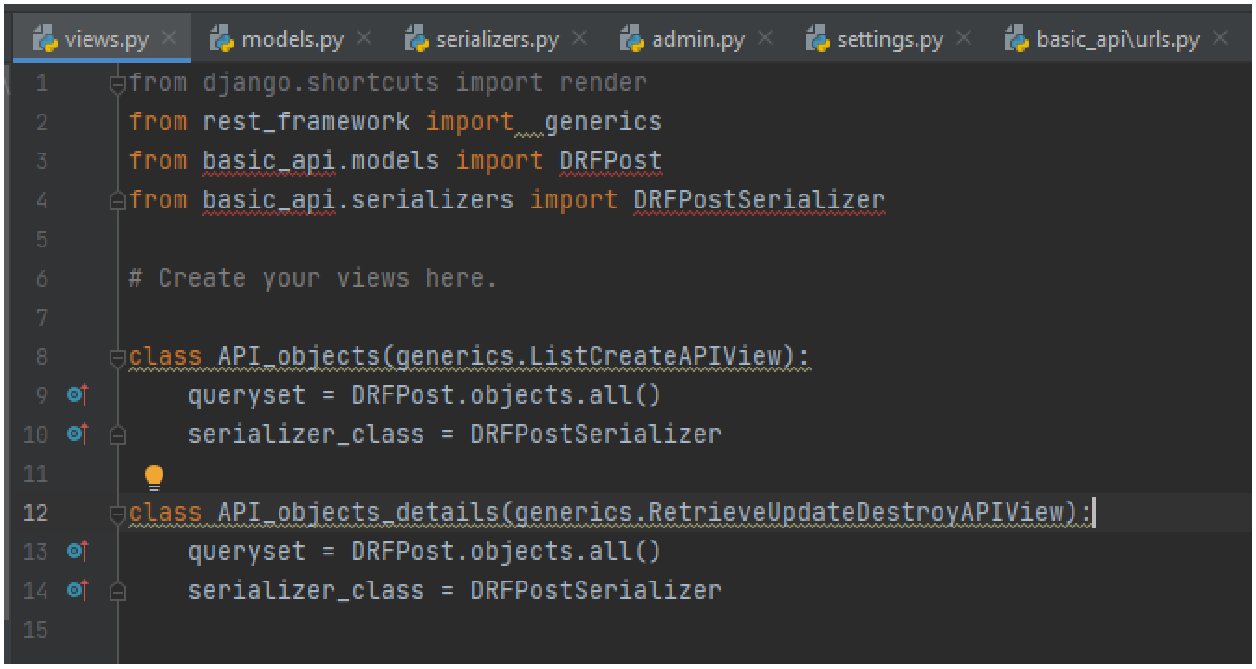
Saad Ali Hafiz 18B-025-SE A

Lab: 12

Student Task:

Code:

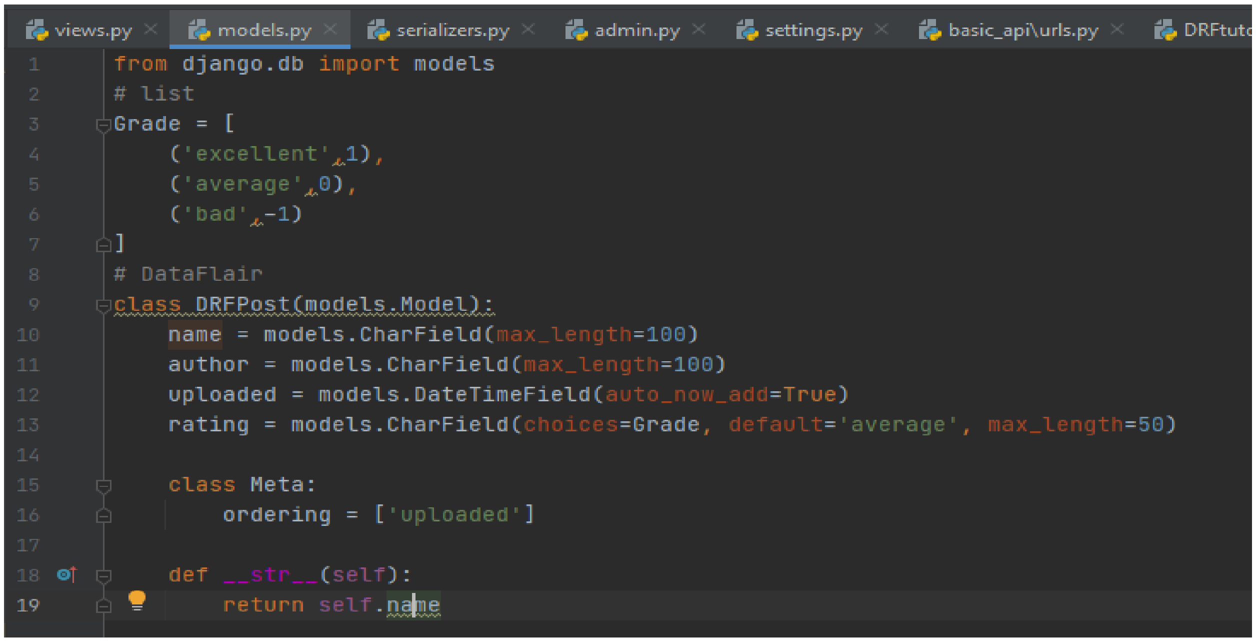
Views.py



views.py models.py serializers.py admin.py settings.py basic_api\urls.py

```
from django.shortcuts import render
from rest_framework import  generics
from basic_api.models import DRFPost
from basic_api.serializers import DRFPostSerializer

# Create your views here.

class API_objects(generics.ListCreateAPIView):
    queryset = DRFPost.objects.all()
    serializer_class = DRFPostSerializer

class API_objects_details(generics.RetrieveUpdateDestroyAPIView):
    queryset = DRFPost.objects.all()
    serializer_class = DRFPostSerializer

```

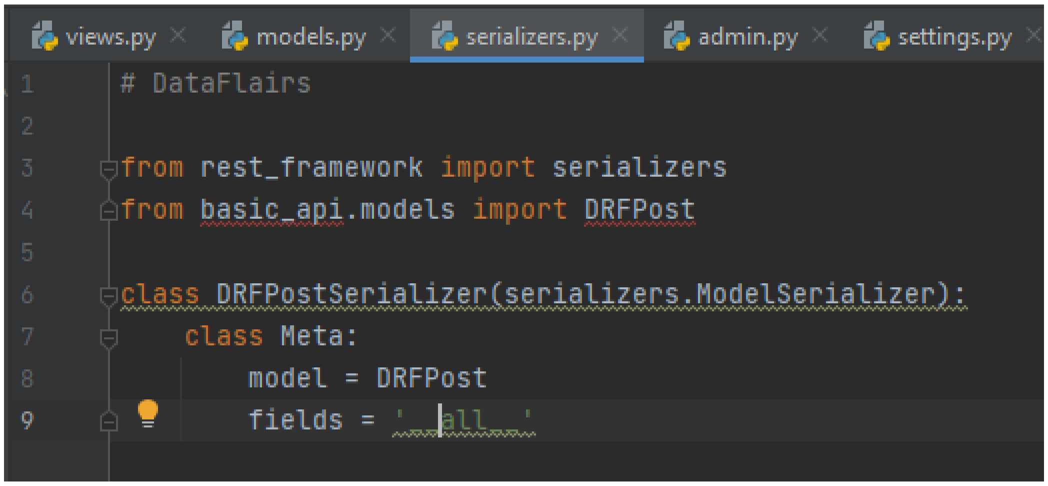
Models.py



views.py models.py serializers.py admin.py settings.py basic_api\urls.py DRFtuto

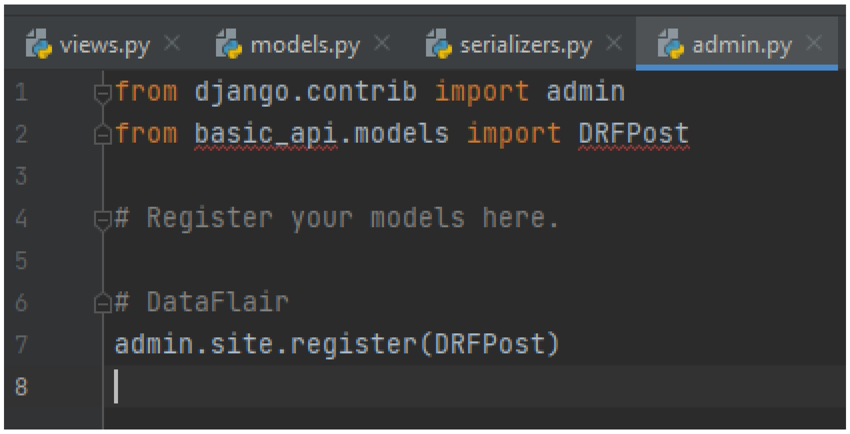
```
from django.db import models
# list
Grade = [
    ('excellent',1),
    ('average',0),
    ('bad',-1)
]
# DataFlair
class DRFPost(models.Model):
    name = models.CharField(max_length=100)
    author = models.CharField(max_length=100)
    uploaded = models.DateTimeField(auto_now_add=True)
    rating = models.CharField(choices=Grade, default='average', max_length=50)

    class Meta:
        ordering = ['uploaded']

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

Serializers.py



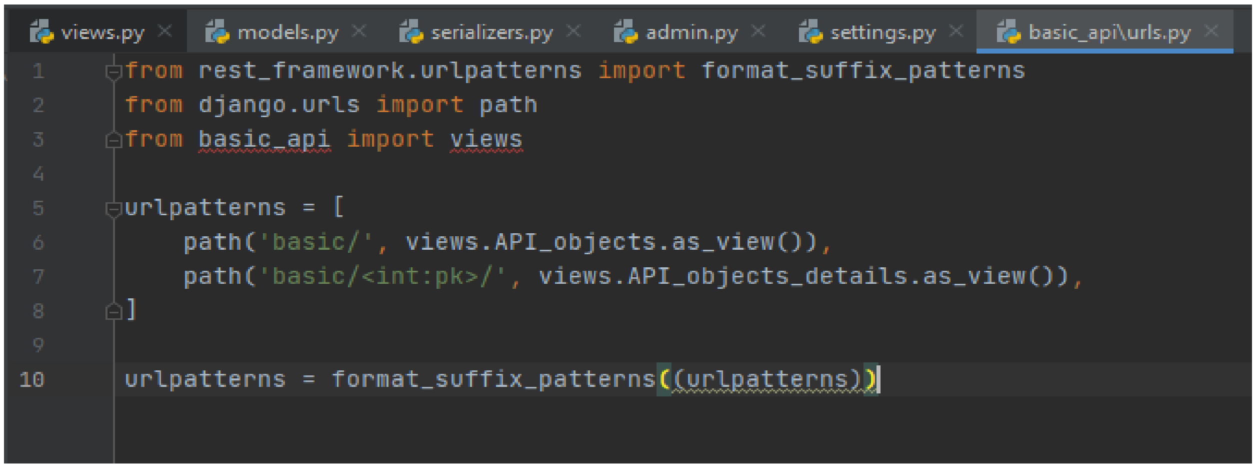
```
# DataFlairs

from rest_framework import serializers
from basic_api.models import DRFPost

class DRFPostSerializer(serializers.ModelSerializer):
    class Meta:
        model = DRFPost
        fields = '__all__'
```

Admin.py



```
from django.contrib import admin
from basic_api.models import DRFPost

# Register your models here.

# DataFlair
admin.site.register(DRFPost)
```

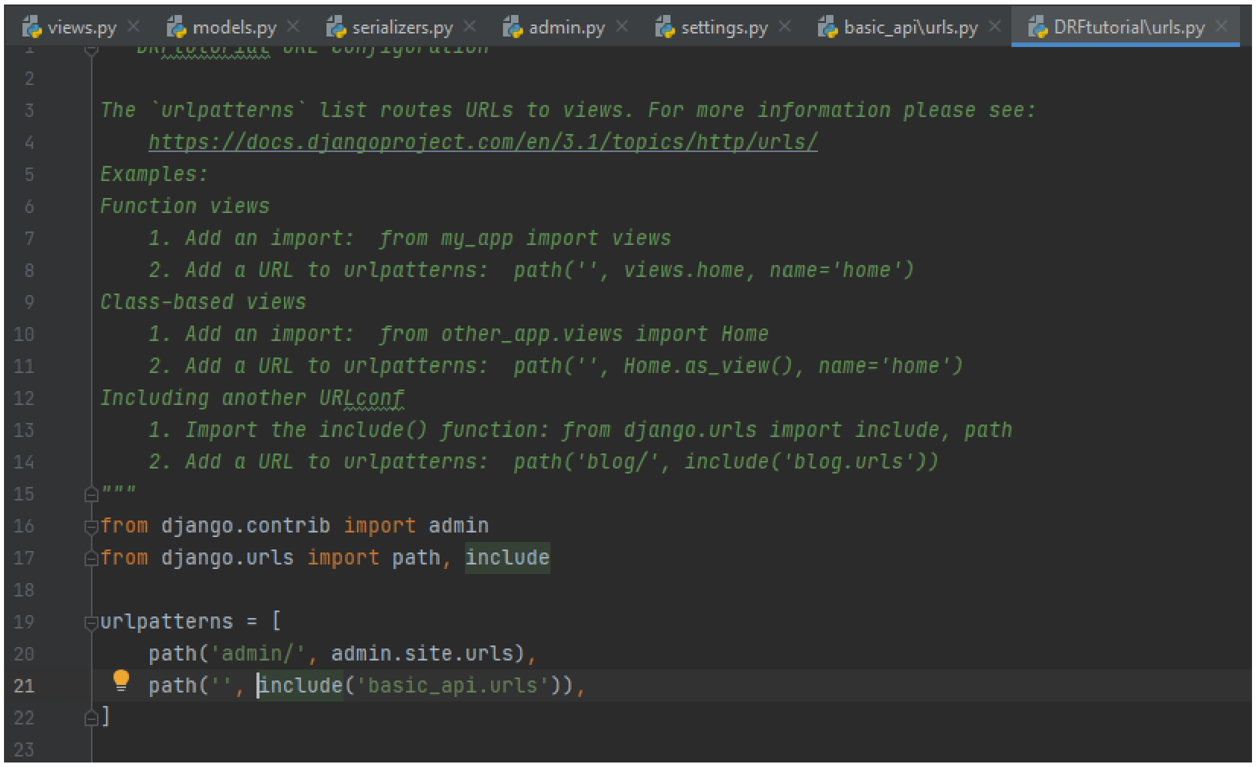
Basi_api\urls.py



views.py × models.py × serializers.py × admin.py × settings.py × basic_api\urls.py ×

```
from rest_framework.urlpatterns import format_suffix_patterns
from django.urls import path
from basic_api import views

urlpatterns = [
    path('basic/', views.API_objects.as_view()),
    path('basic/<int:pk>/', views.API_objects_details.as_view()),
]

urlpatterns = format_suffix_patterns((urlpatterns))
```

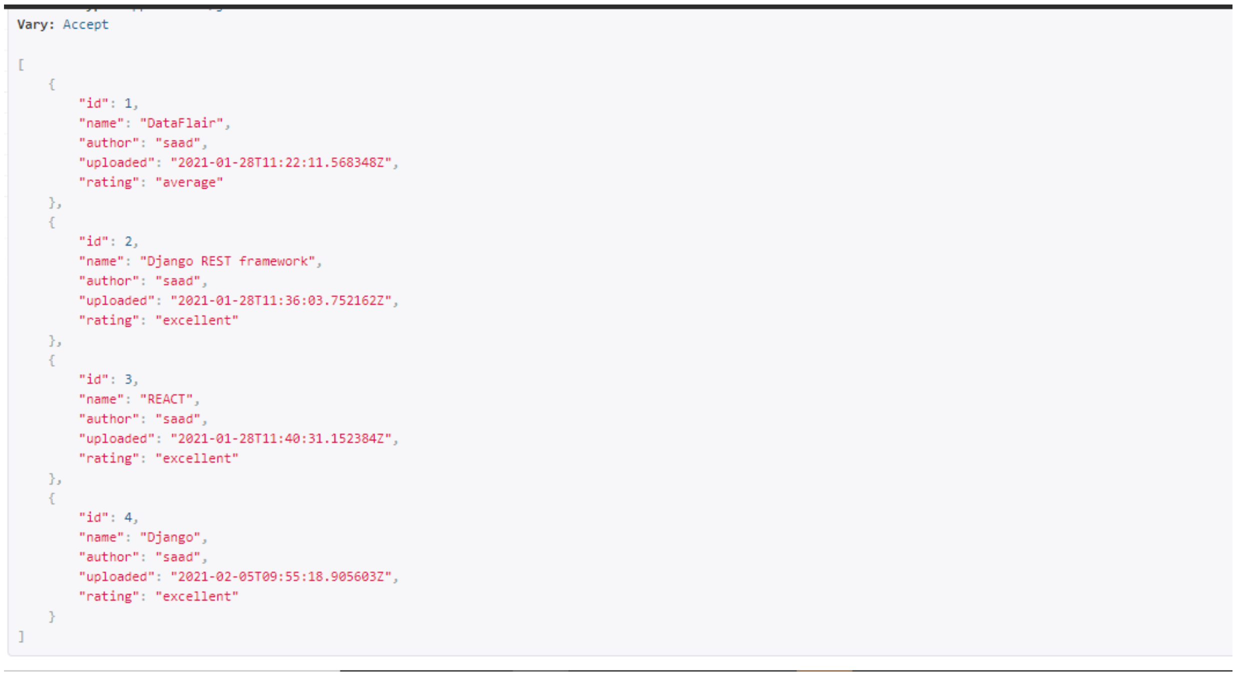
DRFtutorial\urls.py



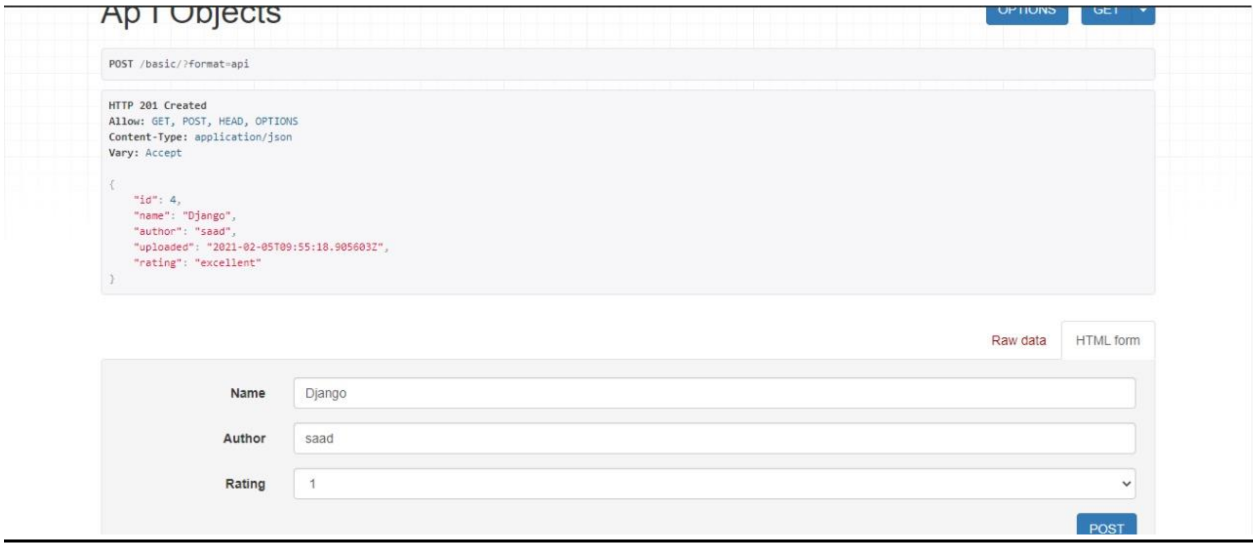
views.py × models.py × serializers.py × admin.py × settings.py × basic_api\urls.py × DRFtutorial\urls.py ×

```
    DRFtutorial URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('basic_api.urls')),
]

```

Outputs:



```
Vary: Accept

[
    {
        "id": 1,
        "name": "DataFlair",
        "author": "saad",
        "uploaded": "2021-01-28T11:22:11.568348Z",
        "rating": "average"
    },
    {
        "id": 2,
        "name": "Django REST framework",
        "author": "saad",
        "uploaded": "2021-01-28T11:36:03.752162Z",
        "rating": "excellent"
    },
    {
        "id": 3,
        "name": "REACT",
        "author": "saad",
        "uploaded": "2021-01-28T11:40:31.152384Z",
        "rating": "excellent"
    },
    {
        "id": 4,
        "name": "Django",
        "author": "saad",
        "uploaded": "2021-02-05T09:55:18.905603Z",
        "rating": "excellent"
    }
]
```



Api Objects

```
POST /basic/?format=api
```

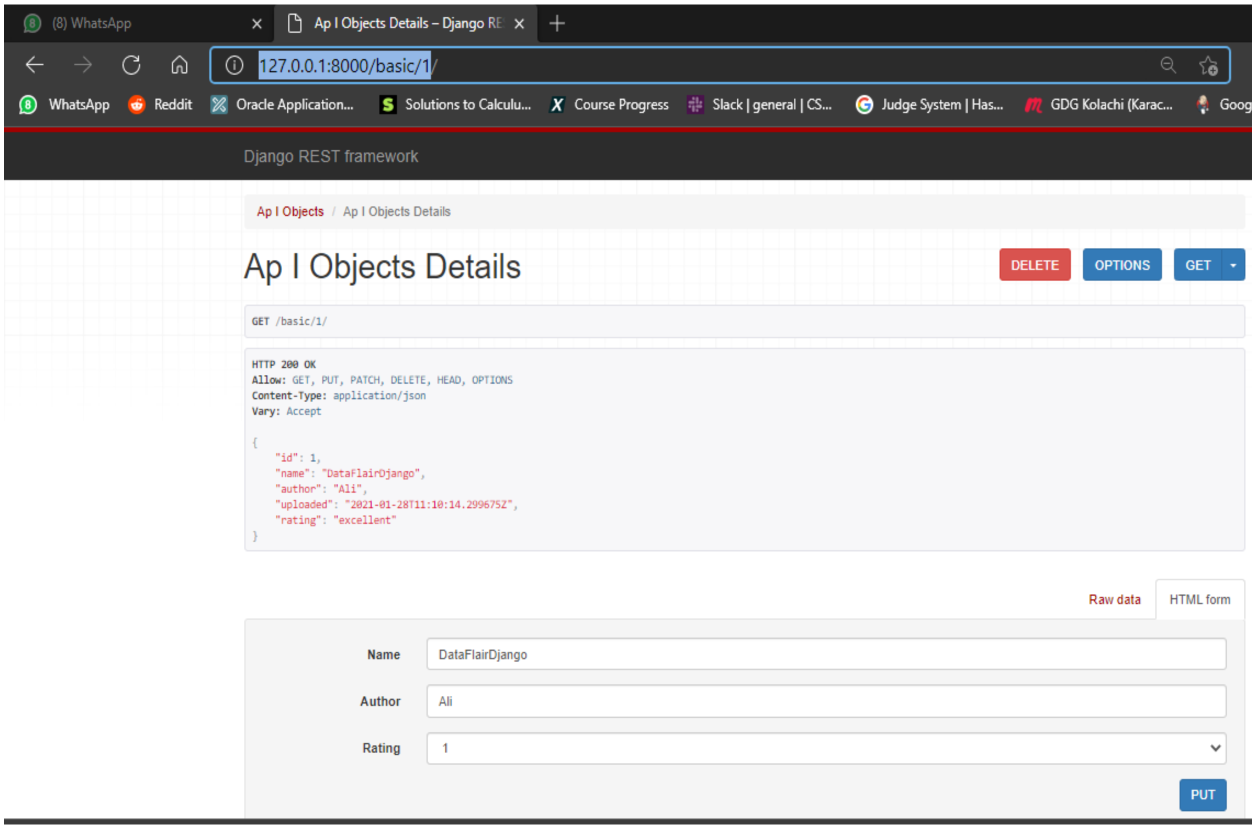
```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "id": 4,
    "name": "Django",
    "author": "saad",
    "uploaded": "2021-02-05T09:55:18.905603Z",
    "rating": "excellent"
}
```

Name	Django
Author	saad
Rating	1

The above output was from - http://127.0.0.1:8000/basic/

Calling unique ID's:

http://127.0.0.1:8000/basic/1/



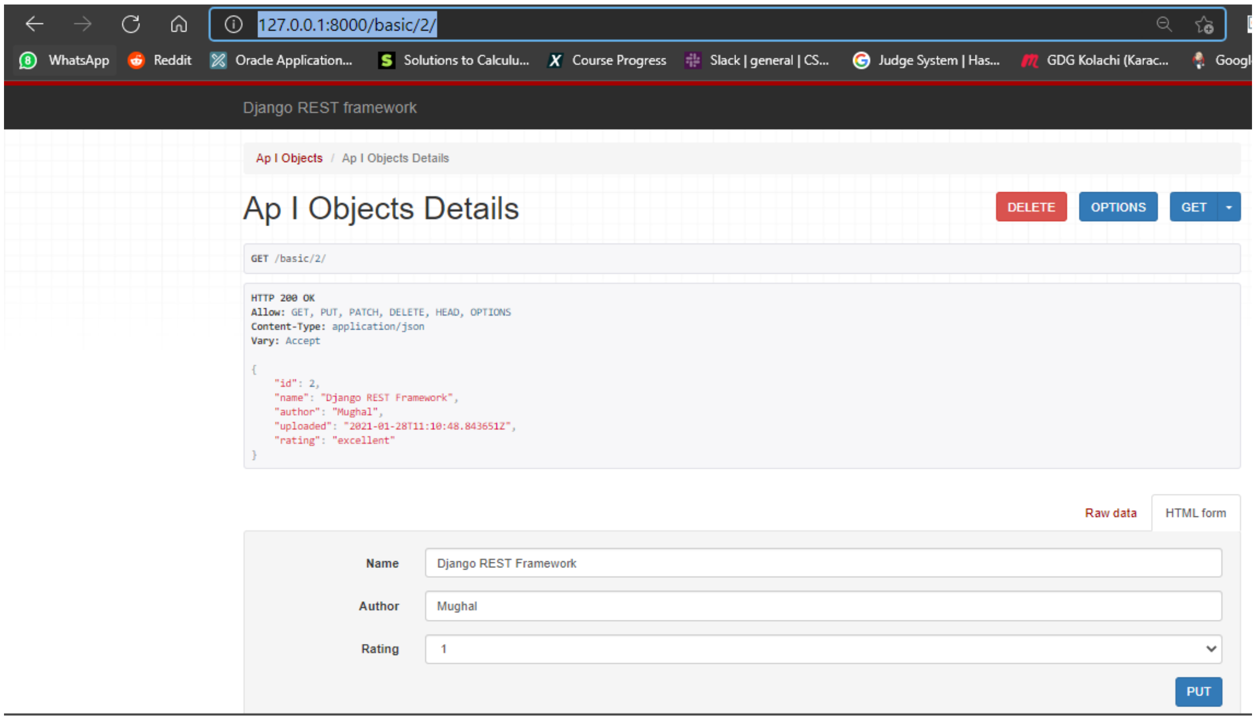
Ap I Objects Details

```
GET /basic/1/
```

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

{
    "id": 1,
    "name": "DataFlairDjango",
    "author": "Ali",
    "uploaded": "2021-01-28T11:10:14.299675Z",
    "rating": "excellent"
}
```

Name	DataFlairDjango
Author	Ali
Rating	1

http://127.0.0.1:8000/basic/2/



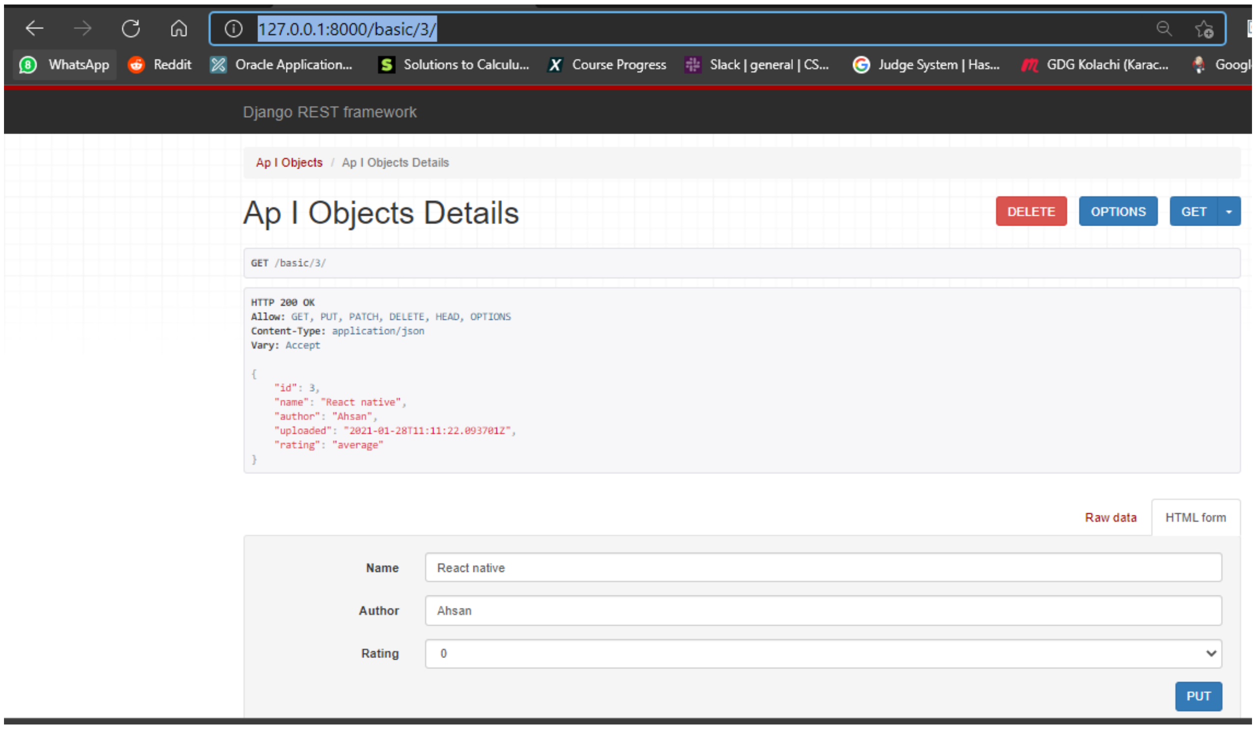
Ap I Objects Details

```
GET /basic/2/
```

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

{
    "id": 2,
    "name": "Django REST Framework",
    "author": "Mughal",
    "uploaded": "2021-01-28T11:10:48.843651Z",
    "rating": "excellent"
}
```

Name	Django REST Framework
Author	Mughal
Rating	1

http://127.0.0.1:8000/basic/3/



Ap I Objects Details

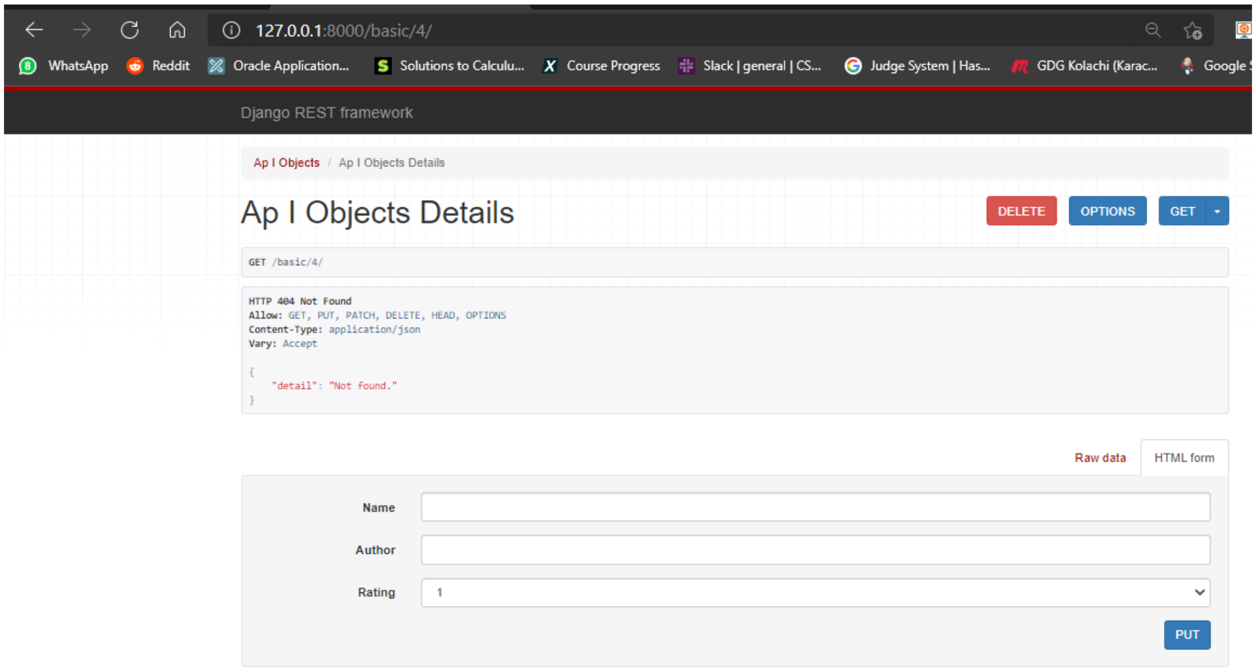
```
GET /basic/3/
```

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

{
    "id": 3,
    "name": "React native",
    "author": "Ahsan",
    "uploaded": "2021-01-28T11:11:22.093701Z",
    "rating": "average"
}
```

Name	React native
Author	Ahsan
Rating	0

Finally, the one which in actual doesn't exists:

http://127.0.0.1:8000/basic/4/



Ap I Objects Details

```
GET /basic/4/
```

```
HTTP 404 Not Found
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "detail": "Not found."
}
```

Name	
Author	
Rating	1