

Name = Pranjal Keshav Patil

Roll No. = 45

Q. 1. Develop a RESTful API for a Library Management System using Django REST Framework (DRF). The system should manage books and authors with basic CRUD functionality.

1. Project Setup

Install Django

Start a new Django project : LibraryManagementSystem

Create an app 1 : library

Add to INSTALLED_APPS in settings.py

2. Model Creation : library/models.py

```
In [ ]: from django.db import models

# Create your models here.

class Book(models.Model):
    title = models.CharField(max_length=200)
    isbn = models.CharField(max_length=13, unique=True)

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

3. Serializer : library/serializers.py

```
In [ ]: from rest_framework import serializers
        from .models import Book

class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = '__all__'
```

4. Views : library/views.py

```
In [ ]: from rest_framework import generics
        from .models import Book
        from .serializers import BookSerializer
        from django.http import HttpResponse

def home(request):
    return HttpResponse("Welcome to the Library Management System!")

# Book Views
class BookListCreateView(generics.ListCreateAPIView):
    queryset = Book.objects.all()
    serializer_class = BookSerializer

class BookDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Book.objects.all()
    serializer_class = BookSerializer
```

5. URL Configuration : library/urls.py

```
In [ ]: from django.urls import path
        from .views import BookListCreateView, BookDetailView

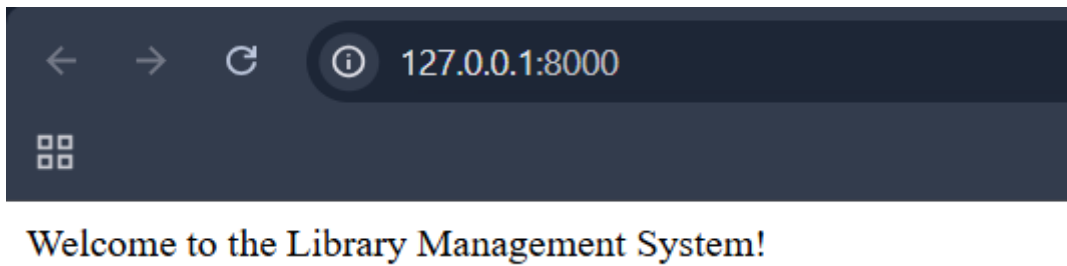
urlpatterns = [
    # Book Endpoints
    path('books/', BookListCreateView.as_view(), name='book-list-create'),
    path('books/<int:pk>/', BookDetailView.as_view(), name='book-detail'),
]
```

URL Configuration : LibraryManagementSystem/urls.py

```
In [ ]: from django.contrib import admin
        from django.urls import path, include
        from library.views import home

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('library.urls')),
    path('', home),
]
```

Output :



CRUD Functionality:

- List all books (GET) or create a book (POST).

Django REST framework pranjal

Book List Create

OPTIONS GET

GET /api/books/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 4,
    "title": "The Wonder That Was India",
    "isbn": "033043909X"
  },
  {
    "id": 5,
    "title": "Midnight's Children",
    "isbn": "978-0-224-018"
  },
  {
    "id": 6,
    "title": "Train to Pakistan",
    "isbn": "978-0-14-3065"
  },
  {
    "id": 7,
    "title": "The White Tiger",
    "isbn": "978-1-4165-62"
  }
]
```

Raw data HTML form

Title

Isbn

POST

- Retrieve, update, or delete book with ID 5 (GET, PUT, DELETE).

Django REST framework pranjal

[Book List Create](#) / [Book Detail](#)

Book Detail

[DELETE](#) [OPTIONS](#) [GET](#)

GET /api/books/5/?format=api

```

HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 5,
  "title": "Midnight's Children",
  "isbn": "978-0-224-018"
}

```

[Raw data](#) [HTML form](#)

Title

Isbn

[PUT](#)

- <http://127.0.0.1:8000/admin/>:

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add Change
Users	+ Add Change

Recent actions

My actions

- + The White Tiger
Book
- + Train to Pakistan
Book
- + Midnight's Children
Book
- + The Wonder That Was India
Book
- ✗ Badalata Bharat: Paratantryatun
Mahasattekade
Book
- + Badalata Bharat: Paratantryatun
Mahasattekade
Book