME, ADMIN", and links for "VIEW SITE / CHANGE PASSWORD / LOG OUT".

The main area is titled "Site administration". It features a sidebar with several categories: "AUTHENTICATION AND AUTHORIZATION" (Groups, Users), "COLLECTION" (Book\_detailss, Categorys), "CONTACT" (Contacts), "DONATEBOOK" (Book donations), "FEEDBACK" (Feedbacks), and "ONEVIEW" (Book\_ratings). Each category has "Add" and "Change" buttons.

On the right side, there are two sections: "Recent actions" and "My actions". "Recent actions" lists recent changes made to book details, such as "Elon Musk: Tesla, SpaceX, 19 May 2015" and "Indian Polity 5th Edition". "My actions" lists recent changes made to books, including "Harry Potter and the Cursed Child", "Queen of Mahishmathi (Baahubali)", and "POSSESSED & Other Tales From The Dark".

## Coding:

### Accounts: views.py

```
1  from django.shortcuts import render,redirect
2  from django.contrib import messages
3  from django.contrib.auth.models import User,auth
4  from collection.models import Category
5
6  # Create your views here.
7  def register(request):
8      cat=Category.objects.all()
9      l=len(cat)
10     cat1=Category.objects.all()[0:l//2]
11     cat2=Category.objects.all()[l//2:]
12
13     context2= {
14         'cat1': cat1,
15         'cat2': cat2
16     }
17
18     if request.method=='POST':
19         fname = request.POST['fnam']
20         lname = request.POST['lnam']
21         uname = request.POST['unam']
22         email = request.POST['email']
23         password = request.POST['pass']
24         cpassword = request.POST['cpass']
25
26
27
28     context2= {
29         'cat1': cat1,
30         'cat2': cat2
31     }
32
33     if request.method == 'POST':
34         username = request.POST['uname']
35         password = request.POST['pass']
36
37         user = auth.authenticate(username=username, password=password)
38
39         if user is not None:
40             auth.login(request, user)
41             return redirect('/')
42
43         else:
44             messages.error(request,'Invalid credentials')
45             return redirect('login')
46
47
48     else:
49         return render(request,'accounts/login.html',context2)
50
51
52     def logout(request):
53         auth.logout(request)
54         messages.success(request,'logout sucessfully')
55         return redirect('/')
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
```

```

26     if password==cpassword:
27         if User.objects.filter(username=username).exists():
28             messages.error(request,'username already exists')
29             return redirect('register')
30
31         elif User.objects.filter(email=email).exists():
32             messages.error(request,'email already exists')
33             return redirect('register')
34
35         else:
36             user = User.objects.create_user(username=username,email=email,password=password,first_name=fname,last_name=lname)
37             user.save()
38             messages.success(request,'user registered successfully')
39             return redirect('login')
40
41     else:
42         messages.error(request,'password do not match')
43         return render(request,'accounts/register.html')
44     else:
45         return render(request,'accounts/register.html',context2)
46
47 def login(request):
48     cat=Category.objects.all()
49     l=len(cat)
50     cat1=Category.objects.all()[0::2]
51     cat2=Category.objects.all()[1::2]

```

## Contact:

### views.py

```

1  from django.shortcuts import render,redirect
2  from django.contrib import messages
3  from django.core.mail import send_mail
4  from .models import Contact
5  from django.conf import settings
6  from collection.models import Category
7  from django.template.loader import get_template
8
9  # Create your views here.
10 def contact(request):
11     cat=Category.objects.all()
12     l=len(cat)
13     cat1=Category.objects.all()[0::2]
14     cat2=Category.objects.all()[1::2]
15
16     context2= {
17         'cat1': cat1,
18         'cat2': cat2
19     }
20
21
22     if request.method == 'POST':
23         name = request.POST['nam']
24         email = request.POST['email']
25         message = request.POST['issue']
26
27
28         contact = Contact(name=name,email=email,message=message)
29         contact.save()

```

```

33     context = {
34         'name': name,
35         'email': email,
36         'message': message
37     }
38     context1= {
39         'name': name
40     }
41
42     txt_ = get_template('contact/emails/contact-form.txt').render(context)
43     from_email = settings.DEFAULT_FROM_EMAIL
44
45
46     send_mail(
47         'Enquiry sent from TheBookThings',
48         txt_,
49         from_email,
50         ['gautamyadav1515@gmail.com'],
51         fail_silently=False
52     )
53
54
55     messages.success(request,"your request has been submitted, a realtor will get back to you")
56     return render(request,'contact/contact.html',context1)
57
58 else:
59     return render(request,'contact/contact.html',context2)

```

## models.py

```

1  from django.db import models
2
3  # Create your models here.
4  class Contact(models.Model):
5      name=models.CharField(max_length=50)
6      email=models.CharField(max_length=50)
7      message=models.TextField()
8
9      def __str__(self):
10          return self.name

```

## Home:

### views.py

```

27     def catfun(request,cate):
28         info=Category.objects.get(catname=cate)
29         infohorror=info.book_details_set.all()
30         cat=Category.objects.all()
31         l=len(cat)
32         cat1=Category.objects.all()[1//2]
33         cat2=Category.objects.all()[1//2:]
34
35         context={
36             'infohorror': infohorror,
37             'cat1': cat1,
38             'cat2': cat2,
39             'categ': cate
40         }
41
42         return render(request,'categories/horror.html',context)

```

```

1  from django.shortcuts import render
2  # from django.http import HttpResponseRedirect
3  # Create your views here.
4  from collection.models import Book_details
5  from collection.models import Category
6
7
8  def home(request):
9      ... #return HttpResponseRedirect("hello...my website")
10     ... info=Book_details.objects.all()[:8]
11     ... cat=Category.objects.all()
12     ... l=len(cat)
13     ... cat1=Category.objects.all()[l//2]
14     ... cat2=Category.objects.all()[l//2+1]
15     ... # print(cat1)
16     ... # print(cat2)
17 ...
18
19     ... context={
20         ... 'info': info,
21         ... 'cat': cat,
22         ... 'cat1': cat1,
23         ... 'cat2': cat2
24     }
25     ... return render(request,'home/home.html',context)

```

## Feedback:

### views.py

```

6  def feedback(request):
7      ... if request.method == 'POST':
8          ... name = request.POST['name']
9          ... email = request.POST['email']
10         ... rating = request.POST['rating']
11         ... feedb = request.POST['fb']
12
13         ... feedback = Feedback(name=name,email=email,feedback=feedb,rating=rating)
14         ... feedback.save()
15
16         ... messages.success(request,"your feedback has been submitted.")
17         ... return render(request,'home/home.html')
18
19     ... else:
20         ... return render(request,'feedback/feedback.html')

```

### models.py

```

1  from django.db import models
2
3  # Create your models here.
4  class Feedback(models.Model):
5      ... name=models.CharField(max_length=50)
6      ... email=models.CharField(max_length=50)
7      ... feedback=models.TextField()
8      ... rating=models.IntegerField(default=0)
9
10     ... def __str__(self):
11         ... return self.name

```

## Donation:

### views.py

```
1  from django.shortcuts import render
2  from .models import BookDonation
3
4  # Create your views here.
5  def bookdonate(request):
6      if request.method == 'POST':
7          name = request.POST['name']
8          email = request.POST['email']
9          mobile = request.POST['num']
10         address = request.POST['addr']
11         pincode = request.POST['pin']
12         book = request.POST['bname']
13         author = request.POST['aname']
14         genre = request.POST['gname']
15         pic = request.POST['fname']
16
17         dbook = BookDonation(name=name,email=email,mobile=mobile,address=address,pincode=pincode,book=bookname,author=authorname,genre=genre,pic=img)
18         dbook.save()
19
20         messages.success(request,"your feedback has been submitted.")
21         return render(request,'home/home.html')
22     else:
23         return render(request,'donation/bookdonation.html')
```

### models.py

```
1  from django.db import models
2  from datetime import datetime
3
4  # Create your models here.
5  class BookDonation(models.Model):
6      name=models.CharField(max_length=200,blank=True)
7      email=models.CharField(max_length=200,blank=True)
8      mobile=models.IntegerField(blank=True)
9      address=models.TextField(blank=True)
10     pincode=models.IntegerField(blank=True)
11     bookname=models.CharField(max_length=200,blank=True)
12     authorname=models.CharField(max_length=200,blank=True)
13     genre=models.CharField(max_length=200,blank=True)
14     img=models.ImageField(upload_to='donate/%Y/%m/%d/',blank=True)
15     date=models.DateTimeField(default=datetime.now)
16
17
18     def __str__(self):
19         return self.name
```

## Collection:

### views.py

```
1  from django.shortcuts import render
2  from .models import Book_details
3  from .models import Category
4  # Create your views here.
5  def collection(request):
6      dests = Book_details.objects.all()
7      cat=Category.objects.all()
8      l=len(cat)
9      cat1=Category.objects.all()[0:l//2]
10     cat2=Category.objects.all()[l//2:]
11
12     context={
13         'dests' : dests,
14         'cat1': cat1,
15         'cat2': cat2
16     }
17     return render(request,'collection/collection.html',context)
```

## models.py

```
1  from django.db import models
2
3  # Create your models here.
4  class Category(models.Model):
5      catname=models.CharField(max_length=50)
6
7
8      def __str__(self):
9          return self.catname
10
11 class Book_details(models.Model):
12     category = models.ForeignKey(Category, on_delete=models.CASCADE) .....
13     name=models.CharField(max_length=100)
14     slug = models.SlugField(unique=True)
15     auther= models.CharField(max_length=50)
16     description = models.TextField()
17     price = models.DecimalField(max_digits=10, decimal_places=2)
18     image = models.ImageField(upload_to='pics')
19     purchase_link = models.URLField(max_length=500,blank=True)
20     bestseller= models.BooleanField(default=False)
21     bestforchildren = models.BooleanField(default=False)
22
23
24     def __str__(self):
25         return self.name
```

## OneView:

### models.py

```
1  from django.db import models
2
3  # Create your models here.
4  class Book_rating(models.Model):
5      book_id=models.IntegerField()
6      rating=models.IntegerField(default=0)
7
8      def __str__(self):
9          return str(self.book_id)
```

### views.py

```
1  from django.shortcuts import render
2  from collection.models import Book_details
3  from collection.models import Category
4  from .models import Book_rating
5  from django.contrib import messages
6  # Create your views here.
7  def oneview(request,view_id):
8      book_view=Book_details.objects.get(id=view_id)
9      brat=Book_rating.objects.filter(book_id=view_id)
10     cat=Category.objects.all()
11     l=len(cat)
12     cat1=Category.objects.all()[1//2]
13     cat2=Category.objects.all()[1//2:]
14     l=list(brat)
15     s=0
16     for i in l:
17         t=i.rating
18         s=s+t
19
20     if len(l)!=0:
21         b=s/len(l)
22     else:
23         b=0
```

```

25     ....context={
26         'book_view': book_view,
27         'br' : b,
28         'cat' : cat,
29         'cat1' : cat1,
30         'cat2' : cat2
31     }
32     return render(request,'book_disp/display.html',context)
33
34     def bookrating(request,view_id):
35         if request.method == 'POST':
36             b_id = view_id
37             rating = request.POST['rating']
38
39             bookrating = Book_rating(rating=rating,book_id=b_id)
40             bookrating.save()
41
42             messages.success(request,"your rating has been submitted")
43             return render(request,'home/home.html')
44
45     else:
46         return render(request,'book_disp/display.html')

```

## Project Configuration:

```

1 """
2 Django settings for bookstore project.
3
4 Generated by 'django-admin startproject' using Django 3.1.5.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.1/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.1/ref/settings/
11 """
12 import os
13 from pathlib import Path
14
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 BASE_DIR = Path(__file__).resolve().parent.parent
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = 'tf769)d170(n1_q@%28pw8ef0&)rrbfx78akh$+i7tjk61b#1'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []

```

```

31 # Application definition
32
33 INSTALLED_APPS = [
34     'home',
35     'accounts',
36     'contact',
37     'feedback',
38     'collection',
39     'oneview',
40     'donatebook',
41     'django.contrib.admin',
42     'django.contrib.auth',
43     'django.contrib.contenttypes',
44     'django.contrib.sessions',
45     'django.contrib.messages',
46     'django.contrib.staticfiles',
47 ]
48
49 MIDDLEWARE = [
50     'django.middleware.security.SecurityMiddleware',
51     'django.contrib.sessions.middleware.SessionMiddleware',
52     'django.middleware.common.CommonMiddleware',
53     'django.middleware.csrf.CsrfViewMiddleware',
54     'django.contrib.auth.middleware.AuthenticationMiddleware',
55     'django.contrib.messages.middleware.MessageMiddleware',
56     'django.middleware.clickjacking.XFrameOptionsMiddleware',
57 ]
58
59 ROOT_URLCONF = 'bookstore.urls'
60
61 TEMPLATES = [
62     {
63         'BACKEND': 'django.template.backends.django.DjangoTemplates',
64         'DIRS': [BASE_DIR,'templates'],
65         'APP_DIRS': True,
66         'OPTIONS': {
67             'context_processors': [
68                 'django.template.context_processors.debug',
69                 'django.template.context_processors.request',
70                 'django.contrib.auth.context_processors.auth',
71                 'django.contrib.messages.context_processors.messages',
72             ],
73         },
74     },
75 ]
76
77 WSGI_APPLICATION = 'bookstore.wsgi.application'
78
79
80 # Database
81 # https://docs.djangoproject.com/en/3.1/ref/settings/#databases

```

```

82
83 DATABASES = {
84     'default': {
85         'ENGINE': 'django.db.backends.sqlite3',
86         'NAME': BASE_DIR / 'db.sqlite3',
87     }
88 }
89
90
91 # Password validation
92 # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validation
93
94 AUTH_PASSWORD_VALIDATORS = [
95     {
96         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
97     },
98     {
99         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
100    },
101    {
102        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
103    },
104    {
105        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
106    },
107 ]
108
109
110 # Internationalization
111 # https://docs.djangoproject.com/en/3.1/topics/i18n/
112
113 LANGUAGE_CODE = 'en-us'
114
115 TIME_ZONE = 'UTC'
116
117 USE_I18N = True
118
119 USE_L10N = True
120
121 USE_TZ = True
122
123
124 # Static files (CSS, JavaScript, Images)
125 # https://docs.djangoproject.com/en/3.1/howto/static-files/
126
127 STATIC_URL = '/static/'
128 STATICFILES_DIRS = [
129     os.path.join(BASE_DIR, 'static')
130 ]
131
132 STATIC_ROOT = os.path.join(BASE_DIR, 'assests')
133
134 MEDIA_URL = '/media/'
135
136 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
137
138 #-----mail-----
139 EMAIL_HOST = 'smtp.gmail.com'
140 EMAIL_PORT = '587'
141 EMAIL_HOST_USER = 'ydkajal123@gmail.com'
142 EMAIL_HOST_PASSWORD = 'Ilovepokemon123@'
143 EMAIL_USE_TLS = True
144 DEFAULT_FROM_EMAIL = 'TheBookThings <ydkajal123@gmail.com>'
145
146 EMAIL_BACKEND = "django.core.mail.backends.smtp.EmailBackend"

```

## Project urls:

```
16  from django.contrib import admin
17  from django.urls import path,include
18  from django.conf import settings
19  from django.conf.urls.static import static
20  from home import views
21
22
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('',include('home.urls')),
26     path('accounts/',include('accounts.urls')),
27     path('contact/',include('contact.urls')),
28     path('feedback/',include('feedback.urls')),
29     path('collection/',include('collection.urls')),
30     path('oneview/',include('oneview.urls')),
31     path('donatebook/',include(['donatebook.urls'])),
32     path('<cate>/',views.catfun,name='catfunction')
33     ...
34 ]+static(settings.MEDIA_URL,document_root=settings.MEDIA_ROOT)
```

## **5. Facilities required for proposed work**

### **5.1 Hardware Requirements**

- Processor : Intel corei3 and above
- RAM : 1 GB
- HDD : 500GB
- Display : 1024 x 768 High color-64-bit SOFTWARE

### **5.2 Software Requirements**

- Browser : chrome, IE 6.0 Mozilla Firefox
- Database : Sqlite3 Database(Django Administration)
- Operating System : Window 7 and above
- Documentation Tool : Ms Word, MS power point
- Scripting Language : JavaScript
- Front-End Language : HTML,CSS
- Back-End Language : Python3

## 6. References

- <https://www.w3schools.com/>
- <https://www.udemy.com/>
- <https://www.youtube.com>
- <http://www.wikipedia.org>
- <http://www.codeproject.com>