



A.Y. 2022-2023

Subject: Python

SAP ID: 60004220253 – Devansh Mehta

Experiment No. 12

Aim: Implement a web application using Django Framework and understand CRUD functionality.

Code:

urls.py (project)

```
from django.contrib import admin
from django.urls import path,include
from api import views
urlpatterns = [
    path('studentAPI/',views.StudentListCreate.as_view()),
    path('studentAPI/<int:pk>',views.StudentRetrieveUpdateDestroy.as_view())
]
```

Models.py

```
from django.db import models
class Student(models.Model):
    name = models.CharField(max_length=100)
    roll = models.IntegerField()
    city = models.CharField(max_length=100)
```

Serializer.py

```
from rest_framework import serializers
from .models import Student

class StudentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Student
        fields = '__all__'
    def validate_name(self,value):
        if (all(chr.isalpha() or chr.isspace() for chr in value)):
            return value
        raise serializers.ValidationError(
            "Name must have alphabets only")
```

Views.py

```
from .models import Student
from .serializers import StudentSerializer
from rest_framework.generics import ListCreateAPIView,RetrieveUpdateDestroyAPIView
```



A.Y. 2022-2023

Subject: Python

SAP ID: 60004220253 – Devansh Mehta

```
class StudentListCreate(ListCreateAPIView):
```

```
    queryset = Student.objects.all()
```

```
    serializer_class = StudentSerializer
```

```
class StudentRetrieveUpdateDestroy(RetrieveUpdateDestroyAPIView):
```

```
    queryset = Student.objects.all()
```

```
    serializer_class = StudentSerializer
```

Output:

Get Data

```
HTTP 200 OK
```

```
Allow: GET, POST, HEAD, OPTIONS
```

```
Content-Type: application/json
```

```
Vary: Accept
```

```
[
  {
    "id": 1,
    "name": "Devansh",
    "roll": 101,
    "city": "mumbai change"
  },
  {
    "id": 22,
    "name": "Rohit",
    "roll": 250,
    "city": "Ranchi"
  }
]
```

Post Data

Name

Virat

Roll

300

City

Mumbai



A.Y. 2022-2023

Subject: Python

SAP ID: 60004220253 – Devansh Mehta

New data after inserting

Vary: Accept

```
[
  {
    "id": 1,
    "name": "Devansh",
    "roll": 101,
    "city": "mumbai change"
  },
  {
    "id": 22,
    "name": "Rohit",
    "roll": 250,
    "city": "Ranchi"
  },
  {
    "id": 25,
    "name": "Virat",
    "roll": 300,
    "city": "Mumbai"
  }
]
```

Updating Existing Data

Name	<input type="text" value="Virat"/>
Roll	<input type="text" value="300"/>
City	<input type="text" value="Delhi"/>



A.Y. 2022-2023

Subject: Python

SAP ID: 60004220253 – Devansh Mehta

New data after updation

```
[  
  {  
    "id": 1,  
    "name": "Devansh",  
    "roll": 101,  
    "city": "mumbai change"  
  },  
  {  
    "id": 22,  
    "name": "Rohit",  
    "roll": 250,  
    "city": "Ranchi"  
  },  
  {  
    "id": 25,  
    "name": "Virat",  
    "roll": 300,  
    "city": "Delhi"  
  }  
]
```

Deleting data with ID 25

```
[  
  {  
    "id": 1,  
    "name": "Devansh",  
    "roll": 101,  
    "city": "mumbai change"  
  },  
  {  
    "id": 22,  
    "name": "Rohit",  
    "roll": 250,  
    "city": "Ranchi"  
  }  
]
```