Session -5

INSTALLATION

1.Django Installation:

- **Install Django via pip**:
  After installing Python, you can install Django using pip:

  ```
  #pip install django
  ```

- **Verify Installation**:
  Check the installation by running

  ```
  #django-admin --version
  ```

2. Set Up Django Models

3. Set Up Django Serializers

Definition:
Serializers in DRF convert complex data types (like Django models) into Python data types that can then be easily rendered into JSON or XML.

Create serializers in serializers.py file

```
from rest_framework import serializers

from . models import User
class UserSerializer(serializers.ModelSerializer):

  class Meta:

    model = User

    fields = ['id', 'name', 'last_name', 'email']
```

Then checkout the types of Https Methods (HTTP response status codes - HTTP | MDN)

Now , Let's Create Api's

```python
from django.shortcuts import render
from django.http import JsonResponse, HttpResponse
from .models import User
from . serializers import UserSerializer
import json
from django.views.decorators.csrf import csrf_exempt

@csrf_exempt
def user_list(request):
  if request.method == 'GET':   #for get api's
    users = User.objects.all()
    serializer = UserSerializer(users,many=True)
    return JsonResponse(serializer.data,safe=False)
  elif request.method == 'POST': #for POST api's
    data = json.loads(request.body)
    serializer = UserSerializer(data=data)
    if serializer.is_valid():
      serializer.save()
      return JsonResponse(serializer.data ,status = 201)
    return JsonResponse(serializer.errors,status=400)

@csrf_exempt
def user_detail(request,pk):
  try:
```

```
    user = User.objects.get(pk=pk)

except User.DoesNotExist:
    return JsonResponse({'error':'User not found '}, status = 404)

if request.method == 'GET':
        serializer = UserSerializer(user)
        return JsonResponse(serializer.data)

elif request.method == 'PUT': #For PUT api's
    data = json.loads(request.body)
    serializer = UserSerializer(user,data=data)
    if serializer.is_valid():
        serializer.save()
        return JsonResponse(serializer.data)
    else:
        return JsonResponse(serializer.errors,status = 400)

elif request.method == 'DELETE':  #For DELETE API
    user.delete()
    return HttpResponse("Successfully Deleted", status=204)
else:
    return HttpResponse(status=405)
```

Then , create urls.py

```
from . import views

from django.urls import path

from .views import user_list,user_detail

#url for app

urlpatterns = [

  path('homepage/', views.User),

  path('users/',user_list,name='user-list'),

  path('users/<int:pk>/', user_detail, name='user-detail'),
```

Test the API endpoints using Postman or curl:

- **GET /users/** to list all users.
- **POST /users/** to create a new user.
- **GET //users/<id>/** to retrieve a specific user.
- **PUT /users/<id>/** to update a user.
- **PATCH /users/<id>/** to partially update a user.
- **DELETE /users/<id>/** to delete a user.