Procedures

1.Creating Virtual Environment

**python -m venv .venv**

**2.Activate the Script**

**.venv\Scripts\activate**

3.Install Packages

**pip install djangorestframework**

**pip install mysqlclient (MySQL DATABASE)**

4.Create Project

django-admin startproject P1\_BillingSystem .

5.Create Application

**python manage.py startapp Inventory**

6.Project Settings File Register the Application

**INSTALLED\_APPS = [**

**'django.contrib.admin',**

**'django.contrib.auth',**

**'django.contrib.contenttypes',**

**'django.contrib.sessions',**

**'django.contrib.messages',**

**'django.contrib.staticfiles',**

**'Inventory'**

**]**

7.Start The Server Once to avoid Errors.

**python manage.py runserver**

8. Project Settings File Set the Database you need

8.1 For SQLITE3 Database

**In VScode Install sqliteviewer Extension.**

**It helps to view Sqlite Database.**

**DATABASES = {**

**'default': {**

**'ENGINE': 'django.db.backends.sqlite3',**

**'NAME': BASE\_DIR / 'db.sqlite3',**

**}**

**}**

8.2 For MySQL DATAbase (MySQL WorkBench)

pip install mysqlclient

**DATABASES = {**

**'default': {**

**'ENGINE': 'django.db.backends.mysql',**

**'NAME': 'P1\_Billing',**

**'USER':'root',**

**'PASSWORD':'',**

**'HOST':'localhost',**

**'PORT':'3306'**

**}**

**}**

8.3 MySQL Database(Wamp Server)

pip install mysqlclient

**DATABASES = {**

**'default': {**

**'ENGINE': 'django.db.backends.mysql',**

**'NAME': 'P1\_Billing',**

**'USER':'root',**

**'PASSWORD':'',**

**}**

**}**

9.Create The Database in Mysql Console or in Work Bench.

**MySQL Commands**

**show databases;**

**create database P1\_Billing;**

**show databases;**

10.Then Create your Tables in Application

## Inventory/models.py

**from django.db import models**

***# Create your models here.***

**class Productdetails(models.Model):**

**product\_name = models.CharField(max\_length=200,null=True)**

**code         = models.CharField(max\_length=200,null=True)**

**price        = models.FloatField(default=0)**

11.Migrate the Database in Terminal

python manage.py makemigrations

python manage.py migrate

PROJECT 1

* 1.Here the API CALLS are prepared using the Class Based.
* 2.Here MySQL Database are used(Wamp Server).
* 3.Here No Django Serializing is Done.

***PROJECT -1***

CODINGS

***### Project urls.py***

**from django.contrib import admin**

**from django.urls import path,include**

**urlpatterns = [**

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

**path('inventory/',   include('Inventory.urls')),**

**]**

***### Application urls.py***

**from django.urls import path**

**from .views import \***

**urlpatterns = [**

**path('products/',          Ourproducts.as\_view() ),**

**path('products/<int:id>/', Ourproductsbyid.as\_view() ),**

**]**

***### Application models.py***

**from django.db import models**

***# Create your models here.***

**class Myproducts(models.Model):**

**product\_name = models.CharField(max\_length=200,null=True)**

**code         = models.CharField(max\_length=200,null=True)**

**price        = models.FloatField(default=0)**

***### Application views.py***

**from rest\_framework .views import APIView**

**from rest\_framework .response import Response**

**from .models import \***

**class Ourproducts(APIView):**

**def post(self,request):**

**uservalues = request.data**

**print(uservalues)**

**new\_product = Myproducts(product\_name=uservalues['product\_name'],code=uservalues['code'],price=uservalues['price'] )**

**new\_product.save()**

**return Response("New Data Saved")**

**def get(self,request):**

**products = Myproducts.objects.all()**

**fullproducts = []**

**for each in products:**

**singleproduct = {**

**'product\_name' : each.product\_name,**

**'product\_code' : each.code,**

**'product\_price': each.price,**

**'product\_id'   : each.id,**

**}**

**fullproducts.append(singleproduct)**

**return Response(fullproducts)**

**class Ourproductsbyid(APIView):**

**def get(self,request,id):**

**detail = Myproducts.objects.get(id=id)**

**product = {**

**'product\_name':detail.product\_name,**

**'product\_code':detail.code,**

**'product\_price':detail.price,**

**'product\_id': detail.id**

**}**

**return Response(product)**

**def put(self,request,id):**

**detail    = Myproducts.objects.get(id=id)**

**uservalue = request.data**

**detail.product\_name = uservalue['name']**

**detail.code         = uservalue['code']**

**detail.price        = uservalue['price']**

**detail.save()**

**return Response('Data Updated Successfully ')**

**def patch(self,request,id):**

**detail    = Myproducts.objects.get(id=id)**

**uservalue = request.data**

**detail.product\_name = uservalue['name']**

**detail.code         = uservalue['code']**

**detail.price        = uservalue['price']**

**detail.save()**

**return Response('Data Updated Successfully by Patch ')**

**def delete(self,request,id):**

**detail   = Myproducts.objects.get(id=id)**

**detail.delete()**

**return Response('Deleted Successfully'**

***API OUTPUT TABLE***

|  |  |  |  |
| --- | --- | --- | --- |
| No | Method | Input | Output |
| 1 | **POST**  http://127.0.0.1:8000/inventory/products/ | **{**  **"product\_name":"Orange",**  **"code":"PRO1003",**  **"price": 300**  **}** | New Data Saved |
| 2 | Get  http://127.0.0.1:8000/inventory/products/ |  | **[**  **{**  **"product\_name": "Orange",**  **"product\_code": "PRO1001",**  **"product\_price": 300.0,**  **"product\_id": 1**  **},**  **{**  **"product\_name": "Apple",**  **"product\_code": "PRO1002",**  **"product\_price": 200.0,**  **"product\_id": 2**  **}**  **]** |
| 3 | **GET**  http://127.0.0.1:8000/inventory/products/1/ |  | {      "product\_name": "Apple",      "product\_code": "PRO1002",      "product\_price": 200.0,      "product\_id": 2  } |
| 4 | **PUT**  http://127.0.0.1:8000/inventory/products/1/ | {      "name":"Strawberry",      "code":"PRO1003",      "price": 300  } | "Data Updated Successfully " |
| 5 | **PATCH**  http://127.0.0.1:8000/inventory/products/2/ | {      "name":"Sappoto",      "code":"PRO1004",      "price": 400  } | "Data Updated Successfully by Patch " |
| 6 | **DELETE**  http://127.0.0.1:8000/inventory/products/3/ |  | "Deleted Successfully" |

**PROJECT 2**

**CODINGS**

***### Project urls.py***

**from django.contrib import admin**

**from django.urls import path,include**

**urlpatterns = [**

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

**path('inventory/', include('Inventory.urls')),**

**]**

***### Application urls.py***

**from django.contrib import admin**

**from django.urls import path**

**from .views import \***

**urlpatterns = [**

**path('products1s/', Ourproducts1.as\_view()),**

**path('products2s/', Ourproducts2.as\_view()),**

**path('products/<int:id>/', OurproductsbyID.as\_view()),**

**]**

***### Application Views.py***

**from rest\_framework .views import APIView**

**from rest\_framework .response import Response**

**from .models import \***

**from .serializers import \***

**class Ourproducts1(APIView):**

**def get(self,request):**

**full\_products = Myproducts.objects.all()**

**serialized\_products = Products\_Serializer1(full\_products,many=True).data       *## Many Records***

**return Response(serialized\_products)**

**class Ourproducts2(APIView):**

**def get(self,request):**

**full\_products = Myproducts.objects.all()**

**serialized\_products = Products\_Serializer2(full\_products,many=True).data       *## Many Records***

**return Response(serialized\_products)**

**class OurproductsbyID(APIView):**

**def get(self,request,id):**

**req\_product = Myproducts.objects.get(id=id)**

**serialized\_products = Products\_Serializer1(req\_product).data**

**return Response(serialized\_products)**

**def put(self,request,id):**

**detail = Myproducts.objects.get(id=id)**

**uservalues = request.data**

**detail.product\_name = uservalues['name']**

**detail.code = uservalues['code']**

**detail.price = uservalues['price']**

**detail.save()**

**return Response(f"Updated Product as  {uservalues['name']}")**

**def delete(self,request,id):**

**detail = Myproducts.objects.get(id=id)**

**detail.delete()**

**return Response(f"Deleted ID {id}")**

***### APP serializer.py***

**from rest\_framework import serializers**

**from .models import \***

**class Products\_Serializer1(serializers.ModelSerializer):**

**class Meta:**

**model = Myproducts**

**fields = '\_\_all\_\_'    *#For all fields in model***

**class Products\_Serializer2(serializers.ModelSerializer):**

**class Meta:**

**model = Myproducts**

**fields = ['product\_name']    *#For single Field***

***### App Models.py***

**from django.db import models**

***# Create your models here.***

**class Myproducts(models.Model):**

**product\_name = models.CharField(max\_length=200,null=True)**

**code         = models.CharField(max\_length=200,null=True)**

**price        = models.FloatField(default=0)**

**PROJECT 2 - API OUTPUT TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **Method/Url** | **Input** | **Output** |
|  | **Get**  **http://127.0.0.1:8000/inventory/products1s/** |  | **[**  **{**  **"id": 1,**  **"product\_name": "Apple",**  **"code": "PRO1003",**  **"price": 300.0**  **},**  **{**  **"id": 2,**  **"product\_name": "Orange",**  **"code": "PRO1002",**  **"price": 1800.0**  **}**  **]** |
| **2** | **GET**  **http://127.0.0.1:8000/inventory/products2s/** |  | **[**  **{**  **"product\_name": "Apple"**  **},**  **{**  **"product\_name": "Orange"**  **}**  **]** |
| **3** | **GET**  **http://127.0.0.1:8000/inventory/products/1/** |  | **{**  **"id": 1,**  **"product\_name": "Apple",**  **"code": "PRO1003",**  **"price": 300.0**  **}** |

**PROJECT 3**

*### Project urls.py*

from **django**.**contrib** import **admin**

from **django**.**urls** import **path**,**include**

urlpatterns = [

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

**path**('inventory/', **include**('Inventory.urls')),

]

***### Application Urls.py***

**from django.contrib import admin**

**from django.urls import path**

**from .views import \***

**urlpatterns = [**

***### Category with Serializer***

**path('category/',           OurCategory.as\_view()),**

**path('category/<int:id>/',  OurCategory.as\_view()),**

***## Products with serializer***

**path('ser\_products/',           Ourproducts\_serial.as\_view()),**

**path('ser\_products/<int:id>/',  Ourproducts\_serialBYID.as\_view()),**

***## Products without serializer***

**path('man\_product/',            Ourproducts\_manual.as\_view()),**

**path('man\_products/<int:id>/',  Ourproducts\_manualBYID.as\_view()),**

**]**

***### Application views.py***

**from rest\_framework .views import APIView**

**from rest\_framework . response import Response**

**from .models import \***

**from .serializers import \***

***### Category Foreignkey Connection***

**class OurCategory(APIView):**

**def post(self,request):**

**uservalues = request.data**

**ProductCategory(category\_name=uservalues['category\_name']).save()**

**return Response ("New Item Saved in Category")**

**def get(self,request):**

**categoryitems =ProductCategory.objects.all()**

**serialized\_category =  Category\_Serializer(categoryitems,many=True).data**

**return Response(serialized\_category)**

**def get(self,request,id):**

**categoryitem =ProductCategory.objects.get(id=id)**

**serialized\_category =  Category\_Serializer(categoryitem).data**

**return Response(serialized\_category)**

***### Products Serialized Method***

**class Ourproducts\_serial(APIView):**

**def post(self,request):**

**new\_product =  Product\_serializer(data=request.data)**

**if new\_product.is\_valid():**

**return Response("New Product using Serializer")**

**else:**

**return Response(new\_product.errors)**

**def get(self,request):**

**viewproducts = Myproducts.objects.all()**

**serialized\_products =  Product\_serializer(viewproducts,many=True).data**

**return Response(serialized\_products)**

**class Ourproducts\_serialBYID(APIView):**

**def get(self,request,id):**

**viewproducts = Myproducts.objects.get(id=id)**

**serialized\_products =  Product\_serializer(viewproducts).data**

**return Response(serialized\_products)**

**def put(self,request,id):**

**viewproducts = Myproducts.objects.get(id=id)**

**serialized\_data = Product\_serializer(viewproducts, data=request.data)**

**if serialized\_data.is\_valid():**

**serialized\_data.save()**

**return Response('Updated')**

**return Response('Error in Updating')**

**def delete(self,request,id):**

**product = Myproducts.objects.get(id=id)**

**product.delete()**

**return Response("Deleted")**

***### Products Manual Method***

**class Ourproducts\_manual(APIView):**

**def post(self,request):**

**uservalues = request.data**

**new\_product = Myproducts(product\_name=uservalues['product\_name'],code=uservalues['code'],price=uservalues['price'] ,category\_refrences\_id=uservalues['category'])**

**new\_product.save()**

**return Response("New Product  Created by Manual Method")**

**def get(self,request):**

**products = Myproducts.objects.all()**

**fullproducts = []**

**for each in products:**

**singleproduct = {  'product\_name' : each.product\_name, 'product\_code' : each.code, 'product\_price':     each.price,'product\_id'   : each.id, 'category\_id':each.category\_refrences\_id }**

**fullproducts.append(singleproduct)**

**return Response(fullproducts)**

**class Ourproducts\_manualBYID(APIView):**

***### GETBYID***

**def get(self,request,id):**

**detail = Myproducts.objects.get(id=id)**

**product = {  'product\_name':detail.product\_name,  'product\_code':detail.code, 'product\_price':detail.price,**

**'product\_id': detail.id , 'category\_id':detail.category\_refrences\_id}**

**return Response(product)**

**def put(self,request,id):**

**detail    = Myproducts.objects.get(id=id)**

**uservalue = request.data**

**detail.product\_name = uservalue['product\_name']**

**detail.code         = uservalue['code']**

**detail.price        = uservalue['price']**

**detail.category\_refrences\_id        = uservalue['category']**

**detail.save()**

**return Response('Data Updated Successfully ')**

**Models.py**

***## Application models.py***

**from django.db import models**

***### Parent Model so top***

**class ProductCategory(models.Model):**

**category\_name = models.CharField(max\_length=200,null=True)**

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

**return self.category\_name**

***### It deponds on Category(Child Model)***

**class Myproducts(models.Model):**

**category\_refrences = models.ForeignKey(ProductCategory,null=True,on\_delete=models.SET\_NULL)**

**product\_name = models.CharField(max\_length=200,null=True)**

**code         = models.CharField(max\_length=200,null=True)**

**price        = models.FloatField(default=0)**

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

**return self.product\_name**

**serializer.py**

***### Application Serializer.py***

**from rest\_framework import serializers**

**from .models import \***

**class Category\_Serializer(serializers.ModelSerializer):**

**class Meta:**

**model = ProductCategory**

**fields = '\_\_all\_\_'    *#For all fields in model***

**class Product\_serializer(serializers.ModelSerializer):**

**class Meta:**

**model = Myproducts**

**fields = '\_\_all\_\_'**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **URL/METHODS** | **Input** | **Output** |
| **1** | **POST**  **http://127.0.0.1:8000/inventory/category/** | **{**  **"category\_name":"vegetables"**  **}** | **"New Item Saved in Category"** |
| **2** | **Get**  **http://127.0.0.1:8000/inventory/category/** |  | **[**  **{**  **"id": 1,**  **"category\_name": "flowers"**  **},**  **{**  **"id": 2,**  **"category\_name": "fruits"**  **},**  **{**  **"id": 3,**  **"category\_name": "vegetables"**  **}**  **]** |
| **3** | **Get http://127.0.0.1:8000/inventory/category/2/** |  | **{ "id": 2,**  **"category\_name": "fruits"**  **}** |
|  | **Serialized METHOD** |  |  |
| **1** | **POST**  **http://127.0.0.1:8000/inventory/ser\_products/** | **{**  **"product\_name":"Mango",**  **"code":"PRO1002",**  **"price":2360,**  **"category":3**  **}** | **"New Product using Serializer"** |
| **2** | **GET**  **http://127.0.0.1:8000/inventory/ser\_products/** |  | **[**  **{**  **"id": 1,**  **"product\_name": "Onion",**  **"code": "PRO1001",**  **"price": 100.0,**  **"category\_refrences": 3**  **},**  **{**  **"id": 2,**  **"product\_name": "Beans",**  **"code": "PRO1003",**  **"price": 160.0,**  **"category\_refrences": 3**  **}**  **]** |
| **3** | **GET**  **http://127.0.0.1:8000/inventory/ser\_products/2/** |  | **{**  **"id": 2,**  **"product\_name": "Beans",**  **"code": "PRO1003",**  **"price": 160.0,**  **"category\_refrences": 3**  **}** |
| **4** | **PUT**  **http://127.0.0.1:8000/inventory/ser\_products/3/** | **{**  **"id": 3,**  **"product\_name": "BMW",**  **"code": "PRO1003",**  **"price": 123.65,**  **"category\_refrences": 1**  **}** | **Updated** |
| **5** | **DELETE**  **http://127.0.0.1:8000/inventory/ser\_products/2/** |  | **"Deleted"** |
|  | **Manual** |  |  |
| **1** | **POST**  **http://127.0.0.1:8000/inventory/man\_products/** | **{**  **"product\_name":"Mango",**  **"code":"PRO1002",**  **"price":2360,**  **"category":3**  **}** | **Created** |
| **2** | **GET**  **http://127.0.0.1:8000/inventory/man\_product/** |  | **[**  **{**  **"product\_name": "Onion",**  **"product\_code": "PRO1001",**  **"product\_price": 100.0,**  **"product\_id": 1,**  **"category\_id": 3**  **},**  **{**  **"product\_name": "Audi",**  **"product\_code": "PRO10023",**  **"product\_price": 123.65,**  **"product\_id": 3,**  **"category\_id": 1**  **},**  **{**  **"product\_name": "Berry",**  **"product\_code": "PRO1001",**  **"product\_price": 100.0,**  **"product\_id": 4,**  **"category\_id": 3**  **}**  **]** |
| **3** | **GET**  **http://127.0.0.1:8000/inventory/man\_products/3/** |  | **{**  **"product\_name": "Audi",**  **"product\_code": "PRO10023",**  **"product\_price": 123.65,**  **"product\_id": 3,**  **"category\_id": 1**  **}** |
| **4** | **PUT**  **http://127.0.0.1:8000/inventory/man\_products/3/** | **{**  **"id": 3,**  **"product\_name": "Audi",**  **"code": "PRO10023",**  **"price": 123.65,**  **"category": 1**  **}** | **Updated** |