**API Creation**

Step 1: install django rest framework application

**pip install djangorestframework**

Step 2: Register django rest application in setting.py

**INSTALLED\_APPS = [‘rest\_framework’]**

Step 3: Configure database

**DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.mysql',  
 'NAME': 'CURD\_API',  
 'USERNAME': 'root',  
 'PASSWORD': 'Mayuri@94'  
 }  
}**

Step 4: Create models.py

**from django.db import models  
  
# Create your models here.  
class ProductDetails(models.Model):  
 no = models.IntegerField(primary\_key=True)  
 name = models.CharField(max\_length=35)  
 quantity = models.IntegerField()  
 trusted = models.BooleanField()**

Step 5: Create file in application

**serializers.py**

Step 6: Convert data from database to JSON string

serializer.py

**from rest\_framework import serializers  
from .models import ProductDetails  
  
class ProductSerializer(serializers.ModelSerializer):  
 class Meta:  
 model = ProductDetails  
 fields = '\_\_all\_\_'**

Step 7: view.py

**from django.shortcuts import render  
from .models import ProductDetails  
from .serializers import ProductDetailsSerializers  
from rest\_framework.response import Response  
from rest\_framework import status  
from rest\_framework.views import APIView  
  
# Create your views here.  
class ProductTable(APIView):  
 def get(self, r, \*args, \*\*kwargs):  
 prd\_obj = ProductDetails.objects.all()  
 ser\_obj = ProductDetailsSerializers(prd\_obj, many=True)  
 return Response(ser\_obj.data)  
  
 def post(self, r):  
 ser\_obj = ProductDetailsSerializers(data=r.data)  
 if ser\_obj.is\_valid():  
 ser\_obj.save()  
 return Response(ser\_obj.data, status=status.HTTP\_201\_CREATED)  
 return Response(ser\_obj.errors, status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def put(self, r, pk):  
 prd\_obj = ProductDetails.objects.get(pk=pk)  
 ser\_obj = ProductDetailsSerializers(prd\_obj, data=r.data)  
 if ser\_obj.is\_valid():  
 ser\_obj.save()  
 return Response(ser\_obj.data, status=status.HTTP\_201\_CREATED)  
 return Response(ser\_obj.errors, status=status.HTTP\_400\_BAD\_REQUEST)  
  
 def delete(self, r, pk):  
 prd\_obj = ProductDetails.objects.get(pk=pk)  
 prd\_obj.delete()  
 return Response(status=status.HTTP\_204\_NO\_CONTENT)**

Step 8: urls

**from API import views  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('get/', views.ProductTable.as\_view()),  
 path('post/', views.ProductTable.as\_view()),  
 path('put/<int:pk>/', views.ProductTable.as\_view()),  
 path('delete/<int:pk>/', views.ProductTable.as\_view()),  
]**