Sumit Patel - 40

Assignment 04 - Python Integration Primer

Models.py

```
from django.db import models

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

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

serializers.py

```
from rest_framework import serializers
from .models import Book

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

views.py

```
from django.shortcuts import render

# Create your views here.
from rest_framework import generics
from .models import Book
from .serializers import BookSerializer

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

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

urls.py (inside library/)

```
from django.urls import path
from .views import BookListCreateView, BookDetailView

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

settings.py (inside -> Lib_management/)

```
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-622(3wi0slycowb+#*hu07sgw3s7py7e%&g^+w)i-ud-j8)f-='

DEBUG = True

ALLOWED_HOSTS = []


INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'library',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'library_management.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
WSGI_APPLICATION = 'library_management.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

manage.py

```
"""Django's command-line utility for administrative tasks."""
import os
import sys
```

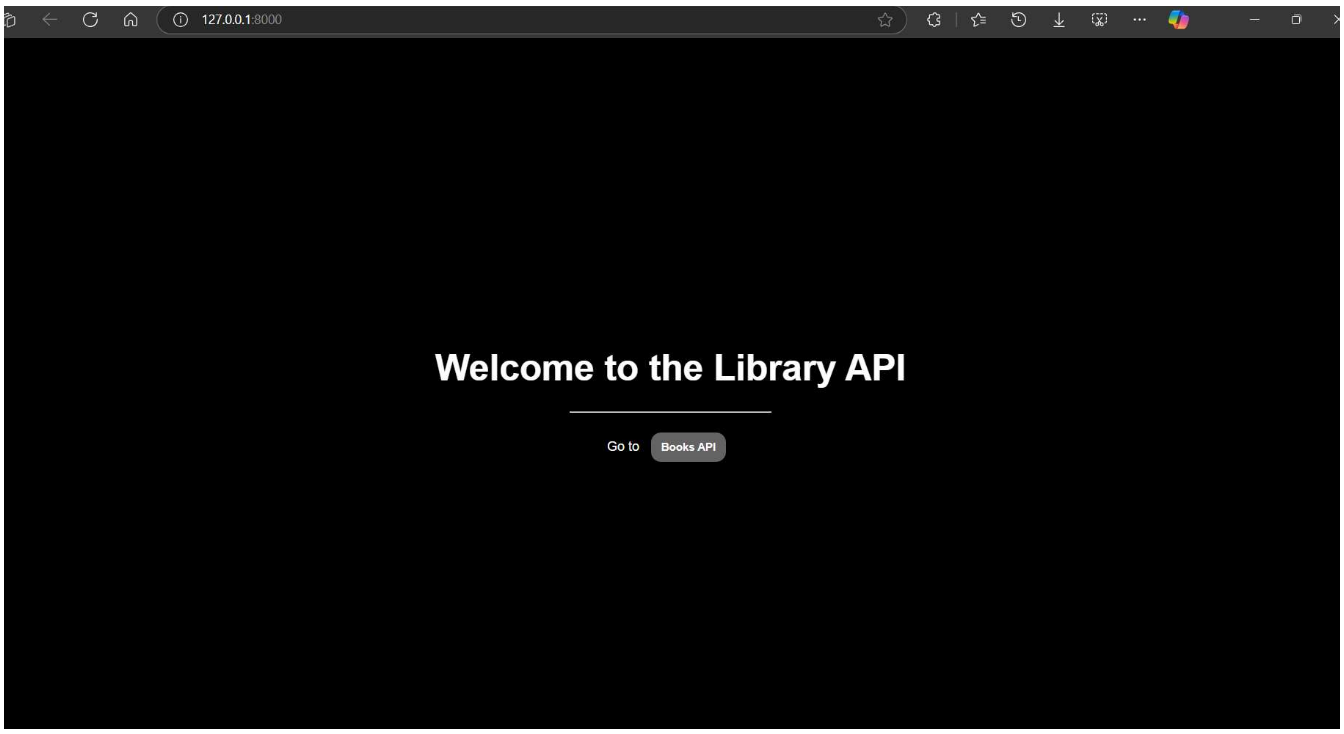
```
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'library_management.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

url.py (inside -> Lib_management/)

```
from django.contrib import admin
from django.urls import path, include
from django.http import HttpResponse

def home(request):
    return HttpResponse("""
       <html>
            <head>
                <title>Library API</title>
                <style>
                    body {
                        font-family: Arial, sans-serif;
                        background-color: #000000;
                    }

                    .container {
                        height: 100vh;
                        display: flex;
                        justify-content: center;
                        align-items: center;
                        flex-direction: column;
                    }

                    h1{
                        font-size: 44px;
                        color: #ffffff;
                        margin-bottom: 20px;
                    }

                    hr {
                        color: #ffffff;
                        width: 15%;
                        border-radius: 5px;
                    }

                    p{
                        color: #ffffff;
                    }

                    .B {
                        width: 90px;
                        height: 35px;
                        margin-left: 10px;
                        margin-right: 10px;
                        border: none;
                        background-color: #646464;
                        border-radius: 12px;
                        transition: 0.3s ease-in-out, color 0.3s ease-in-out;
                        cursor: pointer;
                    }
                    .B:hover{
                        transform: scale(1.05);
                        color: #3498db !important;
                    }

                   a {
                        position: relative;
                        color: #ffffff;
                        text-decoration: none;
                        font-weight: bold;
                        cursor: pointer;
                        transition: color 0.3s ease-in-out;
                    }

                    /* Underline effect */
                    a::after {
                        content: "";
                        position: absolute;
                        left: 0;
                        bottom: -2px; /* Adjust this to control the underline position */
                        width: 0;
                        height: 1.5px;
                        border-radius: 5px;
                        background-color: #3498db;
                        transition: width 0.3s ease-in-out;
                    }

                    /* Expand underline on hover */
                    a:hover {
                        color: #3498db;
                    }

                    a:hover::after {
                        width: 100%;
                    }

                </style>
            </head>
            <body>
                <div class="container">
                    <h1>Welcome to the Library API</h1>
                    <hr>
                    <p>Go to
                        <button class="B">
                            <a href='/api/books/'>Books API</a>
                        </button>
                    </p>
                </div>
            </body>
        </html>

    """)
urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('library.urls')),
    path('', home),  # Set a homepage
]
```

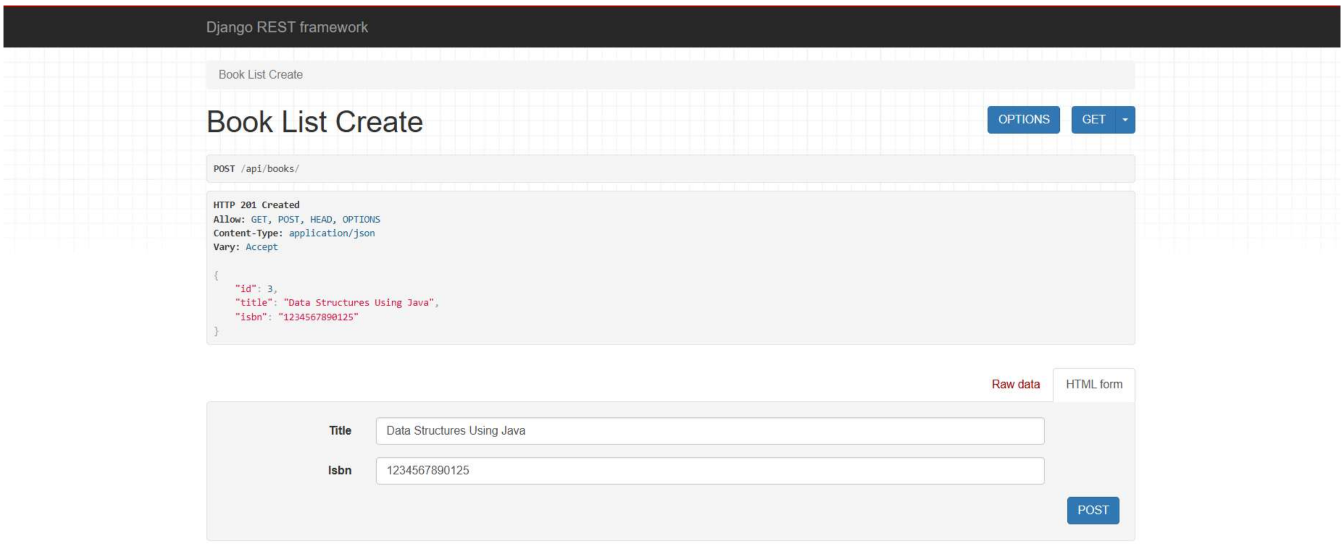
Output

The Frontend



- When Click on the Books Api Button Link redirects to this page.

After Adding title and isbn and Posting the page looks like this



Book List Create

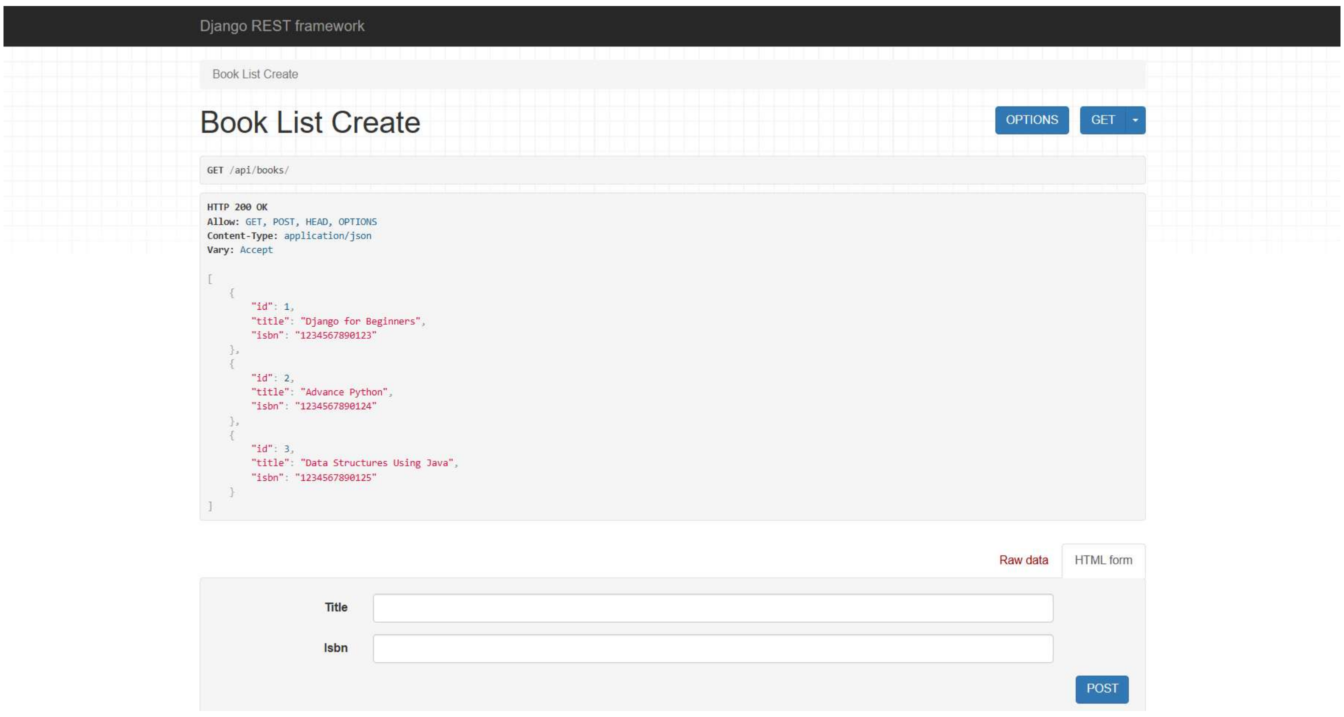
```
POST /api/books/
```

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 3,
    "title": "Data Structures Using Java",
    "isbn": "1234567890125"
}
```

Title: Data Structures Using Java

Isbn: 1234567890125

- when click on get it redirects to the list of pages



Book List Create

```
GET /api/books/
```

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

[
    {
        "id": 1,
        "title": "Django for Beginners",
        "isbn": "1234567890123"
    },
    {
        "id": 2,
        "title": "Advance Python",
        "isbn": "1234567890124"
    },
    {
        "id": 3,
        "title": "Data Structures Using Java",
        "isbn": "1234567890125"
    }
]
```

Title:

Isbn: