

# Final Project Documentation

## Project Prerequisites

### Languages used for our project

- Html
- Css
- BootStrap
- Django
- Ajax
- Json

### Project features

- For Admin
- Add books
- Remove books
- Update books
- View all books
- Update stock status and availability
- Issue books

### For students

- View books
- Select books
- Add student profile
- Edit profile

### How to run the code?

- Save the file on the desktop
- Open cmd and change the directory to the file with your code
- Run code in a virtual environment using - `venv \Scripts\activate`
- Run the server using `> py manae.py runserver`

## Project File Structure

Models.py

```
store > settings.py views.py urls.py detail.html base.html basket.py summary.html
store > models.py > Category > get_absolute_url
1  from django.db import models
2  from django.contrib.auth.models import User
3  from django.urls import reverse
4
5
6
7  # Create your models here.
8  class Category(models.Model):
9      name = models.CharField(max_length=255, db_index=True)
10     slug = models.SlugField(max_length=255, unique=True)
11     class meta:
12         verbose_name_plural = 'categories'
13
14     def get_absolute_url(self):
15         return reverse('store:category_list', args=[self.slug])
16
17     def __str__(self):
18         return self.name
19
20 class Books(models.Model):
21     category = models.ForeignKey(Category, related_name='books', on_delete=models.CASCADE)
22     created_by = models.ForeignKey(User, on_delete=models.CASCADE, related_name='product_creator')
23     title = models.CharField(max_length=255)
24     author = models.CharField(max_length=255)
25     isbn = models.CharField(max_length=255)
26     copies = models.IntegerField(null=True)
27     image = models.ImageField(upload_to='images/')
28     slug = models.SlugField(max_length=255)
29     in_stock = models.BooleanField(default=True)
30     is_active = models.BooleanField(default=True)
31     created = models.DateTimeField(auto_now_add=True)
32     updated = models.DateTimeField(auto_now_add=True)
33
```

## Code Explanation:

Category model:

where we shall store our genre this links to our books in the database

Book model:

Adds and stores our books to our Database

Student Model:

Adds students to the Database, updates profiles, and selects books

Transaction Model:

Tracks the books issued to students, it issues, returns, and removes the books from the transaction list.

## URLs.py

```
settings.py  views.py  urls.py  X  detail.html  base.html  basket.py
basket > urls.py > ...
1  from django.urls import path
2
3  from . import views
4
5  app_name = 'basket'
6  urlpatterns = [
7      path('', views.basket_summary, name='basket_summary'),
8      path('add/', views.basket_add, name='basket_add'),
9  ]
```

```
settings.py  views.py  urls.py basket  urls.py store X  detail.html  base.html  basket.py
store > urls.py > ...
1  from django.urls import path
2  from . import views
3
4
5
6  app_name = 'store'
7
8  urlpatterns = [
9      path('', views.all_books, name='all_books'),
10     path('item/<slug:slug>/', views.book_details, name='book'),
11     path('categories/<slug:slug>/', views.category_list, name='category_list'),
12
13
14 ]
15
```

```
settings.py  views.py  urls.py basket  urls.py DRK2 X  detail.html  base.html  basket.py
DRK2 > urls.py > ...
11  """
12  1. Add an import: from other_app.views import home
13  2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
14  Including another URLconf
15  1. Import the include() function: from django.urls import include, path
16  2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
17  """
18  from django.contrib import admin
19  from django.urls import path, include
20  from django.conf import settings
21  from django.conf.urls.static import static
22  from django.urls import include, path
23
24  urlpatterns = [
25      path('admin/', admin.site.urls),
26      path('', include('store.urls', namespace='store')),
27      path('basket/', include('basket.urls', namespace='basket')),
28  ]
29
30  if settings.DEBUG:
31      urlpatterns += static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
```

## Code Explanation:

### Basket URL

It allows users to select books and add them to the cart.

The basket will also display the list of books chosen by the student.

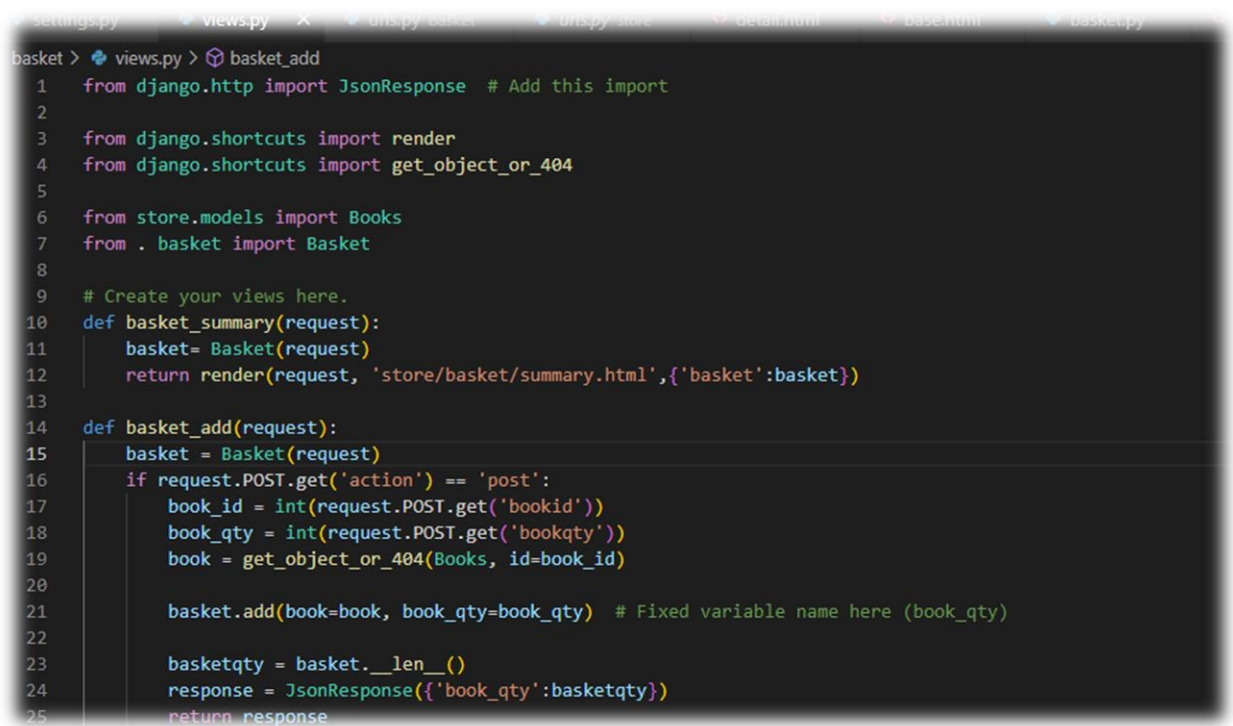
### Store URL

Fetches books from the database using the category/genre as the linkage (slug) to the books

### Main file URL

Links to admin login and database

## Views.py



```
1 from django.http import JsonResponse # Add this import
2
3 from django.shortcuts import render
4 from django.shortcuts import get_object_or_404
5
6 from store.models import Books
7 from . basket import Basket
8
9 # Create your views here.
10 def basket_summary(request):
11     basket= Basket(request)
12     return render(request, 'store/basket/summary.html',{'basket':basket})
13
14 def basket_add(request):
15     basket = Basket(request)
16     if request.POST.get('action') == 'post':
17         book_id = int(request.POST.get('bookid'))
18         book_qty = int(request.POST.get('bookqty'))
19         book = get_object_or_404(Books, id=book_id)
20
21         basket.add(book=book, book_qty=book_qty) # Fixed variable name here (book_qty)
22
23         basketqty = basket.__len__()
24         response = JsonResponse({'book_qty':basketqty})
25         return response
```

### View.basket\_add

Adds books and book quantities selected and added to the cart

Posts the quantity in our html page

```

store > views.py > ...
1 from django.shortcuts import get_object_or_404 ,render
2
3 from .models import *
4
5 # list of categories
6 def categories(request):
7     return {
8         'categories':Category.objects.all()
9     }
10
11 # Create your views here.
12 def all_books(request):
13     my_books = Books.objects.all()
14     return render(request,'store/home.html',{'my_books': my_books})
15
16 #create single page with book details
17 def book_details(request, slug):
18     book = get_object_or_404 (Books, slug=slug , in_stock=True)
19     return render(request,'store/products/detail.html', {'book': book})
20
21 # category_list():
22 def category_list(request, category_slug):
23     category = get_object_or_404(Category, slug=category_slug)
24     my_books = Books.objects.filter(category=category)
25     return render(request, 'store/products/category.html', {'category': category, 'my_books': my_books})

```

```

settings.py  views.py basket  views.py store  urls.py  detail.html  base.html X  basket.py  summary.html
templates > store > base.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>{% block title %}My Library{% endblock %}</title>
7     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-17E8VVD" sha="
8     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD" sha="
9     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.min.js" integrity="sha384-Rx+T1V2Gupg4B" sha="
10    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
11 </head>
12 <body>
13     <header>
14         <nav class="navbar navbar-expand-md navbar-dark bg-dark">
15             <div class="container-fluid">
16                 <a class="navbar-brand" href="#">Library</a>
17                 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
18                     <span class="navbar-toggler-icon"></span>
19                 </button>
20                 <div class="collapse navbar-collapse" id="navbarCollapse">
21                     <ul class="navbar-nav me-auto mb-2 mb-md-0">
22                         <li class="nav-item">
23                             <a class="nav-link active" aria-current="page" href="#">Home</a>
24                         </li>
25                         <li class="nav-item">
26                             <a class="nav-link" href="#">Link</a>
27                         </li>
28                         <li class="nav-item dropdown">
29                             <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown">
30                                 <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
31                                     <li><a class="dropdown-item" href="#">All</a></li>
32                                     <li><a class="dropdown-item" href="{% categories.get_absolute_url %}">{% c.name|title %}</a>

```

View.store.py

Displays books on our home page.

Shows all books we have in stock, and books not in stock will not display once clicked on.

The category list displays books that belong to that category.

## THE HTML CODE

### Base.html

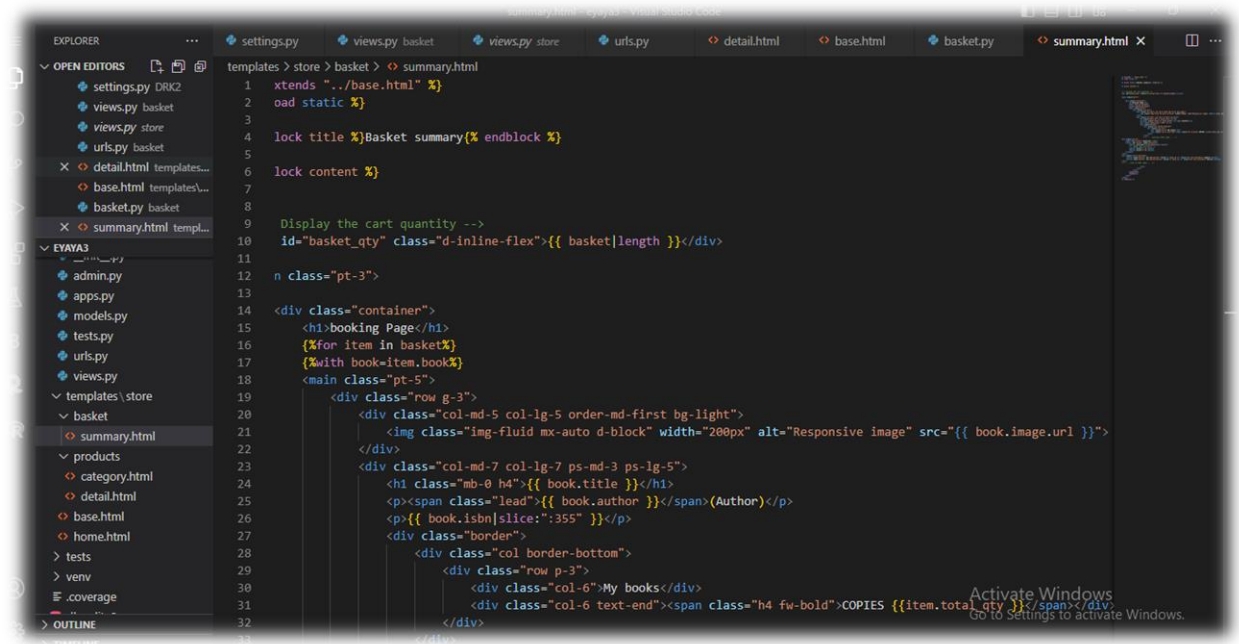
Helps us avoid duplicating our codes and creates a single template for all other pages have to connect to.

Also allows our coding to be more organized.

```
templates > store > base.html
25 <li class="nav-item">
26   <a class="nav-link" href="#">Link</a>
27 </li>
28 <li class="nav-item dropdown">
29   <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown"
30     <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
31     <li><a class="dropdown-item" href="#">All</a></li>
32     {% for c in categories %}
33     <li><a class="dropdown-item" href="{{ categories.get_absolute_url }}">{{ c.name|title }}
34     {% endfor %}
35   </ul>
36 </li>
37 </ul>
38 <a type="button" role="button" href="{% url 'basket:basket_summary' %}" class="btn btn-outline-secondary">
39
40   <div id="basket_qty" class="d-inline-flex">
41     {% with total_qty=basket|length %}
42     {% if total_qty > 0 %}
43       {{ total_qty }}
44     {% else %}
45       0
46     {% endif %}
47     {% endwith %}
48   </div>
49
50   <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-basket">
51     <path d="M0 2.5A5.5 0 0 1 .5 2H2a.5.5 0 0 1 .485.379L2.89 4H14.5a.5.5 0 0 1 .485.621l-1.5 6" data-bbox="0 0 0 0"/>
52   </svg>
53   Basket
54 </a>
55 </div>
56 </div>
```

### Summary HTML





Here books we selected will appear as a list.

Single book details here are details.html shows the image, book author, title, number of copies

```

templates > store > products > detail.html
35         </div>
36     </div>
37 </main>
38
39 </div>
40 <script>
41     $(document).on('click', '#add-button', function (e){
42         e.preventDefault(); // 'preventDefault' should start with a lowercase 'p'
43         $.ajax({
44             type: 'POST',
45             url: '{% url "basket:basket_add" %}',
46             data: {
47                 bookid: $('#add-button').val(),
48                 bookqty: $('#select option:selected').text(),
49                 csrfmiddlewaretoken: '{{ csrf_token }}',
50                 action: 'post'
51             },
52             success: function (json) { // Corrected 'success' spelling
53                 // Handle success here
54                 //console.log(json)
55                 document.getElementById("basket_qty").innerHTML = json.book_qty
56             },
57             error: function (xhr, errmsg, err) { // Corrected 'error' spelling
58                 // Handle error here
59             }
60         });
61     });
62 </script>
63 {% endblock %}

```

```

templates > store > products > detail.html
1 {% extends '../base.html' %}
2
3 {% block content %}
4     <div class="container">
5         <main class="pt-5">
6             <div class="row g-3">
7                 <div class="col-md-5 col-lg-5 order-md-first bg-light">
8                     
9                 </div>
10                <div class="col-md-7 col-lg-7 ps-md-3 ps-lg-5">
11                    <h1 class="mb-0 h4">{{ book.title }}</h1>
12                    <p><span class="lead">{{ book.author }}</span> (Author)</p>
13                    <p>{{ book.isbn|slice:"355" }}</p>
14                    <div class="border">
15                        <div class="col border-bottom">
16                            <div class="row p-3">
17                                <div class="col-6">Available books</div>
18                                <div class="col-6 text-end"><span class="h4 fw-bold">COPIES {{ book.copies }}</span></div>
19                            </div>
20                        </div>
21                        <!-- ... (previous HTML code) ... -->
22                    <div class="col-6">
23                        <label for="select">Quantity</label>
24                        <select id="select" name="quantity"> <!-- Added name attribute -->
25                            <option value="1">1</option>
26                            <option value="2">2</option>
27                        </select>
28                    </div>
29                    <div class="col-6 text-end">
30                        <button type="button" id="add-button" value="{{ book.id }}" class="btn btn-secondary">Add to Basket</button>
31                    </div>
32                </div>
33            </div>
34            <!-- rest of HTML code -->
35        </main>
36    </div>
37 </div>
38
39 </div>
40 <script>
41     $(document).on('click', '#add-button', function (e){
42         e.preventDefault(); // 'preventDefault' should start with a lowercase 'p'
43         $.ajax({
44             type: 'POST',
45             url: '{% url "basket:basket_add" %}',
46             data: {
47                 bookid: $('#add-button').val(),
48                 bookqty: $('#select option:selected').text(),
49                 csrfmiddlewaretoken: '{{ csrf_token }}',
50                 action: 'post'
51             },
52             success: function (json) { // Corrected 'success' spelling
53                 // Handle success here
54                 //console.log(json)
55                 document.getElementById("basket_qty").innerHTML = json.book_qty
56             },
57             error: function (xhr, errmsg, err) { // Corrected 'error' spelling
58                 // Handle error here
59             }
60         });
61     });
62 </script>
63 {% endblock %}

```

From our main menu, once you click a book, it takes you to that individual book and shows you all details about it.

In summary, with Django, we have been able to develop our Project to add books to our database, display them on our home page, select book of interest add to the list.