**RUN THESE COMMANDS**

E:\My\_Django\_Stuff\store\_API\env\Scripts>activate

(env) E:\My\_Django\_Stuff\store\_API\env\Scripts>cd..

(env) E:\My\_Django\_Stuff\store\_API\env>cd..

(env) E:\My\_Django\_Stuff\store\_API>cd..

(env) E:\My\_Django\_Stuff>django-admin startproject USER\_POSTS\_CRUD

(env) E:\My\_Django\_Stuff>cd USER\_POSTS\_CRUD

(env) E:\My\_Django\_Stuff\USER\_POSTS\_CRUD>python manage.py startapp posts

(env) E:\My\_Django\_Stuff\USER\_POSTS\_CRUD>python manage.py runserver

**Now to connect postresql we need to install psycopg2, run command,**

Pip install psycopg2

**Open pgadmin4, enter password, create database and add it to your projects settings.py databases,**

DATABASES = {

'default': {

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

'NAME': 'blogposts',

'USER': 'postgres',

'PASSWORD': '1234',

'HOST': 'localhost',

'PORT': '5432',

}

}

**Now add your app to installed apps in apps settings along with 'rest\_framework',**

**Now create models.py,**

# models.py

from django.db import models

from django.contrib.auth.models import User

class Post(models.Model):

title = models.CharField(max\_length=255)

body = models.TextField()

author = models.ForeignKey(User, on\_delete=models.CASCADE)

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

**Now run COMMANDS,**

python manage.py makemigrations

python manage.py migrate

**Create Serializers:**

**Define serializers to convert your model instances to JSON and vice versa.**

from rest\_framework import serializers

from .models import Post

class PostSerializer (serializers.ModelSerializer):

class Meta:

model = Post

fields = ['id', 'title', 'body', 'author', 'created\_at']

**Create Views:**

**Implement views for CRUD operations. Ensure that users can only update or delete their own posts.**

**# views.py**

from rest\_framework import generics, permissions

from .models import Post

from .serializers import PostSerializer

from .permissions import IsAuthorOrReadOnly

class PostListCreateView(generics.ListCreateAPIView):

queryset = Post.objects.all()

serializer\_class = PostSerializer

permission\_classes = [permissions.IsAuthenticatedOrReadOnly]

def perform\_create(self, serializer):

serializer.save(author=self.request.user)

class PostDetailView(generics.RetrieveUpdateDestroyAPIView):

queryset = Post.objects.all()

serializer\_class = PostSerializer

permission\_classes = [IsAuthorOrReadOnly]

**Create Permissions:**

**Define a custom permission class to ensure that users can only update or delete their own posts.**

**# permissions.py**

from rest\_framework import permissions

class IsAuthorOrReadOnly(permissions.BasePermission):

def has\_object\_permission(self, request, view, obj):

if request.method in permissions.SAFE\_METHODS:

return True

return obj.author == request.user

**Configure URLs:**

**Set up URL patterns for your views.**

**# urls.py**

from django.urls import path

from .views import PostListCreateView, PostDetailView

urlpatterns = [

path('posts/', PostListCreateView.as\_view(), name='post-list-create'),

path('posts/<int:pk>/', PostDetailView.as\_view(), name='post-detail'),

]

**set up your project's URLs to include the URLs from your app.**

from django.contrib import admin

from django.urls import path,include

urlpatterns = [

path('admin/', admin.site.urls),

path('',include('posts.urls'))

]

**Now Run COMMAND,**

Python manage.py createsuperuser

python manage.py runserver