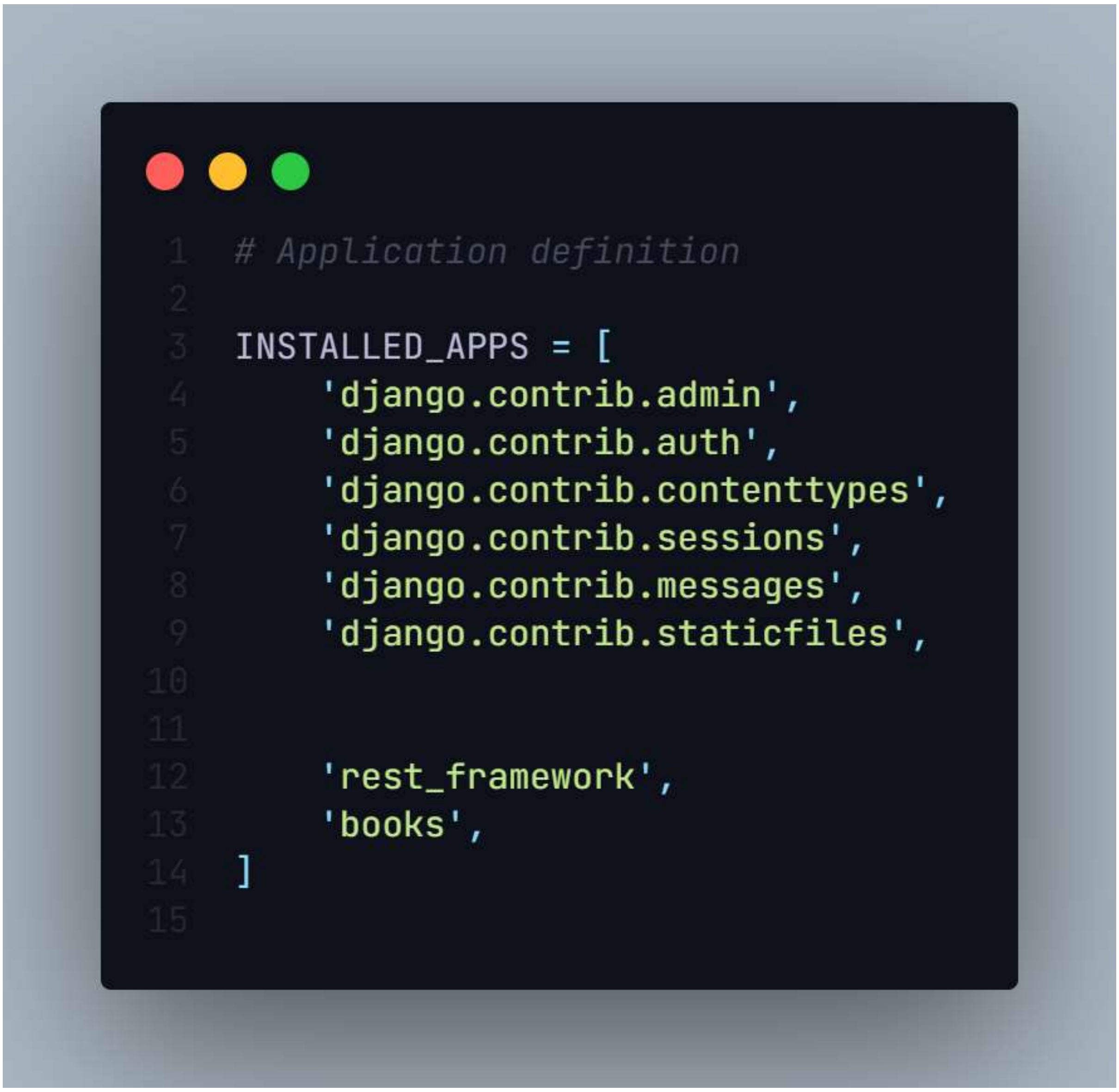
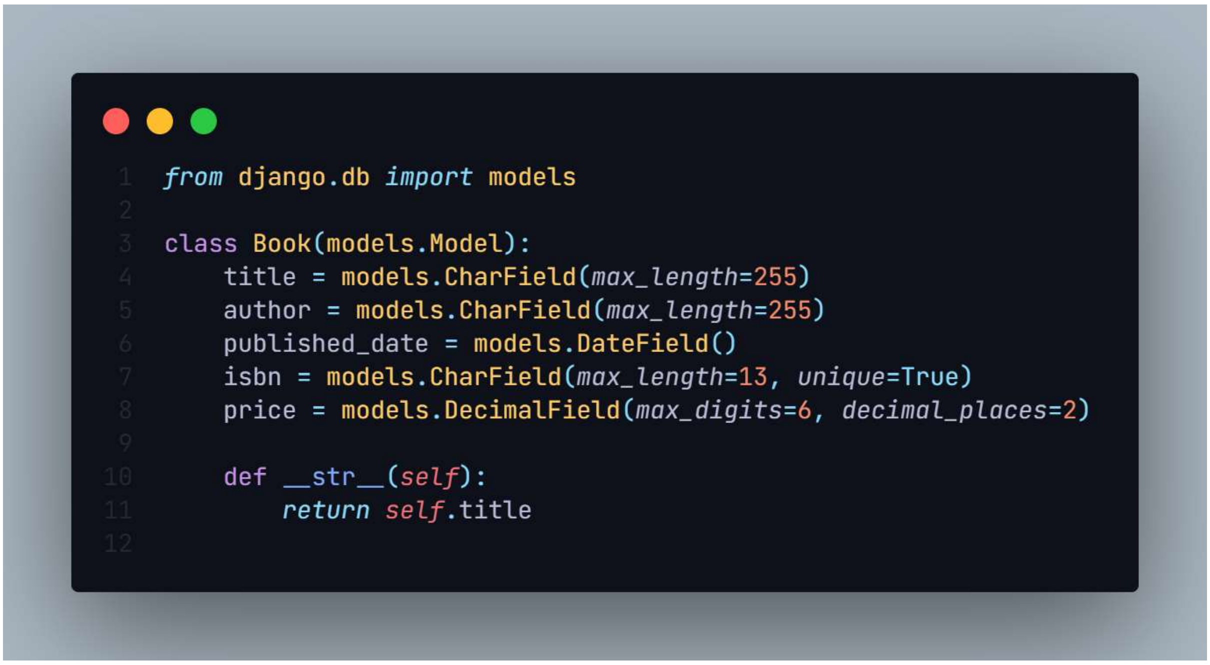
Settings.py



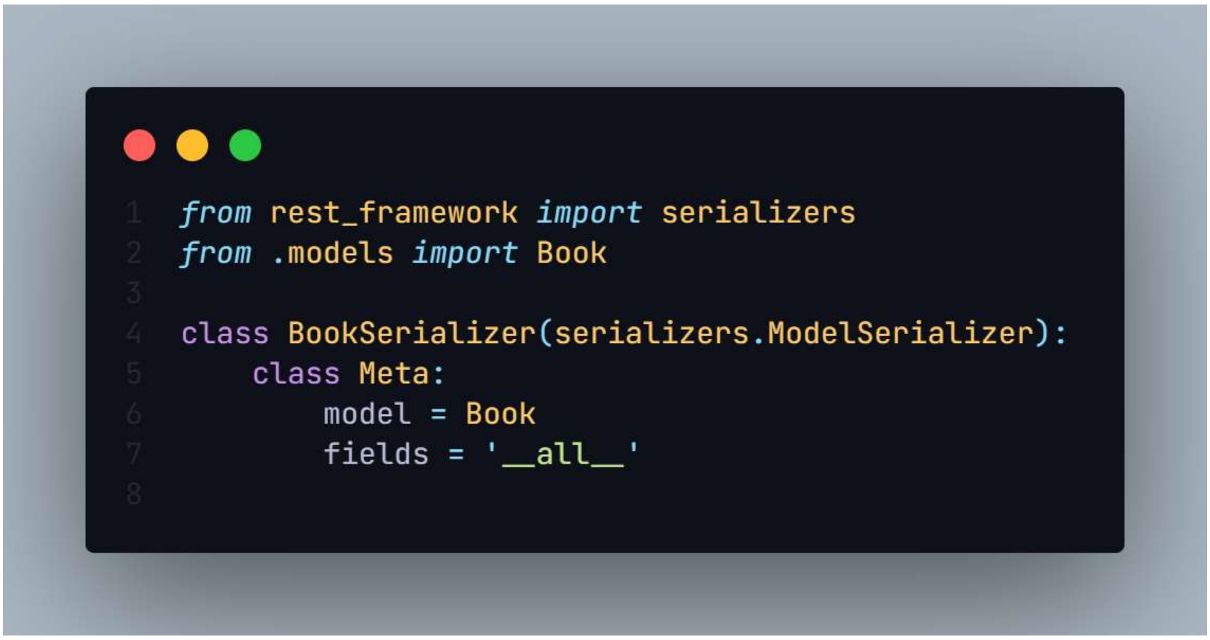
```python
# Application definition

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


    'rest_framework',
    'books',
]

```

Models.py



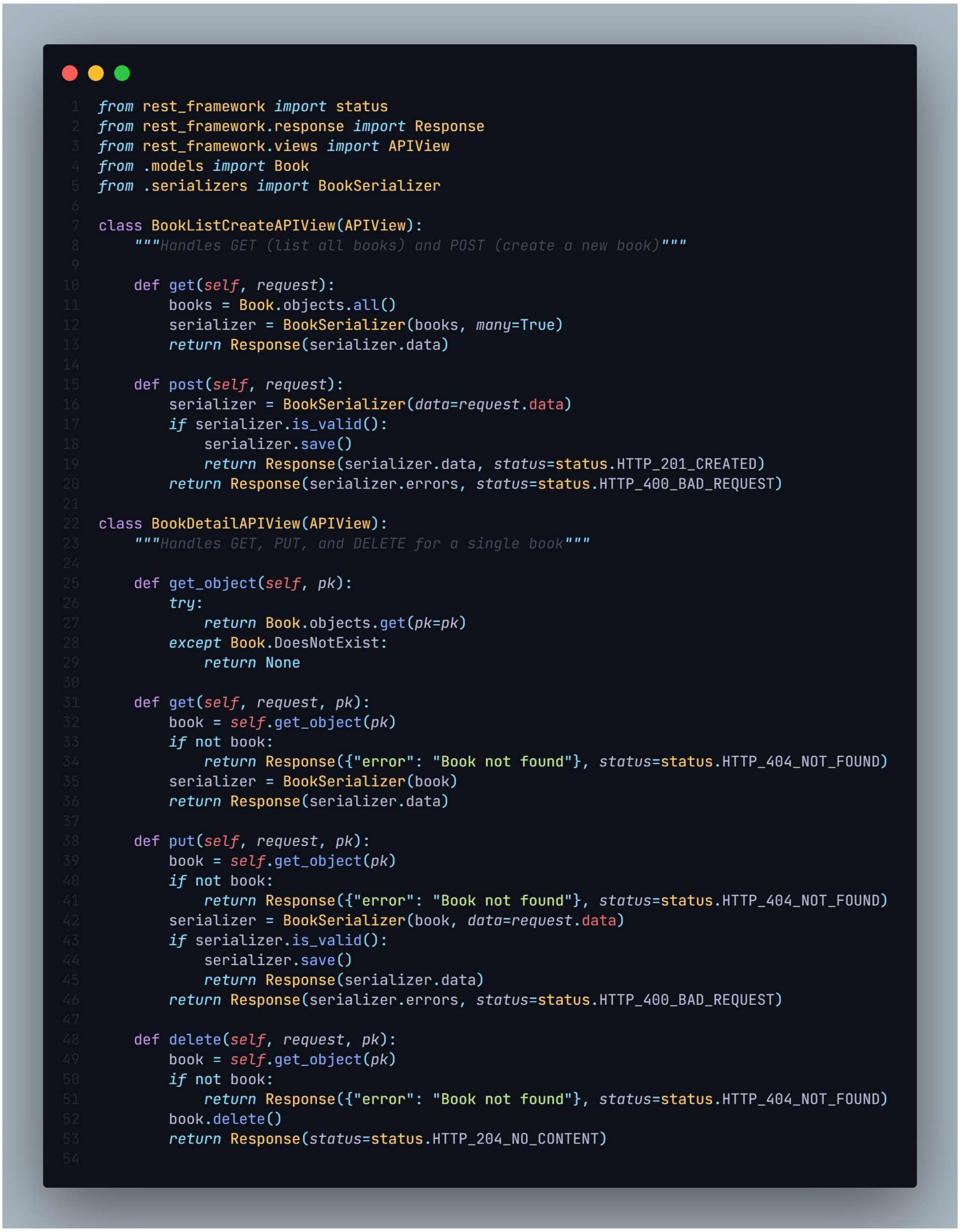
```python
from django.db import models

class Book(models.Model):
    title = models.CharField(max_length=255)
    author = models.CharField(max_length=255)
    published_date = models.DateField()
    isbn = models.CharField(max_length=13, unique=True)
    price = models.DecimalField(max_digits=6, decimal_places=2)

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

serializers.py



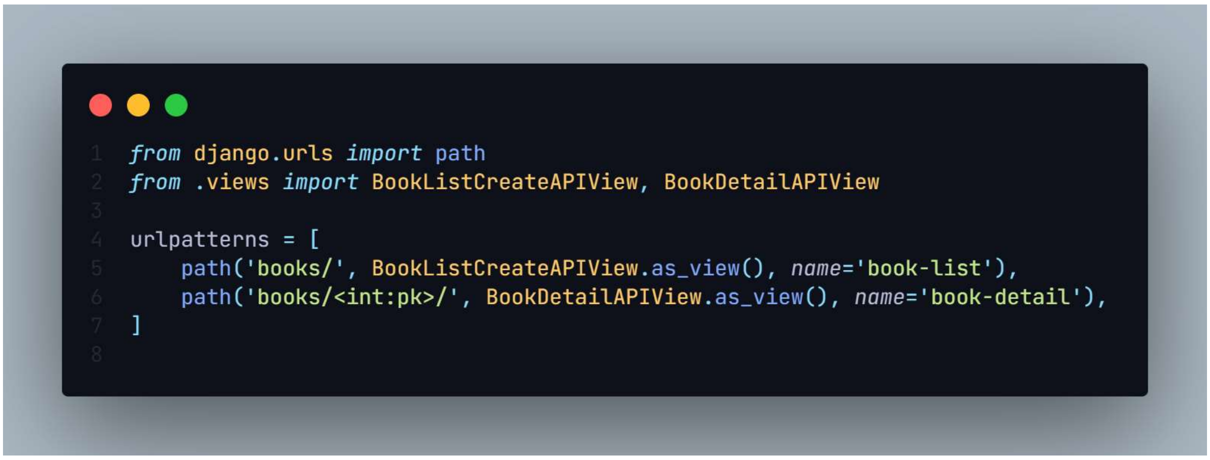
```python
from rest_framework import serializers
from .models import Book

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

views.py



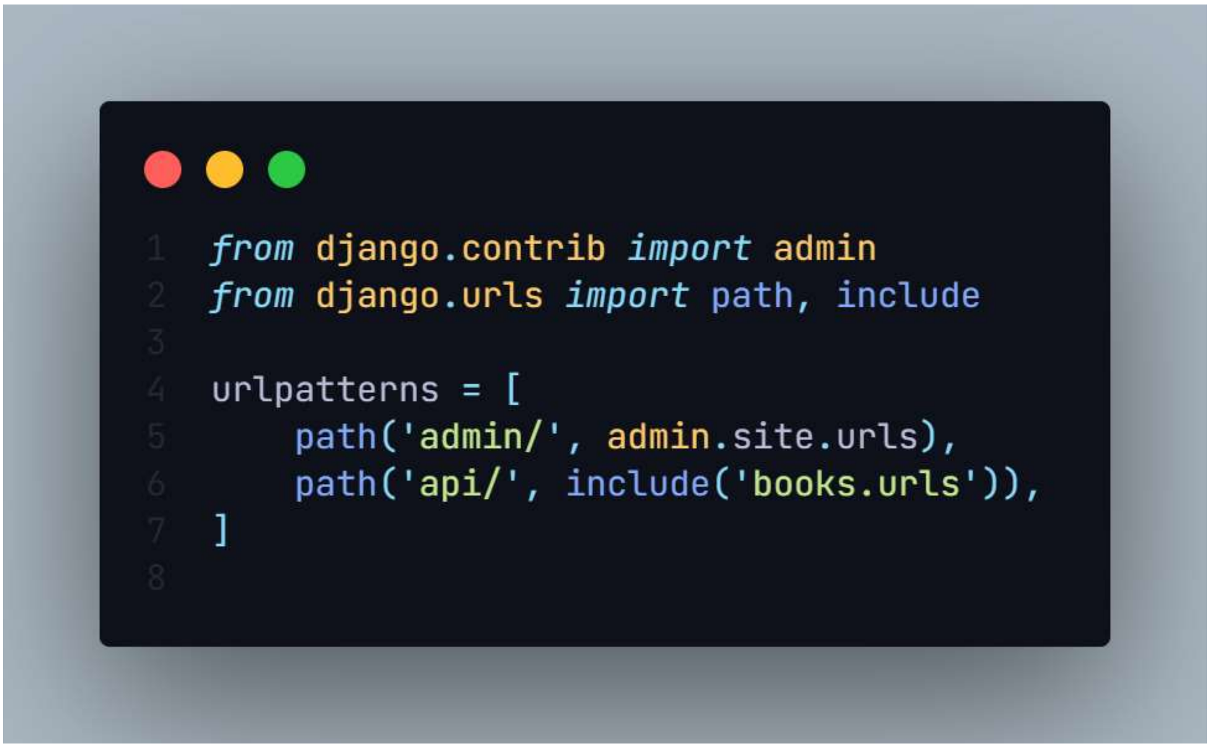
```python
from rest_framework import status
from rest_framework.response import Response
from rest_framework.views import APIView
from .models import Book
from .serializers import BookSerializer

class BookListCreateAPIView(APIView):
    """Handles GET (list all books) and POST (create a new book)"""

    def get(self, request):
        books = Book.objects.all()
        serializer = BookSerializer(books, many=True)
        return Response(serializer.data)

    def post(self, request):
        serializer = BookSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

class BookDetailAPIView(APIView):
    """Handles GET, PUT, and DELETE for a single book"""

    def get_object(self, pk):
        try:
            return Book.objects.get(pk=pk)
        except Book.DoesNotExist:
            return None

    def get(self, request, pk):
        book = self.get_object(pk)
        if not book:
            return Response({"error": "Book not found"}, status=status.HTTP_404_NOT_FOUND)
        serializer = BookSerializer(book)
        return Response(serializer.data)

    def put(self, request, pk):
        book = self.get_object(pk)
        if not book:
            return Response({"error": "Book not found"}, status=status.HTTP_404_NOT_FOUND)
        serializer = BookSerializer(book, data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    def delete(self, request, pk):
        book = self.get_object(pk)
        if not book:
            return Response({"error": "Book not found"}, status=status.HTTP_404_NOT_FOUND)
        book.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)

```

Books/urls.py

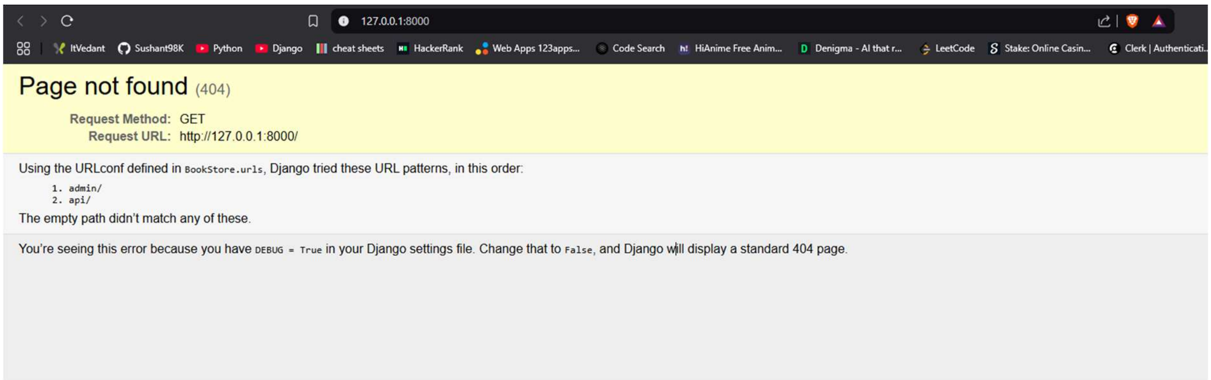


```python
from django.urls import path
from .views import BookListCreateAPIView, BookDetailAPIView

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

Urls.py



```python
from django.contrib import admin
from django.urls import path, include

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



Page not found (404)

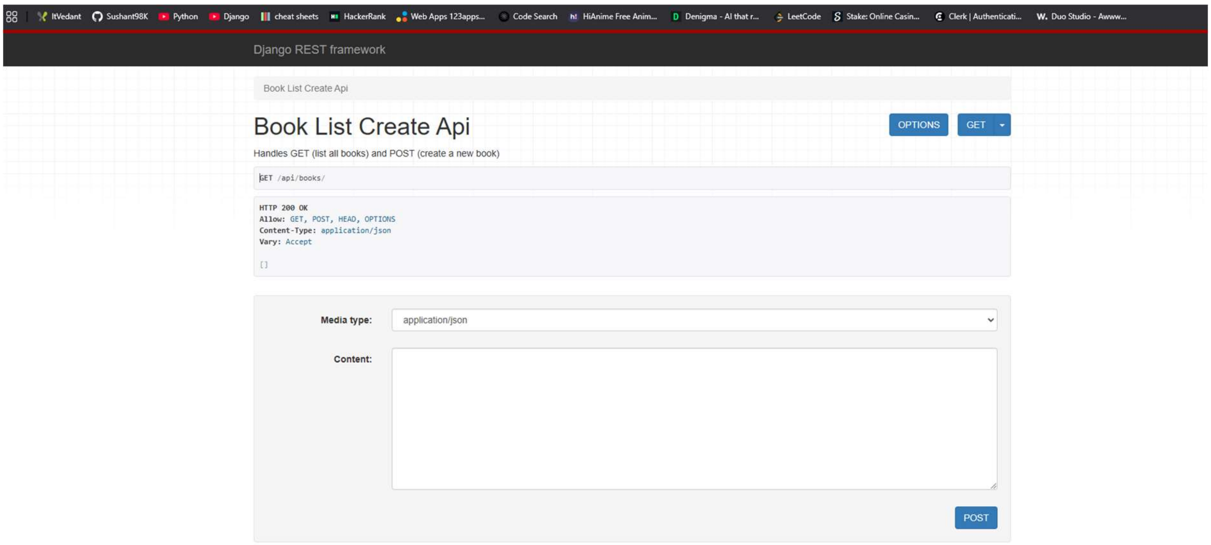
Request Method: GET
Request URL: http://127.0.0.1:8000/

Using the URLconf defined in `BookStore.urls`, Django tried these URL patterns, in this order:

```
1. admin/
2. api/
```

The empty path didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



Book List Create Api

Handles GET (list all books) and POST (create a new book)

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

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

[]
```

Media type: application/json

Content: