

---

# BOOK SPACE

*A Free Books Website*

---



---

# ACKNOWLEDGEMENTS

In the present world of competition, there is a race of existence in which those who are having the will to come forward succeed. Project is like a bridge between theoretical and practical work. With this willingness, we joined this particular project. First of all, we would like to thank the supreme power, the Almighty God who is obviously the one who has always guided me to work on the one who has always guided me to work on the right path of life. Without his grace this project could not become a reality. Next to him are my parents whom I am greatly indebted to me brought up with love and encouragement to this stage. I am feeling obliged in taking the opportunity to sincerely thank our Professors. Moreover, I am highly obliged in taking the opportunity to sincerely thank all the staff members of the computer department for their generous attitude and friendly behavior. At last but not least I am thankful to all my teachers and friends who have been always helping and encouraging me throughout the year. I have my valuable words to express my thanks, but my heart is full of the favors received from every person.

# ABSTRACT

- ❖ In this project, we made significant usage of both Front-end and Back-end parts in developing our website and understood various parts of usage of many libraries and their scope.
- ❖ We learned using Python, Django, DBMS, Web servers, and many more. We hosted our website on GitHub and are planning to deploy it on the cloud-based service.
- ❖ We also dealt with Unit testing and Integration testing to find out various bugs in our website and resolved them. Overall, the course was really helpful in understanding several modules and functionalities of a website and developing it.

# Index

<b>1. Problem Motivation</b>	<b>04</b>
<b>2. Problem Statement</b>	<b>04</b>
<b>3. Problem Solution</b>	<b>05</b>
<b>4. Project Overall Idea</b>	<b>06</b>
<b>5. Tools Required</b>	<b>07</b>
<b>6. Get Started</b>	<b>08</b>
<b>7. Use case &amp; Class diagrams</b>	<b>10</b>
<b>8. Overview Of Website:</b>	<b>17</b>
a. Home Page	17
b. Donate	18
c. Shop	20
d. Hall Of Fame	24
e. About Us	25
f. Contact Us	26
g. FAQ	27
h. Log In & Sign-Up	28
i. Account Details	30
j. Order Details	31
k. Navbar & Footer	32
l. Privacy Policy & License	33
<b>9. Explanation of Modules of the code</b>	<b>35</b>
<b>10. Details of Database</b>	<b>49</b>
<b>11. Phases of project and time allocation</b>	<b>62</b>
<b>12. References</b>	<b>63</b>

# **Problem Motivation**

The main motivation of the project is to help the poor by giving books for free. The primary focus will be to help the poor, but it will also be helpful to other people as well.

# **Problem Statement**

In these present situations, it is difficult for poor people to even afford food, thus affording money for buying books has become a big burden for their child's education!!

---

# **Problem Solution**

- ❖ This website will play a major role in giving books to the poor for free which will be very helpful for them.
- ❖ The donors will have to donate their old books to our website and we collect them and add them to our website.
- ❖ The users can come to our website and get the books free of cost from our website.

# Project Overall Idea

- ❖ People who want to donate their books can log in to this website and can fill in the details of the book like name, genre, and condition of the book.
- ❖ If the book is in good condition to read, then the user can fill in the address, and then a pick-up guy comes to receive it.
- ❖ Donors will get points or coupons which will be helpful to buy books from the website at a low cost!!
- ❖ The book will be posted on the website with the title, edition, proper photo, and grade of the book.

- 
- ❖ Customers who are interested to buy the book can place an order with only delivery charges and the order will be received as soon as possible.
  - ❖ Customers can keep track of orders and can contact customer care for any further queries and updates.

## Tools Required

- ❖ Browsers like Chrome can run HTML5, CSS3, JavaScript, Jquery, and Bootstrap.
- ❖ To get the project running, you need Python and Django installed. You also require libraries like Pillow, Django-filters also installed.

# Get Started

★ Download git from the link:

<https://git-scm.com/downloads>

★ Download and install python from the link:

<https://www.python.org/downloads/>

★ Open the command prompt and run the command: `git clone https://github.com/bookspace30/BookSpace.git`

★ Then cd to Project folder and then perform the below actions :

- ❖ Run the command: **pip install django**.
- ❖ Next run the command: **pip install django-filter**.
- ❖ Next run the command: **pip install Pillow**

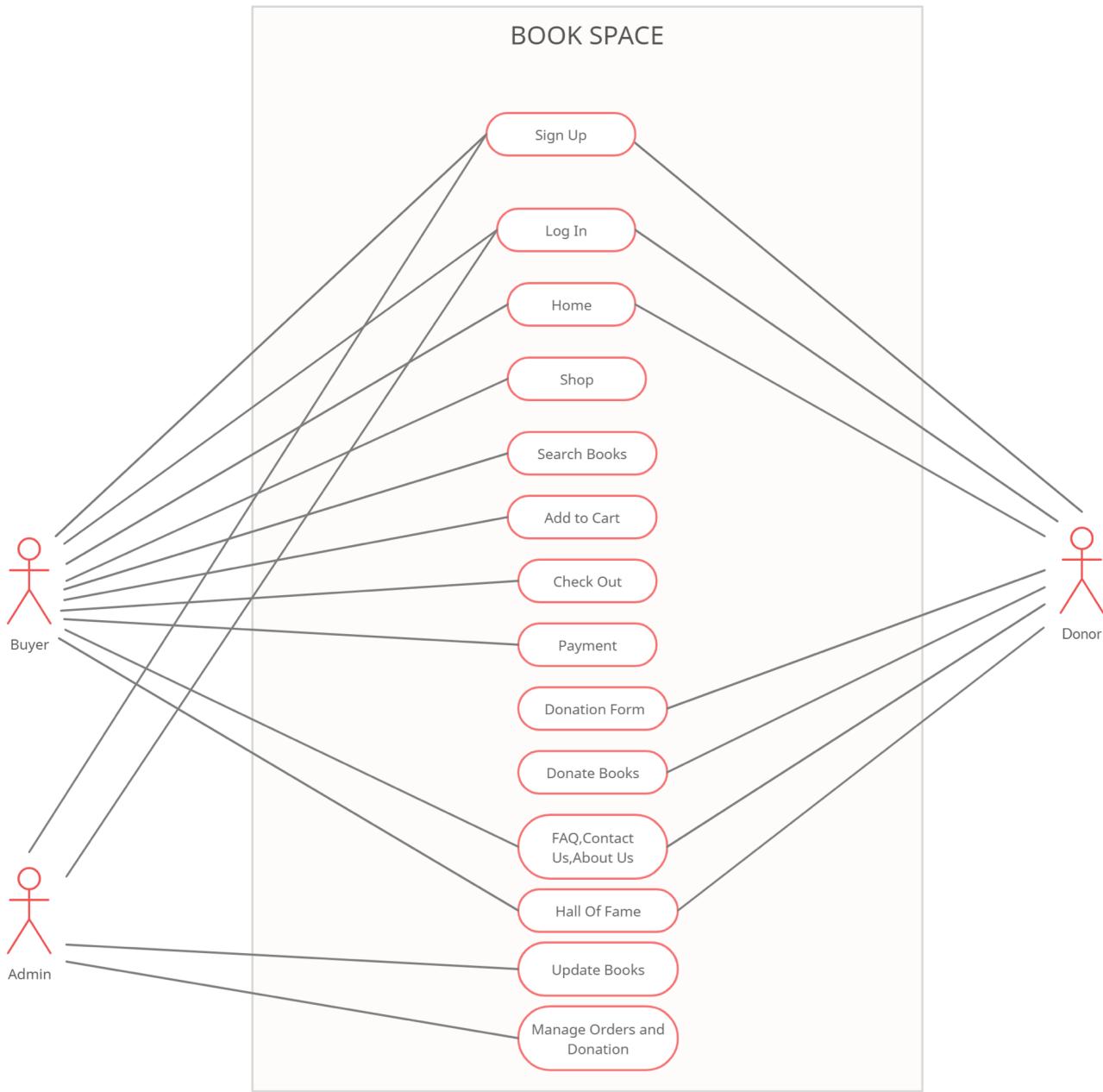
- ❖ Next run the command: **python manage.py makemigrations.**
  - ❖ Next, run the command: **python manage.py migrate**
- ★ Finally to run the server use the following command:  
**python manage.py runserver**
- ★ Then the terminal will prompt to open a URL or else type [\*\*http://127.0.0.1:8000\*\*](http://127.0.0.1:8000).
- ★ Below is the screenshot to get started with running the server.

```
neerajdusa@Neerajs-MacBook-Air hi % python3 manage.py makemigrations
No changes detected
neerajdusa@Neerajs-MacBook-Air hi % python3 manage.py migrate
Operations to perform:
  Apply all migrations: accounts, admin, auth, contenttypes, sessions
Running migrations:
  Applying accounts.0035_delete_finalorder... OK
neerajdusa@Neerajs-MacBook-Air hi % python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

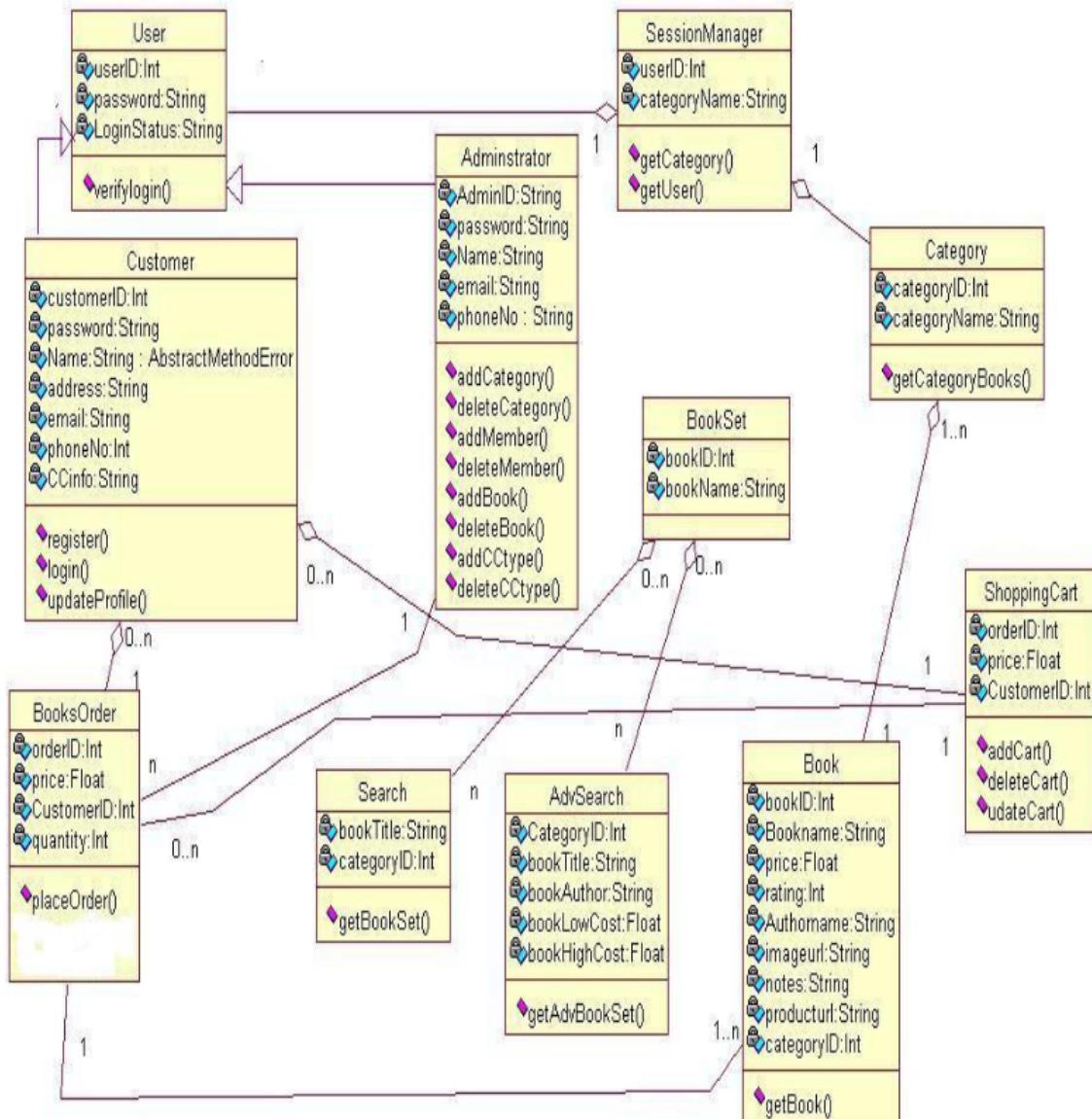
System check identified no issues (0 silenced).
May 08, 2021 - 16:26:02
Django version 3.1.7, using settings 'hi.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

## Use Case Diagram :

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions (Use cases) with the system. Below is the use case diagram of the Book Space system.

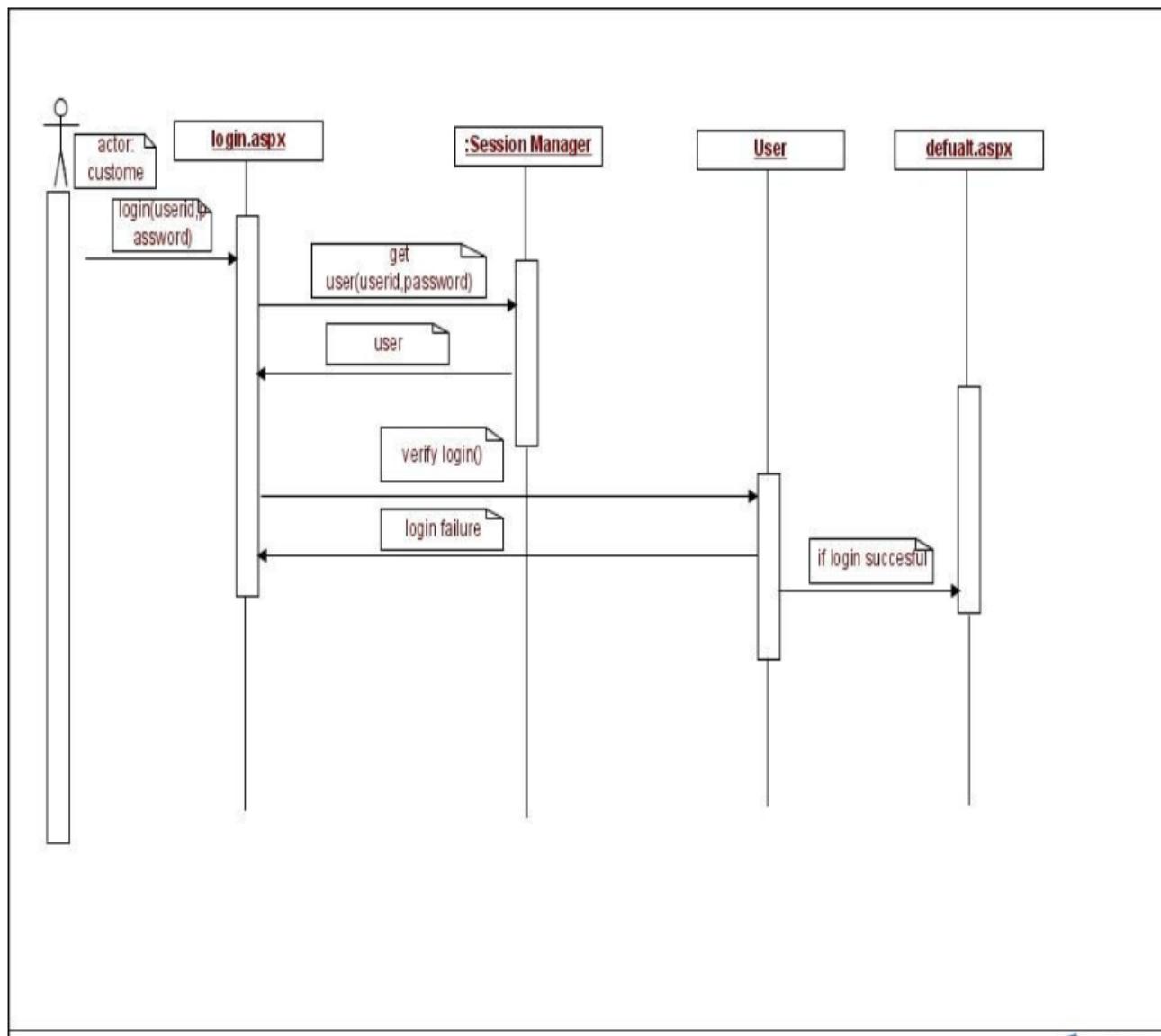


# Class Diagram :

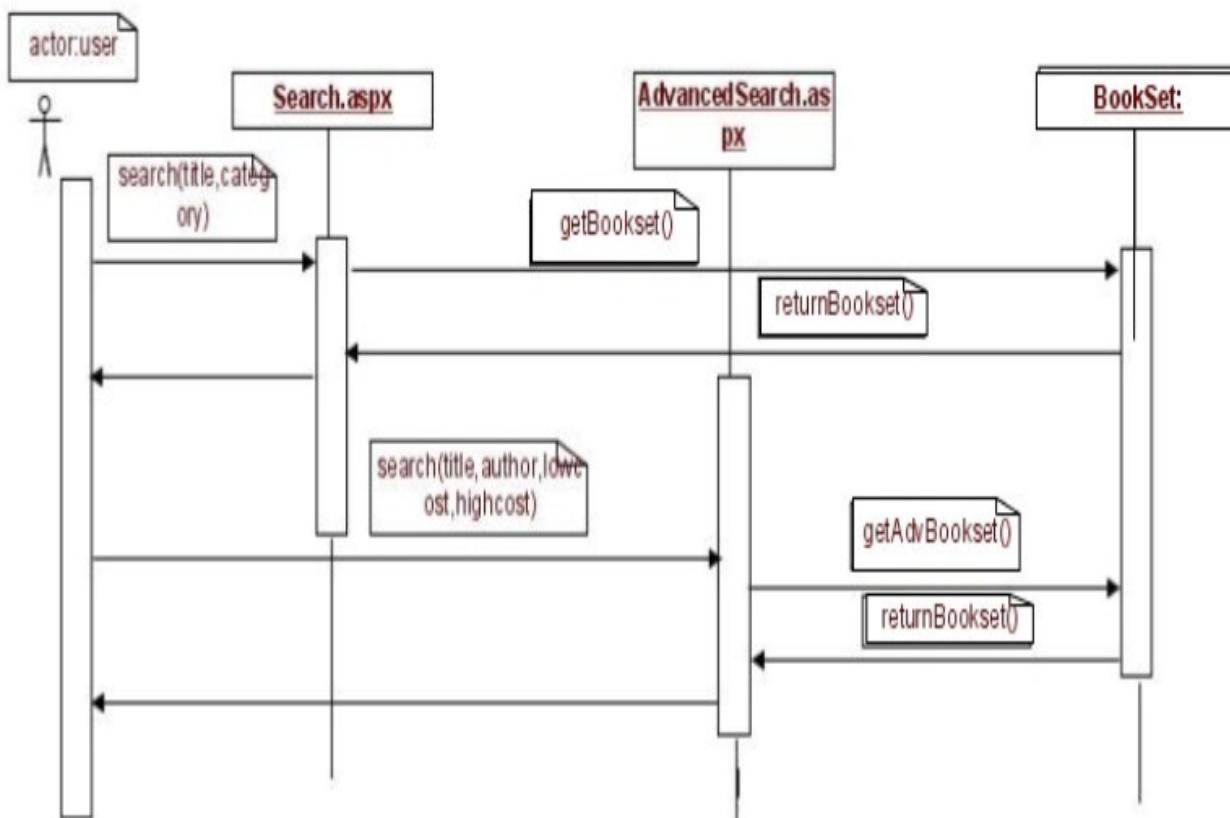


## Sequence Diagrams

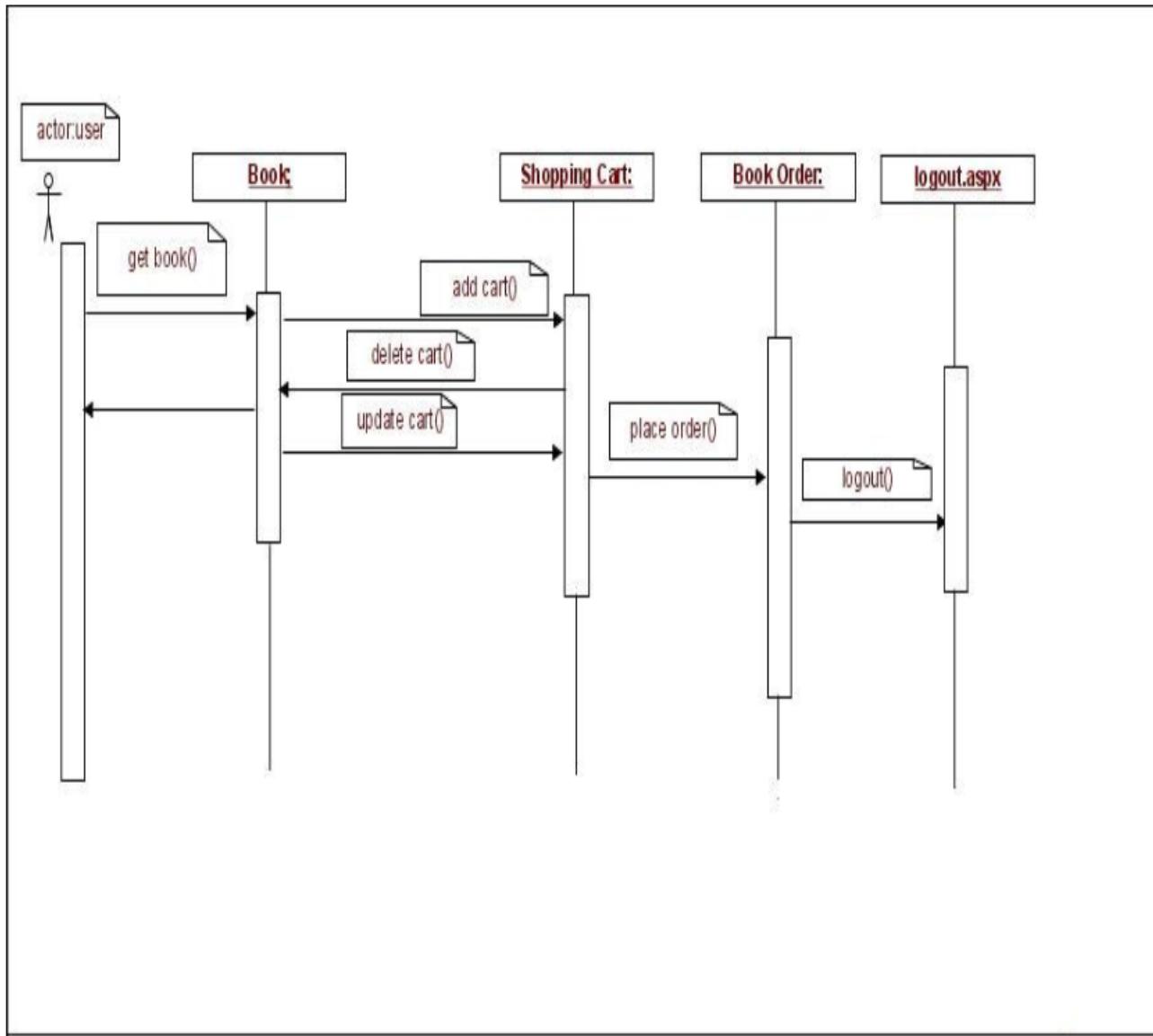
### 1) User Login



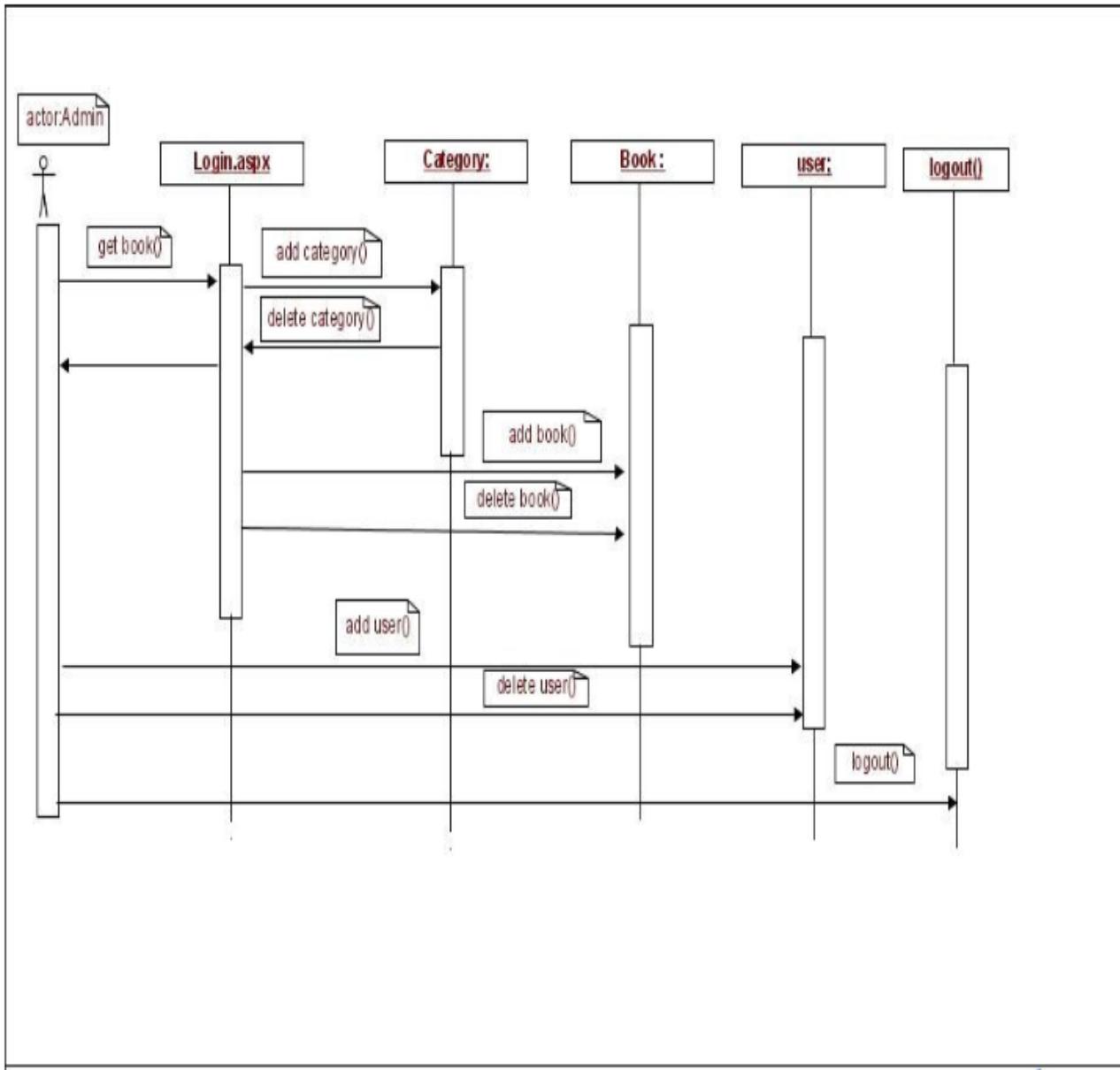
## 2) Book Search



### 3) Add to shopping cart



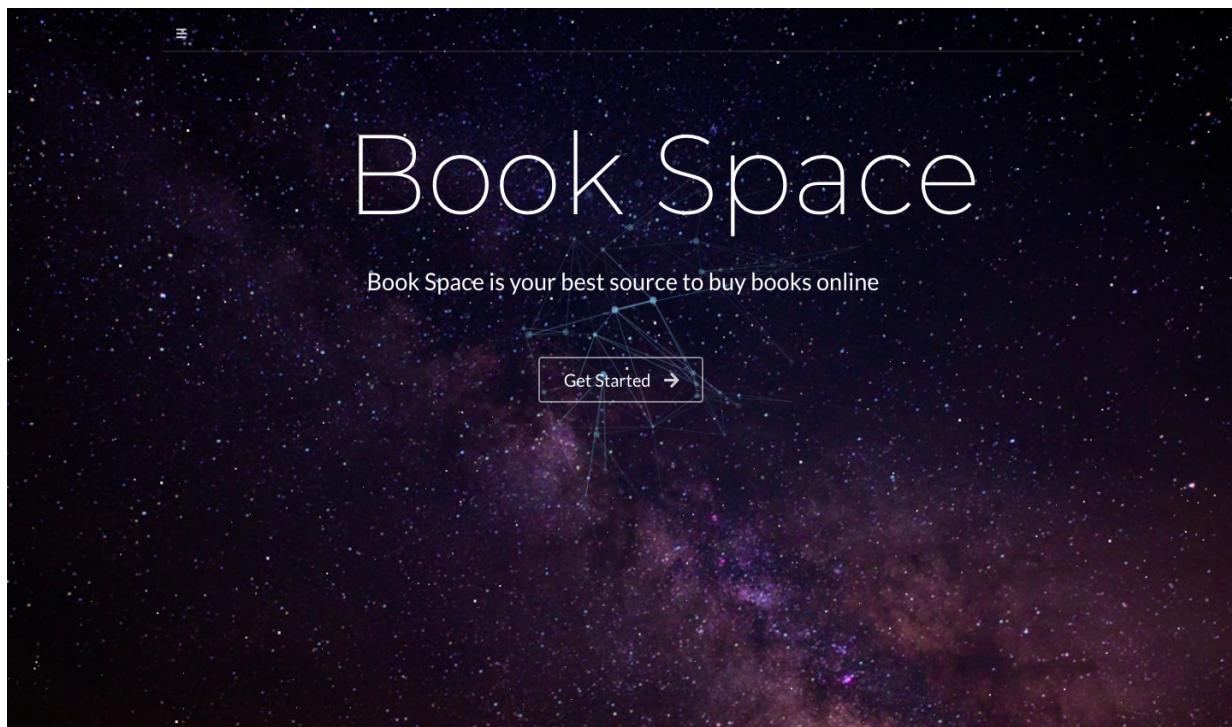
#### 4) Administrator



# Overview Of Website

## HOME

This is the Book Space Home page with innovative UI and attractive animations. Here, at the top left corner, we have a side nav bar through which users can redirect to any page easily. Now, clicking on the GET STARTED button, we will look at the Donate page and explore the website.



# DONATE

This is the donation form where the donors can donate books by filling in the required credentials such as name, email, phone number, the number of books donated, and shipping address.

The screenshot shows a donation form titled "DONATION FORM" set against a background of many books. The form includes fields for First Name, Last Name, Email, Phone Number, Quantity Of Books, Door No., Landmark, City/Town, State, and Pin Code. Each field has a blue icon next to it. A large blue "Submit" button is at the bottom. The top navigation bar includes links for Shop, Donate, Hall Of Fame, About, Hello, Neeraj, and Logout.

Field	Description
First Name	Text input field for the donor's first name.
Last Name	Text input field for the donor's last name.
Email	Text input field for the donor's email address.
Phone Number	Text input field for the donor's phone number.
Quantity Of Books	Text input field for the number of books being donated.
Door No.	Text input field for the door number.
Landmark	Text input field for the landmark.
City/Town	Text input field for the city/town.
State	Text input field for the state.
Pin Code	Text input field for the pin code.

After filling the form, the donors can click the submit button, and then there will be a thank you page appreciating such donors. Then users will be redirected to the home page within a few seconds.

# Thank you !

Thank you so much for filling the form.  
Please click this if you are not redirected within few secs.

[Go Home](#)

Have a question? [Click here to visit FAQ](#)

## Why Donate Books?

We are committed to carrying your emotions with your books and making them grow older! Your books can help someone in the same way they did for you. Let's become a cause for the change & make a difference.

# SHOP

This is the shop page where we can find different kinds of books. On this page, one can find a stack of all books and we have a filter that has category type, ISBN, and name of the book. There is a dedicated “Add to Cart” button, by which one can add the books directly into the cart.

The screenshot shows the Book Space shop page. At the top, there is a navigation bar with links for Book Space, Shop, Donate, Hall Of Fame, About, Hello, Neeraj, Logout, and a shopping cart icon. Below the navigation bar, the word "Shop" is centered above a search bar. The search bar contains fields for "Categories", "isbn:", and "Name contains:" with a "Search" button. Below the search bar, there is a grid of eight book covers:

- Clean Code** by Robert C. Martin (Cover image: Black background with a glowing ring)
- Algorithm Design** (Cover image: Blue background with orange diagonal stripes)
- charaka samhita** by Acharya Charaka (Cover image: Green background with a stylized plant)
- POWER ELECTRONICS** (Cover image: Red background with circuit board patterns)
- Engineering Mechanics: STATICS** (Volume 1, SI Version) (Cover image: Red background with a hand holding a pencil)
- WILD REMEDIES** (Cover image: Dark green background with a close-up of a leaf)
- COMPUTER SCIENCE with C++** (Volume II) (Cover image: Yellow and orange background with a large red arrow pointing right)
- BASIC STRUCTURAL ANALYSIS** by C. S. REDDY (Cover image: Blue and white geometric pattern)

Each book cover includes its title, author, and a row of buttons for "Add to Cart" and price information (₹245.0 ₹0.00).

# FILTER

The screenshot shows the Book Space website's header. It features a logo with colorful books, the text "BOOK SPACE", and a navigation bar with links for "Shop", "Donate", "Hall Of Fame", "About", "Hello, Neeraj", and "Logout". A shopping cart icon with a red "0" is also present.

The screenshot shows a search results page with the title "Shop". The search bar contains "Categories: MPC Inter", "ISBN: 12321", "Name contains: chemistry", and a "Search" button. Below the search bar, a book titled "Chemistry" by Robert C. Martin is displayed. The book cover features a blue background with molecular structures. The price is listed as ₹450.0 ₹0.00. There is an "Add to Cart" button.

# BUY NOW

This is the broad view of each book. We can find the complete details regarding the selected book and place an order or add it to the cart.

The screenshot shows a detailed book page for "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert Cecil Martin. The page includes the book cover, author information (Robert Cecil Martin, ISBN: 54120035), book condition (VERY GOOD), and delivery options (Cash On Delivery, 7 Day Replacement, In Stock). There are "Add to Cart" and "Buy Now" buttons. A "Description" section at the bottom states: "Ideal for an introductory undergraduate course in control systems for engineering students."

# CART

- ❖ Here we can find the books that are added to the cart. It shows the name of the book, price, quantity, total amount, and savings.
- ❖ You can see that every time you add a book to the cart, in the top right corner, the cart's count will be updated.
- ❖ If you are ready with the final set of books, click on the checkout button.

The screenshot shows the BookSpace website's shopping cart page. At the top, there is a navigation bar with links for 'Shop', 'Donate', 'Hall Of Fame', 'About', 'Hello, Neeraj' (logged in user), 'Logout', and a shopping cart icon with a red notification badge showing '4'. Below the navigation bar, the word 'Shop' is centered. A 'Continue Shopping' button is visible. The main content area displays a summary of the cart: 'Items: 4', 'Total: ₹ 0.00', and 'Total Savings: ₹ 1095.00'. A green 'Checkout' button is located to the right of the summary. Below this, a table lists the three items currently in the cart:

Item	Price	Quantity	Total	BS Total
Algorithm Design	₹300.00	2 ▲	₹600.00	₹ 0.00
charaka samhita handbook on ayurveda Acharya Charaka	₹150.00	1 ▲	₹150.00	₹ 0.00
POWER ELECTRONICS Components, applications	₹345.00	1 ▲	₹345.00	₹ 0.00

# CHECKOUT & PAYMENT

- ❖ Here the order summary of the books you want to buy is displayed. It shows the name, quantity, total savings, shipping charges, and net total.
- ❖ Once you confirm these details, fill in the shipping information and click on make payment.
- ❖ After the confirmation, a pop-up message will be shown for successful payment.

BOOK SPACE [Shop](#) [Donate](#) [Hall Of Fame](#) [About](#) Hello, Neeraj Logout [Cart](#) 4

Address.. City..

State.. Zip code..

Paypal Options :

Cash On Delivery(COD)

[Make payment](#)

Order Summary			
	Algorithm Design	₹300.00	₹ 0.00 x 2
	charaka samhita	₹150.00	₹ 0.00 x 1
	Power Electronics	₹345.00	₹ 0.00 x 1
Items: 4			
Total Savings: ₹ 1095.00			
Shipping Charges: ₹ 174			
Total: ₹ 174			

# HALL OF FAME

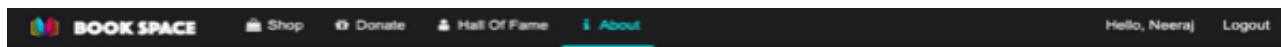
- ❖ The top 15 donors who donated the books are listed here in descending order.
- ❖ Here, we can see that the avatar, name, and no of donated books are displayed for each donor.

The screenshot shows the 'Hall Of Fame' section of the BookSpace website. At the top, there is a navigation bar with links for 'Shop', 'Donate', 'Hall Of Fame' (which is underlined in blue), and 'About'. On the right side of the bar, it says 'Hello, Neeraj' and has a 'Logout' button. The main title 'Hall Of Fame' is centered above three profile cards. Each card contains an avatar, the donor's name, and the number of books donated. The first donor, Harish Mullagura, has a white background and donated 130 books. The second, Ganeshula Rakesh, has an orange background and donated 97 books. The third, Jayanth Sri Ram, has an orange background and donated 95 books.

Donor Name	Books Donated
Harish Mullagura	130
Ganeshula Rakesh	97
Jayanth Sri Ram	95

# ABOUT US

The About Us page is one of the most important pages for the website. It is an opportunity where the users get to know about the website. It also gives brief information about important pages on our website.



## About Us

**How it works?**

1. Fill the form
2. Upload Books
3. Donate

**Worry not!** Because Book Space is here to answer all your literary prayers. Knowledge is priceless but the books come with a bill and the Book Space is committed to bring to you the best of the brilliant from the world of written text, at prices which are literally a steal. Donate the books for us we will use every weapon there is to meet your bibliophilic fantasies. So fellas' put on your reading capes because it is time to fly into the mystical world of best sellers.

**Shop**

Knowledge is a powerful weapon that everyone should have to change and empower the world. Following the books on Book Space are available for every citizen of India. Buy a book and become kingpin of world

**Donate**

We are on a mission to create an ecosystem where no reader in the country is left to complete his/her education due to the affordability and unavailability of books. Since books are made to spread the knowledge.

**Hall of Fame**

Our website provides the top 15 donors are posted in the Hall of Fame with their names, photos and contact information. You can search the donor by donation id. User can see the number of books donated by the donor by their id.

**BOOK SPACE**

Shop  
Donate  
Hall Of Fame  
About Us  
Contact Us

**BOOKS**

Engineering Books  
Medical Books  
Economics  
JEE MPC  
JEE BIPC

**HELP**

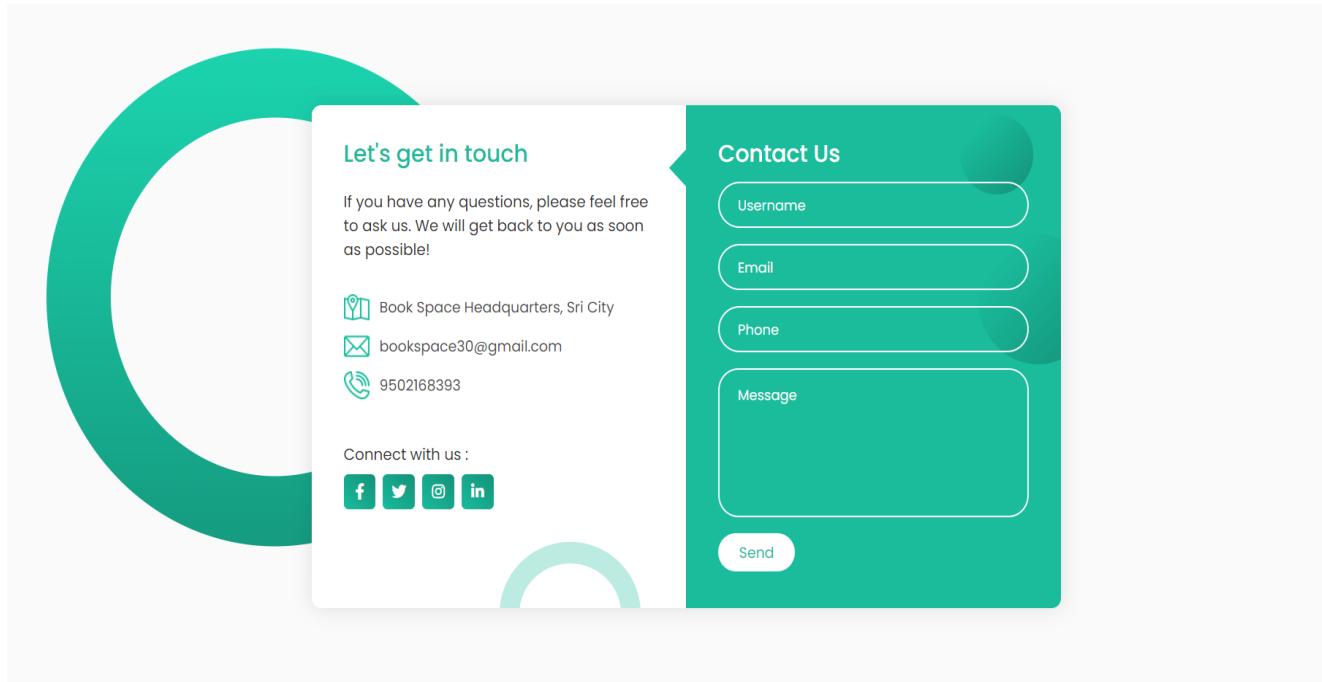
Privacy Policy  
License  
Book Condition Guidelines  
Return Policy  
FAQ

**CONTACT**

India  
bookspace30  
+ 01 234 567 88  
+ 01 234 567 89

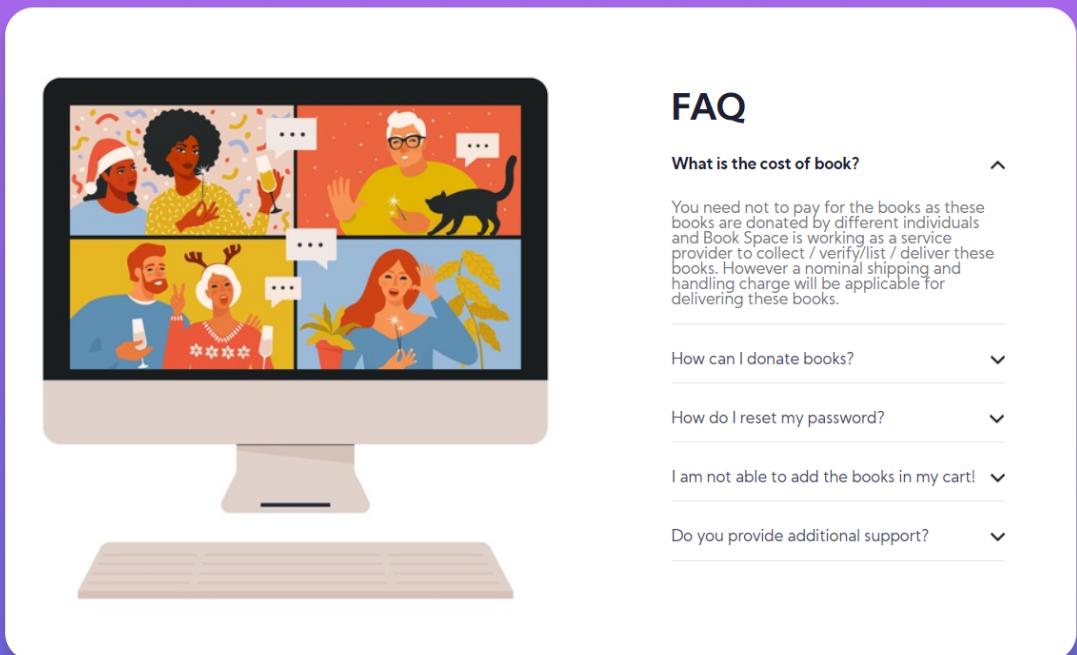
# CONTACT US

In this section, the users can directly contact us through email, phone number, or through any of the social media websites. Also, there is a contact form with name, email, phone, and message. We will reply as soon as possible through email.



# FAQ

FAQ stands for Frequently Asked Questions. An FAQ page on the website is an organized collection of valuable information that the user asks about products and services. It improves customer service by answering common questions.



The image shows a computer setup with a monitor and a keyboard. On the monitor, there is a festive collage with four panels. The top-left panel shows two people in Santa hats. The top-right panel shows an older man with glasses and a cat. The bottom-left panel shows a couple in reindeer antlers. The bottom-right panel shows a woman holding a lit candle. To the right of the monitor is a white card titled "FAQ". Under the title, there is a question "What is the cost of book?" followed by a detailed answer. Below this are five more questions with dropdown arrows: "How can I donate books?", "How do I reset my password?", "I am not able to add the books in my cart!", and "Do you provide additional support?".

**FAQ**

**What is the cost of book?** ^

You need not to pay for the books as these books are donated by different individuals and Book Space is working as a service provider to collect / verify/list / deliver these books. However a nominal shipping and handling charge will be applicable for delivering these books.

How can I donate books? ▾

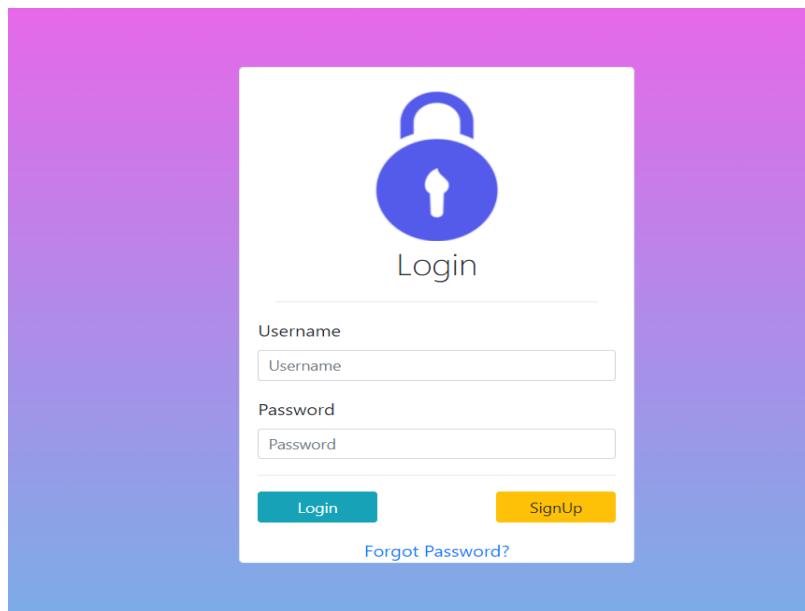
How do I reset my password? ▾

I am not able to add the books in my cart! ▾

Do you provide additional support? ▾

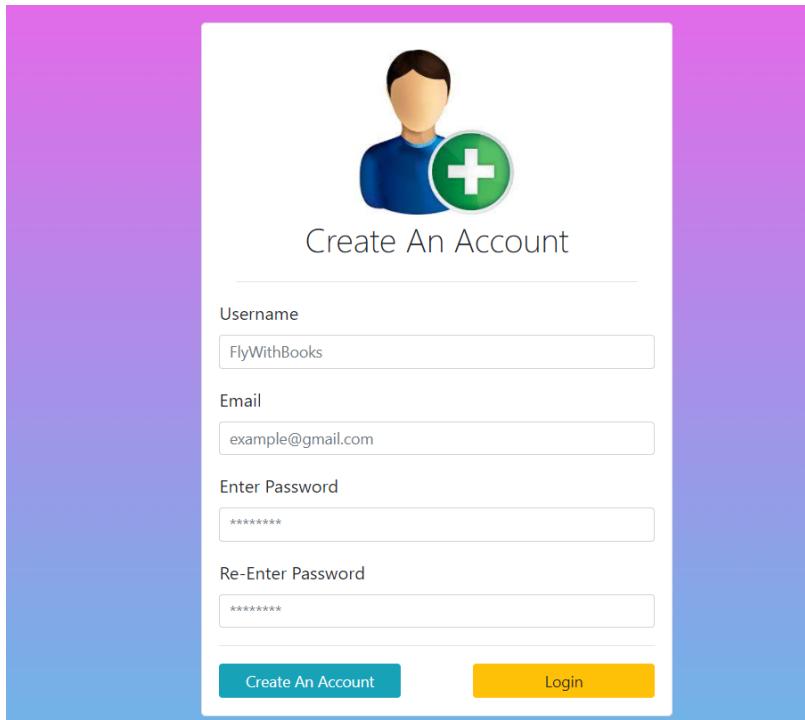
# LOGIN

- ❑ The existing users of our website can Log In through username and password in this section.
- ❑ Here, we have implemented it by HTML form attribute and added styling attributes using CSS classes and pseudo special classes.
- ❑ Now if you have forgotten your password? worry not. You can reset your password using your email address with a reset link sent to mail.
- ❑ Now, If you are a new user around here, you can sign up from here. The below image is of the log-in page.



# SIGNUP

In this section, one can create an account and register themselves. New users can sign up by entering a username, email, password, re-enter the password, and register themselves which will help in building a community. Tada!, creation of the account is done and we always welcome you to join our family and we are very happy to serve you.



# ACCOUNT DETAILS

The users can find all their essential details in this section. This page can be found in the “Hello User” section. Details like name, email, phone number, address are present here. If any user wishes to update their details in the future, one can come to this section and can easily update them. Here, one can upload their profile image also.

The screenshot shows the 'Account Details' section of the BookSpace website. On the left, there is a sidebar with a blue background featuring a user icon and the text 'My Account'. Below that are links for 'Orders' and 'Donated: 12'. The main area has a teal background and displays account information in a list format:

Account Details	
User icon	Neeraj
Email icon	neerajdusa@gmail.com
Phone icon	9984752345
Address icon	2-2-185/B/10&11
Address icon	Near SBI Bank
Address icon	Hyderabad
Address icon	500013
Address icon	Telangana

At the top of the page, there is a black navigation bar with the BookSpace logo, 'Shop', 'Donate', 'Hall Of Fame', 'About', 'Hello, Neeraj', and 'Logout' buttons.

# ORDER DETAILS

In this section, users can find all the history of their orders. Users can find all the specific details of the order here. Here, the orders log is nicely shown with 4 containers like Date when the product was placed, Transaction ID, Quantity of books purchased, and status of the order.

The screenshot shows a user profile interface on the left and an 'Order Details' table on the right. The profile area includes a user icon, 'My Account', 'Orders', and 'Donated: 12'. The 'Orders' link is highlighted with a teal bar. The 'Order Details' table lists 15 entries with columns for Date Ordered, Transaction ID, Quantity, and Status.

Date Ordered	Transaction ID	Quantity	Status
May 4, 2021, 10:04 a.m.	1620122660.720915	6	Pending
May 4, 2021, 10:04 a.m.	1620214161.183008	2	Pending
May 5, 2021, 11:29 a.m.	1620216487.941857	2	Pending
May 5, 2021, 12:08 p.m.	1620216778.02423	1	Pending
May 5, 2021, 12:12 p.m.	1620316211.262645	8	Pending
May 6, 2021, 3:50 p.m.	1620316265.19612	4	Pending
May 6, 2021, 3:51 p.m.	1620318532.618821	2	Pending
May 6, 2021, 4:28 p.m.	1620318625.188875	2	Pending
May 6, 2021, 4:30 p.m.	1620318732.394411	3	Pending
May 6, 2021, 4:32 p.m.	1620318962.066872	2	Pending
May 6, 2021, 4:36 p.m.	1620321052.468769	2	Pending
May 6, 2021, 5:10 p.m.	1620321079.237306	2	Pending
May 6, 2021, 5:11 p.m.	1620322897.270246	4	Pending
May 6, 2021, 5:41 p.m.	1620323049.723911	2	Pending
May 6, 2021, 11:14 p.m.	1620323071.769409	2	Pending
May 6, 2021, 11:14 p.m.	1620323101.233402	4	Pending

# NAVBAR AND FOOTER

In the navbar, users can redirect to pages easily by clicking on them.

In the footer, users can rate the website by selecting the number of ribbons on the top left corner of the footer. Also, there are all pages listed with social accounts of Book Space from where users can visit the other pages quickly and can subscribe to know the latest information via Email.

The screenshot displays the Book Space website's layout. At the top, a dark header bar contains the "BOOK SPACE" logo with a stylized book icon, followed by navigation links: "Shop", "Donate", "Hall Of Fame", and "About". On the far right, it shows "Hello, Neeraj" and a "Logout" link. Below the header, a large white sign-up form asks, "Are you wanting to continue your quest for Book Space enlightenment?", followed by "SIGN UP NOW AND NEVER MISS OUT." and a text input field labeled "E-mail". A "Sign-up" button is centered below the input field. At the bottom, a dark footer section includes a "Rate Our Website" rating bar with five yellow stars, a "Brought To You By" list of names (Neeraj Dusa, Harish Mulagura, Jayanth Sriram, Rakesh Ganeshula, Anirudh Jakholia), a "Pages" list (Book Space, Shop, Donate, Hall of Fame, About Us, Contact Us, FAQ), a "Connect With Us" list of social media links (LinkedIn, Facebook, Instagram, Twitter, GitHub, Codepen), and a "Subscribe" form with an input field for email and a "Submit" button. A note at the bottom of the subscribe area states, "You can opt out of our newsletters at any time. See our [privacy policy](#)".

# PRIVACY POLICY

One can find the clear details of privacy information and cookies that the website is collecting from users. All the details of the user are safely hidden in the database with strong SSL 2.0 Security & anti-malware software and will never be exposed to the public.

Book Space

## Privacy Policy

### Privacy Statement

Book Space is committed to protecting your privacy and developing technology that gives you the most powerful and safe online experience. This Statement of Privacy applies to the Book Space website and governs data collection and usage. By using our website, you consent to the data practices described in this statement.

[Privacy Policy](#)

[License](#)

### Collection of your Personal Information

When you visit our Website, and make a purchase or attempt to make a purchase through our Website, we also collect certain information from you, including your name, billing address, shipping address, payment information, email address, and phone number.

We will only collect and process data about you where we have a lawful basis. Lawful bases include where you have given consent, where processing is necessary for the delivery of product you have ordered through our website and other "legitimate interests." Where we rely on your consent to process Data, you have the right to withdraw or decline your consent at any time and where we rely on legitimate interests, you have the right to object.

# LICENSE

It shows the license of the Book Space website. It displays the ownership of the website, redistribution, terms & conditions, warranty, termination, and copyrights of the Book Space website.

Book Space

## License

[Privacy Policy](#)

[License](#)

### Definitions

"Licensor" means any person or entity that distributes its Work.

"Software" means the original work of authorship made available under this License.

"Work" means the Software and any additions to Software that are made available under this License.

"This Agreement" refers to this license agreement, and terms and conditions specified herein.

"Web Template" refers to Book Space project, website template, theme, skin, or any copyrightable work licensed under this Agreement.

"Book Space" refers to the website [www.bookspace.com](http://www.bookspace.com), its owner, and its successors, or manufacturer of this Template.

### Agreement

After reading this agreement carefully, if you (Licensee) do not agree to all of the terms of this agreement, you may not download or use this Web Template. Your download or use of this Web Template indicates your acceptance of This Agreement. All updates to the Web Template provided by Book Space shall be considered part of the Web Template and subject to the terms of this Agreement.

# Modules Explanation

## Customer Model:

```
class Customer(models.Model):
    user = models.OneToOneField(User, null=True, blank=True,
on_delete=models.CASCADE)
    name = models.CharField(max_length=200, null=True)
    email = models.CharField(max_length=200, null=True)
    phone = models.CharField(max_length=200, null=True)

    door = models.CharField(max_length=200, null=True)
    landmark = models.CharField(max_length=200, null=True)

    pincode = models.CharField(max_length=200, null=True)
    city = models.CharField(max_length=200, null=True)
    state = models.CharField(max_length=200, null=True)

    profile_pic = models.ImageField(default="profile1.png",
null=True, blank=True)
    date_created = models.DateTimeField(auto_now_add=True,
null=True)
    booksdonated = models.IntegerField(default=0)

    def __str__(self):
        return self.name
```

## **Explanation:**

Every customer has a name, email, phone, door, landmark, pin code, city, state, profile pic, and an account created date. This data can be used in orders, shipping addresses, etc. One can update them on the account details page.

### **Category Model:**

```
class Category(models.Model):
    name = models.CharField(max_length=200, null=True)

    def __str__(self):
        return self.name
```

### **Explanation:**

This model is used to categorize books on the shop page. Ex-JEE Books, Engineering, Medical, etc.

### **Contact Us Model:**

```
class Contact(models.Model):
    name = models.CharField(max_length=200, null=True)
    email = models.EmailField(max_length=200, null=True)
    phone = models.CharField(max_length=13, null=True)
    message = models.CharField(max_length=200, null=True)

    def __str__(self):
        return self.name
```

### **Explanation:**

This model is used on the contact us page of the website. It stores the user's name, email, phone, and message.

## **Product Model:**

```
class Product(models.Model):
    CONDITION = (
        ('LIKE NEW', 'LIKE NEW'),
        ('VERY GOOD', 'VERY GOOD'),
        ('GOOD', 'GOOD'),
        ('FAIR', 'FAIR'),
        ('POOR', 'POOR'),
    )

    TAGS = (
        ('1', 'Engineering'),
        ('2', 'Medical'),
        ('3', 'Economics'),
        ('4', 'MPC Inter'),
        ('5', 'BiPC Inter'),
    )

    image = models.ImageField(null=True, blank=True)
    name = models.CharField(max_length=200, null=True)
    price = models.FloatField(null=True)
    author = models.CharField(max_length=200, null=True, blank=True)
    publisher = models.CharField(max_length=200, null=True,
                                 blank=True)
    isbn = models.CharField(max_length=200, null=True, blank=True)
    condition = models.CharField(max_length=200, null=True,
                                 choices=CONDITION)
    description = models.CharField(max_length=200, null=True,
                                   blank=True)
    date_created = models.DateTimeField(auto_now_add=True,
                                         null=True)
    categories = models.ManyToManyField(Category, choices=TAGS)
    # tags = models.ForeignKey(Tag, null=True,
    # on_delete=models.SET_NULL)
    digital = models.BooleanField(default=False, null=True,
                                 blank=True)
    product_count = models.IntegerField(default=0)

    def __str__(self):
        return self.name

@property
```

```
def imageURL(self):
    try:
        url = self.image.url
    except:
        url = ''
    return url
```

## Explanation:

The product model has fields like condition, tags, image, name, price, author, publisher, ISBN, description, date\_created, categories, digital, product\_count. This model can be used for admin to get information about products. Image URL will return the image section in each book detail.

## Order Model:

```
class Order(models.Model):
    STATUS = (
        ('Pending', 'Pending'),
        ('Out for delivery', 'Out for delivery'),
        ('Delivered', 'Delivered'),
    )

    customer = models.ForeignKey(Customer, null=True,
on_delete=models.SET_NULL)
    product = models.ForeignKey(Product, null=True,
on_delete=models.SET_NULL)
    date_ordered = models.DateTimeField(auto_now_add=True,
null=True)
    status = models.CharField(max_length=200, null=True,
choices=STATUS)
    note = models.CharField(max_length=1000, null=True)
    complete = models.BooleanField(default=False)
    transaction_id = models.CharField(max_length=100, null=True)

    def __str__(self):
```

```

    return "%s - %s - with transaction id - %s" % (self.id,
self.customer.name, self.transaction_id)

@property
def shipping(self):
    shipping = False
    orderitems = self.orderitem_set.all()
    for i in orderitems:
        if i.product.digital == False:
            shipping = True
    return shipping

@property
def get_cart_total(self):
    orderitems = self.orderitem_set.all()
    total = sum([item.get_total for item in orderitems])
    return total

@property
def get_cart_items(self):
    orderitems = self.orderitem_set.all()
    total = sum([item.quantity for item in orderitems])
    return total

@property
def get_name(self):
    orderitems = self.orderitem_set.all()
    return orderitems

```

## Explanation:

Order model has fields like STATUS, customer, product, date\_ordered, status, note, complete, transaction\_id. This data can be used for the admin to get detailed information about orders given by the customer.

## **Order Item Model:**

```
class OrderItem(models.Model):
    product = models.ForeignKey(Product, on_delete=models.SET_NULL,
null=True)
    order = models.ForeignKey(Order, on_delete=models.SET_NULL,
null=True)
    quantity = models.IntegerField(default=0, null=True, blank=True)
    date_added = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return "%s) Name - %s, Book - %s, Quantity-%s" %
(self.order.id, self.order.customer.name, self.product.name,
self.quantity)

@property
def get_total(self):
    total = self.product.price * self.quantity
    return total
```

## **Explanation:**

Order item model has fields like product, order, quantity, date\_added. Admin can get the whole details of the order and the customer who ordered and the product attached as a foreign key to this class.

## **Shipping Address Model:**

```
class ShippingAddress(models.Model):
    customer = models.ForeignKey(Customer,
on_delete=models.SET_NULL, null=True)
    order = models.ForeignKey(Order, on_delete=models.SET_NULL,
null=True)
    address = models.CharField(max_length=200, null=False)
    city = models.CharField(max_length=200, null=False)
    state = models.CharField(max_length=200, null=False)
    zipcode = models.CharField(max_length=200, null=False)
    date_added = models.DateTimeField(auto_now=True)

    def __str__(self):
        return "%s - %s - with transaction id - %s" %
(self.order.id, self.city, self.order.transaction_id)
```

## **Explanation:**

The shipping address model has fields like a customer, order, address, city, state, zip code, data\_added. This data can be used for the admin to get the information about shipping addresses given by customers.

## **Donate Model:**

```
class Donate(models.Model):
    fstnm = models.CharField(max_length=200, null=True)
    lsnm = models.CharField(max_length=200, null=True)
    email = models.EmailField(max_length=12, null=True)
    phone = models.IntegerField(null=True)
    books = models.IntegerField(null=True)
    doorno = models.CharField(max_length=200, null=True)
    landmark = models.CharField(max_length=200, null=True)
    city = models.CharField(max_length=200, null=True)
    state = models.CharField(max_length=200, null=True)
    pincode = models.CharField(max_length=200, null=True)

    def __str__(self):
        return self.fstnm
```

## **Explanation:**

The Donate model has fields like the first name, last name, email, phone, books, door number, landmark, city, state, and pin code of the donor.

The admin can get a detailed view of these details and can send the representative to the donor and bring the books to the warehouse.

## Register Class:

```
@unauthenticated_user
def registerPage(request):
    form = CreateUserForm()
    if request.method == 'POST':
        form = CreateUserForm(request.POST)
        if form.is_valid():
            user = form.save()
            username = form.cleaned_data.get('username')

            messages.success(request, 'Account was created for ' +
username)

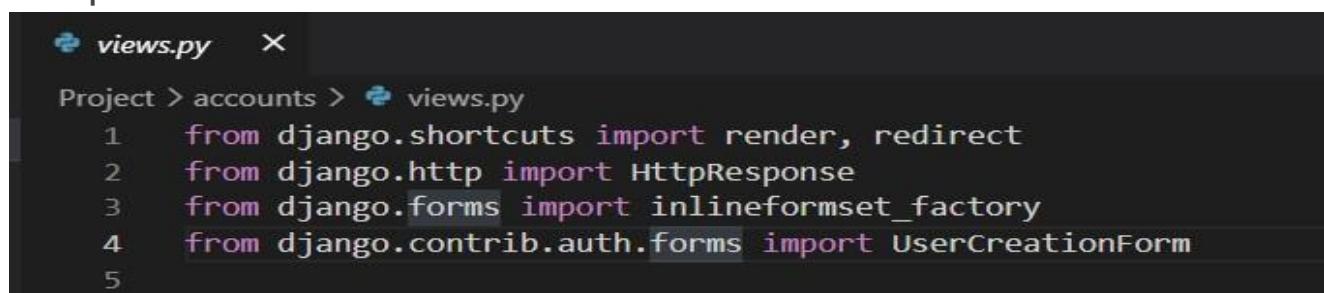
            return redirect('login')

    context = {'values': form}

    # return render(request, 'accounts/Login/signup.html', context)
    return render(request, 'accounts/login2/register.html', context)
```

## Explanation:

Create a view that maps to a template where we can display the user registration form. Import the UserCreationForm that is inherited from the ModelForm class to handle the creation of new users. UserCreationForm has three fields, namely, username, password1, and password2.



The screenshot shows a code editor window with a dark theme. The title bar says 'views.py'. The file path is 'Project > accounts > views.py'. The code is as follows:

```
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from django.forms import inlineformset_factory
from django.contrib.auth.forms import UserCreationForm
```

Then we receive the request from the user through the POST method and then render the request of the user.

### **Login Class:**

```
def loginPage(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(request, username=username,
password=password)

        if user is not None:
            login(request, user)
            return redirect('home')
        else:
            messages.info(request, 'Username OR password is
incorrect')

    context = {}
    return render(request, 'accounts/login2/login.html',
context)
# return render(request, 'accounts/Login/signin.html',
context)
```

### **Explanation:**

This is a Login class. This uses the HTTP POST method to retrieve the data from the user. We used if and else conditions to handle the invalid passwords and it throws an error message that the username or password is incorrect.

## **Account Details Class:**

```
@login_required(login_url='login')
@allowed_users(allowed_roles=['customer'])
def accountdetails(request):
    customer = request.user.customer
    form = CustomerForm(instance=customer)

    if request.method == 'POST':
        form = CustomerForm(request.POST, request.FILES,
instance=customer)
        if form.is_valid():
            form.save()

    context = {'form': form}
    return render(request,
'accounts/AccountDetails/dashb.html', context)
```

## **Explanation:**

This page requires the user to be logged in as a customer. So we are taking an instance of the customer and getting the details of the customer on this page. This uses the POST method to take customer details and we use the if statement to save the form if the form is valid and return the request details to dashb.html.

## **Order Details Class:**

```
@login_required(login_url='login')
def orderdetails(request):
    customer = request.user.customer

    orders = customer.order_set.all()
    order_count = orders.count()
```

```

myFilter = OrderFilter(request.GET, queryset=orders)
myFilter2 = OrderItemFilter(request.GET, queryset=orders)

orders = myFilter.qs
orderitem = myFilter2.qs

context = {'customer': customer, 'orders': orders,
'order_count': order_count,
'myFilter': myFilter, 'myFilter2': myFilter2,
'orderitem': orderitem}
return render(request, 'accounts/AccountDetails/order.html',
context)

```

## **Explanation:**

This class uses all the necessary customer details such as orders, count of orders, filters, order items. It gets all order details from the customer and will be shown on the order details page.

## **Order Processing Class:**

```

@login_required(login_url='login')
@allowed_users(allowed_roles=['customer'])
def processOrder(request):
    transaction_id = datetime.datetime.now().timestamp()
    data = json.loads(request.body)

    if request.user.is_authenticated:
        customer = request.user.customer
        order, created =
Order.objects.get_or_create(customer=customer, complete=False)
    else:
        customer, order = guestOrder(request, data)

    total = float(data['form']['total'])
    order.transaction_id = transaction_id

```

```
if total == order.get_cart_total:
    order.complete = True
    order.status= "Pending"

order.save()

if order.shipping == True:
    ShippingAddress.objects.create(
        customer=customer,
        order=order,
        address=data['shipping']['address'],
        city=data['shipping']['city'],
        state=data['shipping']['state'],
        zipcode=data['shipping']['zipcode'],
    )

return JsonResponse('Payment submitted..', safe=False)
```

## Explanation:

This page requires the user to be logged in as a customer. We get the transaction id by getting the present time and timestamp attached to it. We check if the user is authenticated or not by using the if statement and we check if the total from the website and the cart total is the same or not and then proceed with the order processing for payment. Once payment is done, the order is saved and payment is submitted.

### **Customer Form:**

```
class CustomerForm(ModelForm):  
    class Meta:  
        model = Customer  
        fields = '__all__'  
        exclude = ['user', 'booksdonated']
```

### **Order Form:**

```
class OrderForm(ModelForm):  
    class Meta:  
        model = Order  
        fields = '__all__'
```

### **New User Form:**

```
class CreateUserForm(UserCreationForm):  
    class Meta:  
        model = User  
        fields = ['username', 'email', 'password1',  
        'password2']
```

### **Donate Form:**

```
class DonationForm(UserCreationForm):  
    class Meta:  
        model = Donate  
        fields = '__all__'
```

## **Explanation:**

These are the different types of forms we used in different classes in the views.py file. For each class form, it contains the model and the fields it has used.

## DETAILS ABOUT DATABASE - db.sqlite3

### Django Administration

Django administration

Site administration

ACCOUNTS		
Categorys	<a href="#">+ Add</a>	<a href="#">Change</a>
Contacts	<a href="#">+ Add</a>	<a href="#">Change</a>
Customers	<a href="#">+ Add</a>	<a href="#">Change</a>
Donates	<a href="#">+ Add</a>	<a href="#">Change</a>
Order items	<a href="#">+ Add</a>	<a href="#">Change</a>
Orders	<a href="#">+ Add</a>	<a href="#">Change</a>
Products	<a href="#">+ Add</a>	<a href="#">Change</a>
Shipping addresss	<a href="#">+ Add</a>	<a href="#">Change</a>

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>

**Explanation:** This is the Django Administration page. It can only be accessed by admin. This helps to easily maintain the website. It contains all the accounts of the users and their authentication and authorization.

## Category Model

Category model in the database is shown as below,

Select category to change

Action:	-----	Go	0 of 5 selected
<input type="checkbox"/>	CATEGORY		
<input type="checkbox"/>	BiPC Inter		
<input type="checkbox"/>	MPC Inter		
<input type="checkbox"/>	Economics		
<input type="checkbox"/>	Medical		
<input type="checkbox"/>	Engineering		
5 categories			

**Explanation:** This model is used to categorize books on the shop page. The categories include BiPC Inter, MPC Inter, Economics, Medical, and Engineering. A user can find all these categories on the shop page and select the books according to these. The books can be easily filtered according to these categories and a user can buy them accordingly.

## Contact Us

Contact Us model in the database is shown below.

Select contact to change

Action:   0 of 5 selected

<input type="checkbox"/>	CONTACT
<input type="checkbox"/>	Jayanth
<input type="checkbox"/>	neeraj
<input type="checkbox"/>	Kanduri Jayanth Sri Ram
<input type="checkbox"/>	dsfg
<input type="checkbox"/>	Kanduri Jayanth Sri Ram

5 contacts

This model is used on the contact us page of the website.  
It stores the user's name, email, phone, and message.

## Contact Us Details

Change contact

HISTORY

Name:	<input type="text" value="Kanduri Jayanth Sri Ram"/>
Email:	<input type="text" value="jayanthsriram44@gmail.com"/>
Phone:	<input type="text" value="09490342221"/>
Message:	<input type="text" value="Testing 2.0"/>

In this way all the information given by the user (in Contact us form) is saved.

## Customer Model

Customer model in the database is shown as below,

Select customer to change

Action:   0 of 12 selected

<input type="checkbox"/>	CUSTOMER
<input type="checkbox"/>	bfgjhb
<input type="checkbox"/>	jayanth_sriram
<input type="checkbox"/>	webfeb
<input type="checkbox"/>	rakesh2
<input type="checkbox"/>	sdhvbsd

Every customer has a name, email, phone, door, landmark, pin code, city, state, profile pic, and an account created date. This data can be used in orders, shipping addresses, etc. One can update them on the account details page.

## Customer Details

Change customer

HISTORY

User:	rakesh2	 
Name:	rakesh2	
Email:	rakesh123@gmail.com	
Phone:	2365418	
Door:	89	
Landmark:	1lop	
Pincode:	896545	
City:	hyd	
State:	telangana	
Profile pic:	Currently: download_2_PFaFbtj.jpg	<input type="checkbox"/> Clear
	Change: <input type="button" value="Choose File"/>	No file chosen
Booksdonated:	0	

In this way all the information given by the user (in account details form) is saved.

## Donate Form Model

Donate model in the database is shown as below,

Select donate to change

Action:   0 of 6 selected

<input type="checkbox"/>	DONATE
<input type="checkbox"/>	Mullagura
<input type="checkbox"/>	Anirudh
<input type="checkbox"/>	Kanduri Jayanth
<input type="checkbox"/>	dfvd
<input type="checkbox"/>	Rakesh

- ❖ The Donate model has fields like the first name, last name, email, phone, books, door number, landmark, city, state, and pin code of the donor.
- ❖ The admin can get a detailed view of these details and can send the representative to the donor and bring the books to the warehouse.

## Donate Form Details

Change donate

HISTORY

Fstnm:	Mullagura
Lsnm:	Harish
Email:	harishmullagura@gmail.com
Phone:	8688341
Books:	4
Doorno:	12
Landmark:	near KIMS Hospital
City:	Anantapur
State:	Andhra Pradesh
Pincode:	515001

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)

In this way all the information given by the user (in donate form) is saved.

## Orders

Select order to change

ADD ORDER +

Action: -----  0 of 40 selected

ORDER

81 - Neeraj - with transaction id - None

80 - Neeraj - with transaction id - 1620468741.947368

79 - Neeraj - with transaction id - 1620369369.122088

78 - Neeraj - with transaction id - 1620367591.816092

77 - Neeraj - with transaction id - 1620366417.104622

Here, the orders of users are displayed with Order ID, user name, and Transaction ID.

## Order Details

Change order

HISTORY

Customer:    

Product:    

Status:

Note:

Complete

Transaction id:

In this section, the details of the customer such as name, product name, the status of the delivery, and Transaction ID.

## Each Order Model

Order item model in the database is shown as below,

Select order item to change

[ADD ORDER ITEM +](#)

Action:   0 of 85 selected

<input type="checkbox"/> ORDER ITEM
<input type="checkbox"/> 80) Name - Neeraj, Book - Power Electronics, Quantity-1
<input type="checkbox"/> 80) Name - Neeraj, Book - Charaka Samhitha, Quantity-1
<input type="checkbox"/> 80) Name - Neeraj, Book - Algorithm Design, Quantity-2
<input type="checkbox"/> 79) Name - Neeraj, Book - Basic Structural analysis, Quantity-1
<input type="checkbox"/> 79) Name - Neeraj, Book - Wild Remedies, Quantity-1
<input type="checkbox"/> 79) Name - Neeraj, Book - Statistics, Quantity-1
<input type="checkbox"/> 79) Name - Neeraj, Book - Algorithm Design, Quantity-1
<input type="checkbox"/> 79) Name - Neeraj, Book - Clean Code, Quantity-2

Order item model has fields like product, order, quantity, date\_added. Admin can get the whole details of the order and the customer who ordered and the product attached as foreign key to this class.

## Each Order Model

Change order item

[HISTORY](#)

Product:    

Order:    

Quantity:

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)

In this way, all the information was given by the user (in the order form) is saved.

## Product Model

Product model in the database is shown as below,

Select product to change

Action: -----  0 of 17 selected

<input type="checkbox"/> PRODUCT
<input type="checkbox"/> Machine Learning
<input type="checkbox"/> Cure Tooth Decay
<input type="checkbox"/> Essential Pediatrics
<input type="checkbox"/> The Economics Book
<input type="checkbox"/> Why Nations Fail
<input type="checkbox"/> Mathematics
<input type="checkbox"/> Poor Economics
<input type="checkbox"/> Physics(XI)

The product model has fields like condition, tags, image, name, price, author, publisher, ISBN, description, date\_created, categories, digital, product\_count. This model can be used for admin to get information about products. Image URL will return the image section in each book detail.

## Each Product Model

Change product

HISTORY

Image: Currently: mitchell\_lwprvYQ.png  Change:  No file chosen

Name:

Price:

Author:

Publisher:

Isbn:

Condition:

Description:

Categories:

Digital:

Product count:

In this way all the information given by the user (in product form) is saved.

## Shipping Address Model

Shipping address model in the database is shown as below,

Select shipping address to change

Action:   0 of 30 selected

- SHIPPING ADDRESS
- 80 - - with transaction id - 1620468741.947368
- 79 - Hyderabad - with transaction id - 1620369369.122088
- 78 - Hyderabad - with transaction id - 1620367591.816092

## Shipping Address Model

Change shipping address

HISTORY

Customer:  ▼ ✖ ✚ ✎

Order:  ▼ ✖ ✚ ✎

Address:

City:

State:

Zipcode:

Delete Save and add another Save and continue editing SAVE

The shipping address model has fields like a customer, order, address, city, state, zip code, data\_added. This data can be used for the admin to get the information about shipping addresses given by customers.

## Groups Authentication

Here, there are 2 groups. Admin and customer who has different authentication.

Select group to change

ADD GROUP +

Search:  Search

Action:   0 of 2 selected

GROUP

admin

customer

2 groups

## Users Authorization

Here are the users who have created their accounts on the website and their staff status can be seen here.

Select user to change

ADD USER +

SEARCH  Search

Action:  Go 0 of 12 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Sam		Sam		<span style="color: red;">X</span>
<input type="checkbox"/>	anirudh_jak	anirudh.jakhotia6@gmail.com			<span style="color: red;">X</span>
<input type="checkbox"/>	bfgjhb	sdjhfb@hjsdvc.com			<span style="color: red;">X</span>
<input type="checkbox"/>	jayanth_k	kandurisriram99@gmail.com			<span style="color: red;">X</span>
<input type="checkbox"/>	jayanth_sriram	lol@gmail.com			<span style="color: red;">X</span>
<input type="checkbox"/>	neejjd	akhdsuf@bd.com			<span style="color: red;">X</span>
<input type="checkbox"/>	neerajdusa	neerajdusa@gmail.com	Neeraj	Dusa	<span style="color: green;">✓</span>

FILTER

By staff status

- [All](#)
- [Yes](#)
- [No](#)

---

By superuser status

- [All](#)
- [Yes](#)
- [No](#)

---

By active

- [All](#)
- [Yes](#)
- [No](#)

## Each User Authorization

Here each user details can be seen when a user signs up for their account with their personal details.

Change user

HISTORY

Username:  Required: 150 characters or fewer. Letters, digits and @/./~/\_ only.

Password:  algorithm: pbkdf2\_sha256 iterations: 216000 salt: ByPBEF\*\*\*\*\* hash: IZKflt\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

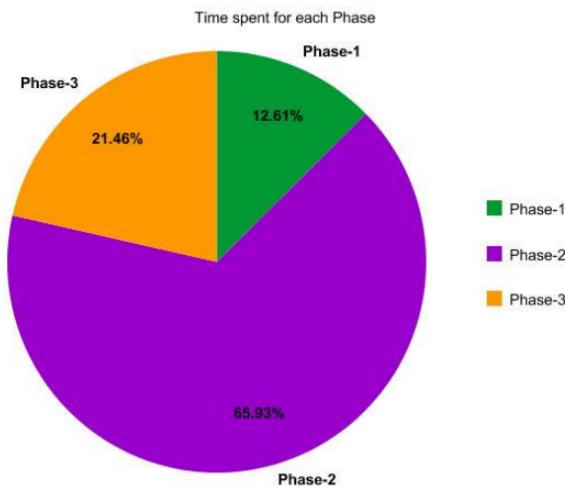
**Personal info**

First name:	<input type="text" value="Neeraj"/>
Last name:	<input type="text" value="Dusa"/>
Email address:	<input type="text" value="neerajdusa@gmail.com"/>

**Permissions**

- Active  
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.
- Staff status  
Designates whether the user can log into this admin site.
- Superuser status  
Designates that this user has all permissions without explicitly assigning them.

# 10. Project phase Schedule



The deliverables for all three phases are listed below:

## Phase I

- Vision Document 1.0
- Project Plan 1.0
- Software Quality Assurance Plan

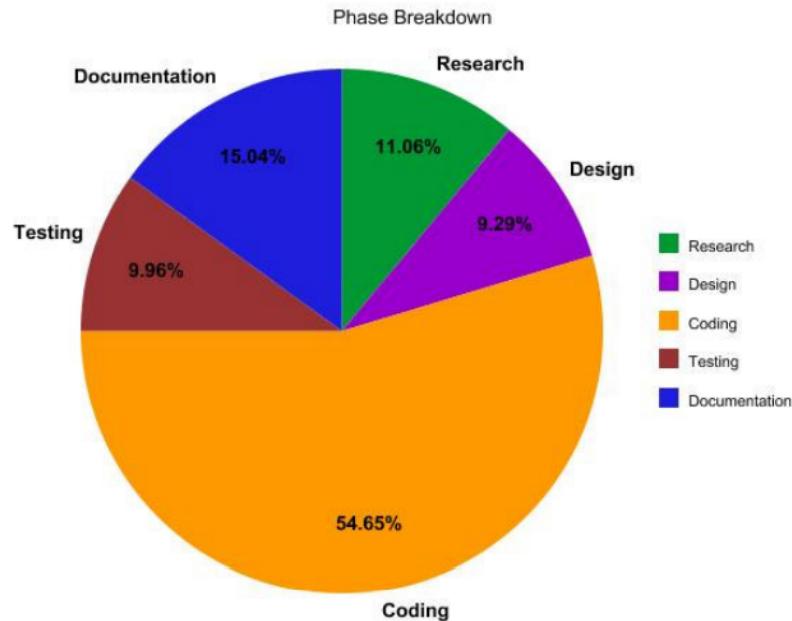
## Phase II

- Action Items –identified during phase I
- Vision Document 2.0
- Project Plan 2.0
- Formal Requirements Specification
- Architecture Design
- Test Plan
- Formal Technical Inspection
- Executable Architecture Prototype

## Phase III

- Action Items - identified during phase II
- User Manual & Source Code
- Component Design
- Assessment Evaluation
- Project Evaluation
- References & Formal Technical Inspection

# Time allocated for various tasks



	Inception (hours)	Implementation (hours)	Elaboration (hours)	Total
Research	15	19	6	40
Design	18	30	10	58
Coding	2	45	30	77
Testing	5	10	40	55
Documentation	2	18	10	30
<b>Total</b>	<b>42</b>	<b>122</b>	<b>96</b>	<b>260</b>

# REFERENCES

- <https://www.w3schools.com/html/default.asp>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://stackoverflow.com>
- <https://docs.djangoproject.com/en/3.2/>
- [https://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language)
- [https://www.visual-paradigm.com/guide/software-development-pr  
ocess/what-is-a-software-process-model/](https://www.visual-paradigm.com/guide/software-development-process/what-is-a-software-process-model/)
- [https://en.wikipedia.org/wiki/Unit\\_testing](https://en.wikipedia.org/wiki/Unit_testing)

# TEAM MEMBERS

*Neeraj Dusa*

*Rakesh Ganeshula*

*Harish Mullagura*

*Anirudh Jakhotia*

*Jayanth Sriram Kanduri*

