**Simple Steps to Create a REST API Using Django**

**REST Framework in Win10**

**1st Download & Install : for database software**

[**https://www.postgresql.org/ftp/pgadmin/pgadmin4/v9.1/windows/**](https://www.postgresql.org/ftp/pgadmin/pgadmin4/v9.1/windows/) **- Setup,**

**2nd Create new database, Username & Password**

**3rd Run in Background**

**Download Python Form Office Site**

**https://www.python.org/ftp/python/3.13.2/python-3.13.2-amd64.exe**

**Install File & Restart OS Windows**

**Download VS Code & Install : Open VS Code**

*Run : python –version*

*Run : django-admin –version*

**If Not Present, Then you need to install all packages**

*Run :* pip freeze List Show packages || Not Available then install all packages

*Run :* **pip install -r requirements.txt (if Existing Project then Run important)**

*Run :* pip install django

*Run :* pip install djangorestframework

*Run : pip install django\_rest\_framework*

*Run :* pip install psycopg2

***If Security Error Show Then***

*Run : Set-ExecutionPolicy Unrestricted -Scope Process*

**1st Create Main Project :** *Run :* **django-admin startproject app**

**Cd app**

**2nd Create Project Module or app :** *Run :***python manage.py startapp upload\_app**

GoTo main app folder decretory, open **setting.py** and goto section config #

Application definition- install apps name

Under add 2 line simple.

  'upload\_app',

    'rest\_framework',

Add Lines

# MEDIA settings

MEDIA\_URL = '/media/'

MEDIA\_ROOT = BASE\_DIR / 'media'

save & exit;

After go to database setting: use default

 'ENGINE': 'django.db.backends.postgresql',

        'NAME':'api\_health',

        'USER':'postgres',

        'PASSWORD':'admin',

        'HOST': 'localhost',  # Change if using a remote DB

        'PORT': '5432',  # Default PostgreSQL port

After database credentials add save & exit;

Open **urls.py** And add this, basically api folder root

from django.contrib import admin

from django.urls import path,include

from django.conf import settings

from django.conf.urls.static import static

from django.urls import path, include

urlpatterns = [

path('api/', include('upload\_app.urls')),

]

if settings.DEBUG:

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

After save & exit; All setting are done in main project.

**Now Open Api Folder or app folder**

**Open apps.py and view code**

from django.apps import AppConfig

class UploadAppConfig(AppConfig):

    default\_auto\_field = 'django.db.models.BigAutoField'

    name = 'upload\_app'

After save & exit;

**open models.py and Create**

from django.db import models

class UploadedFile(models.Model):

file = models.FileField(upload\_to='uploads/')

uploaded\_at = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.file.name

After save & exit;

**Create a black serializers.py page …**

**open serializers.py and edit**

from rest\_framework import serializers

from .models import UploadedFile

class FileUploadSerializer(serializers.ModelSerializer):

class Meta:

model = UploadedFile

fields = ['id', 'file', 'uploaded\_at']

After save & exit;

**open views.py and edit**

from rest\_framework.parsers import MultiPartParser, FormParser

from rest\_framework.response import Response

from rest\_framework import status, generics

from .models import UploadedFile

from .serializers import FileUploadSerializer

class FileUploadView(generics.CreateAPIView):

    queryset = UploadedFile.objects.all()

    serializer\_class = FileUploadSerializer

    parser\_classes = [MultiPartParser, FormParser]  # Handle file uploads

    def get(self, request, \*args, \*\*kwargs):

       # """Retrieve all uploaded files."""

        files = self.get\_queryset()

        serializer = self.get\_serializer(files, many=True)

        return Response(serializer.data, status=status.HTTP\_200\_OK)

    def post(self, request, \*args, \*\*kwargs):

        serializer = self.get\_serializer(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)

After save & exit;

**open urls.py and edit**

from django.urls import path

from .views import FileUploadView

urlpatterns = [

path('upload/', FileUploadView.as\_view(), name='file-upload'),

]

After save & exit;

cd app

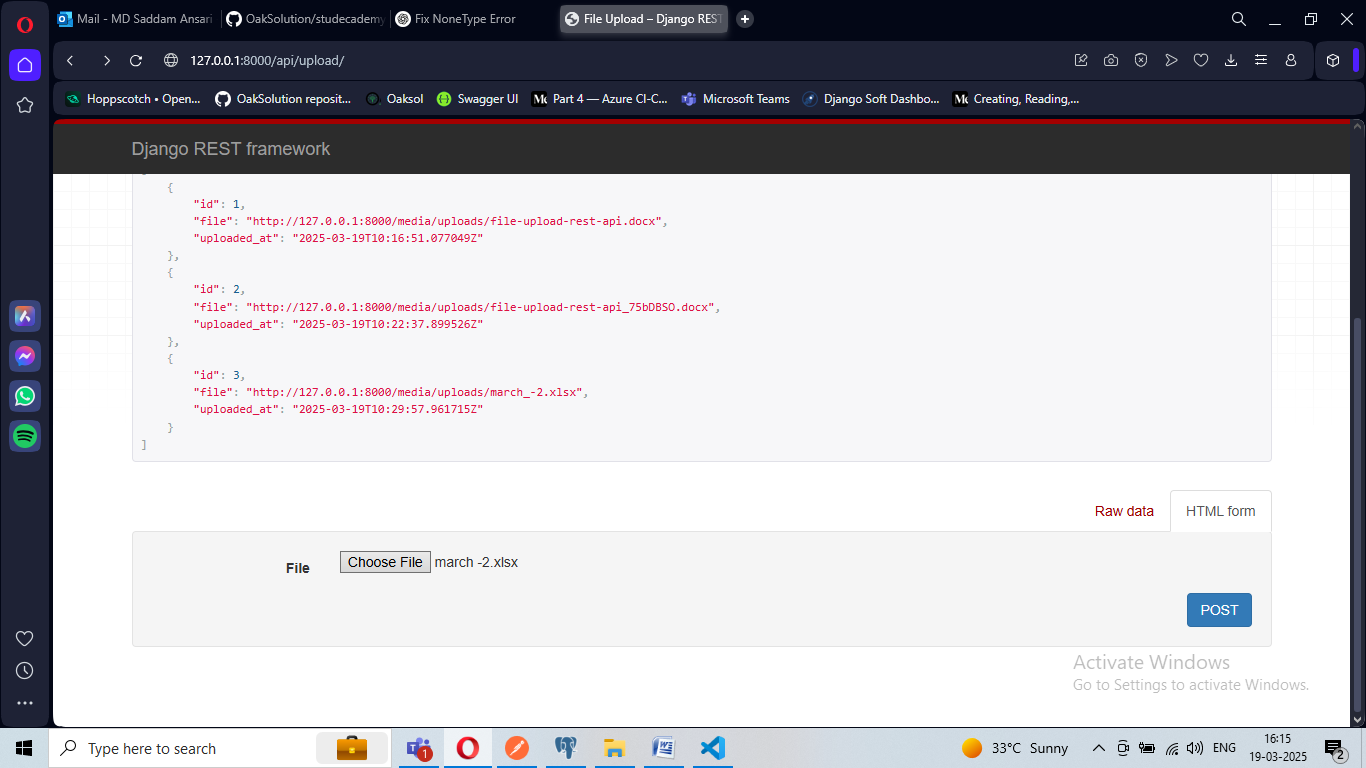
python manage.py makemigrations

python manage.py migrate **every work after**

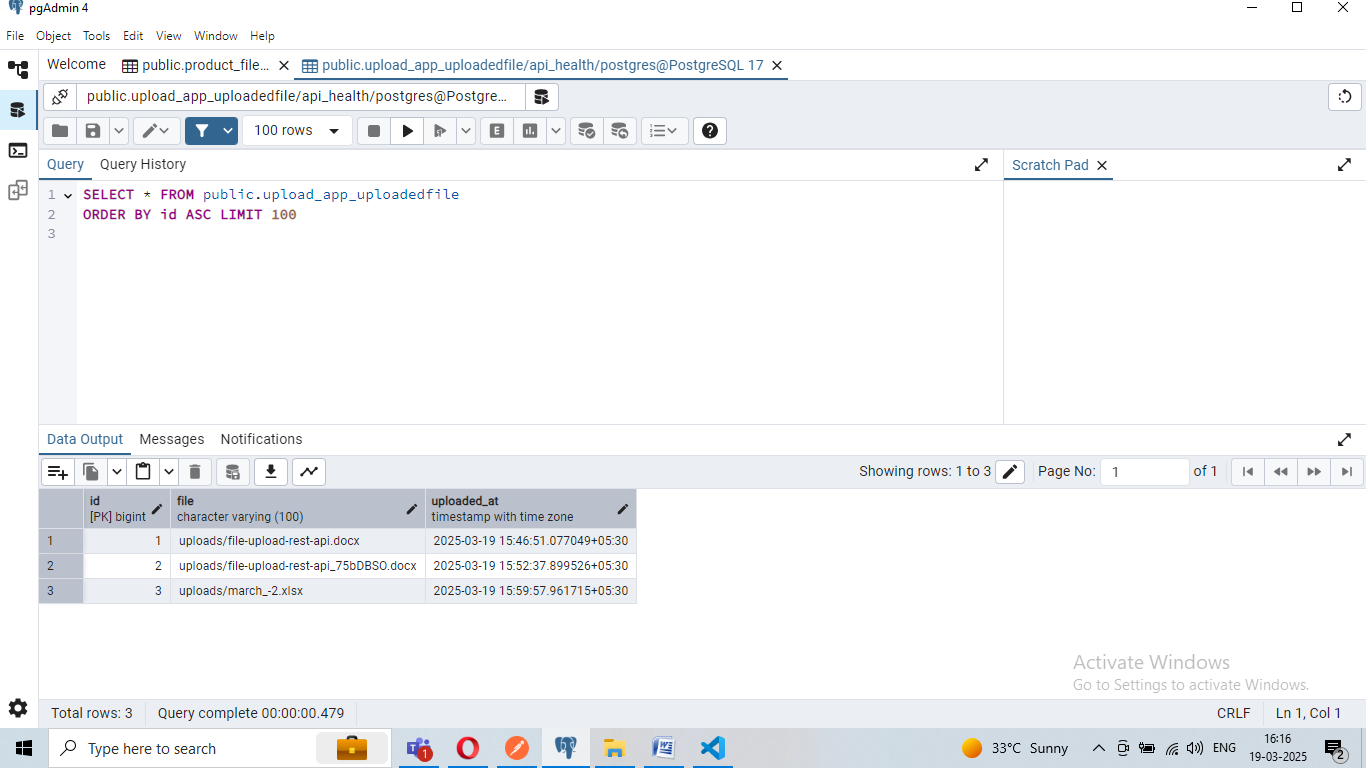
python manage.py runserver 8000 – **Get web address with custom port**

**after run open browser**

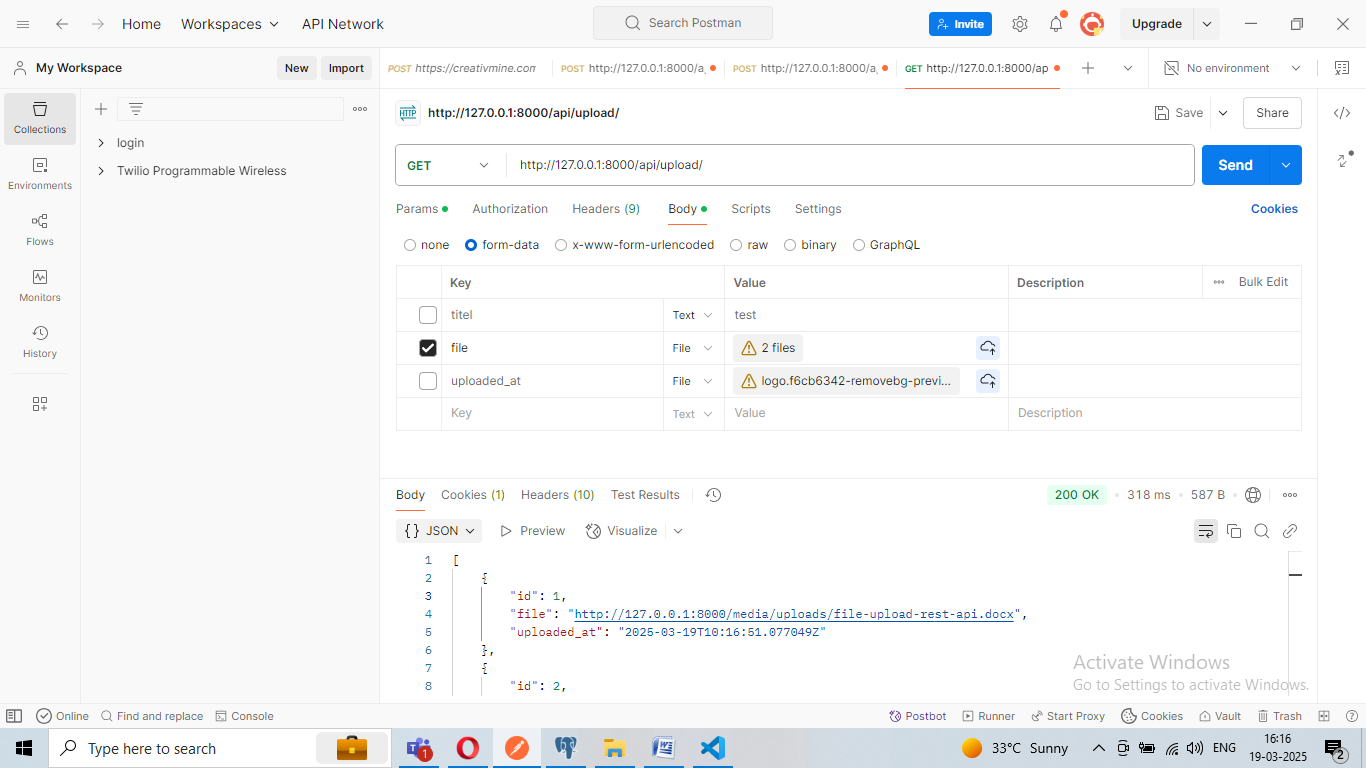
<http://127.0.0.1:8000/api/upload/>

****

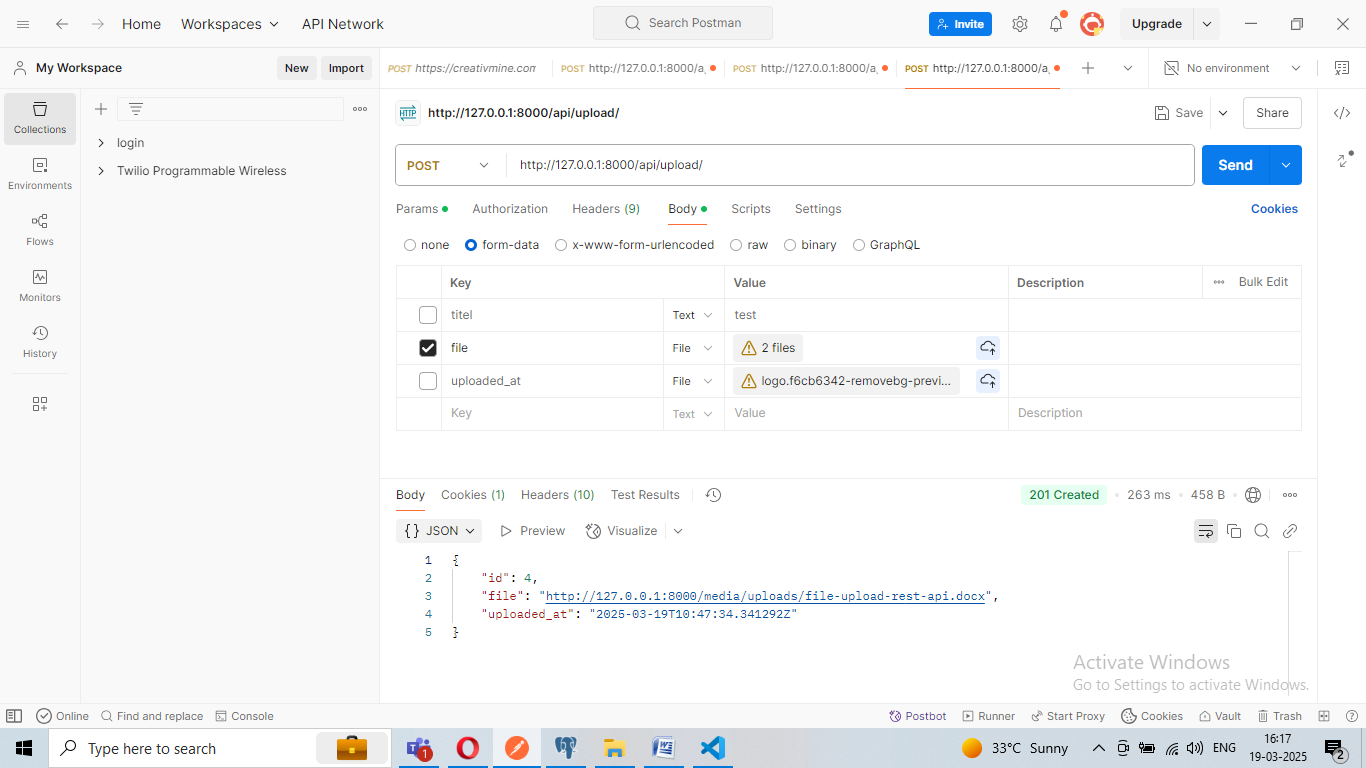
**Server root view mode**

****

**Show data in table under database .**

****

**Get Method Vai Check in database list .**

****

**Post Method successfully Working .**

**Ref : Chat-GPT**