

ORMFilters(OBJECT RELATIONAL MAPPER)

- ✓ ORM means object-relational mapper.
- ✓ Transmit data between a relational database and application model.
- ✓ Not need write any SQL.

MODEL**EMPLOYEE**

| NO | NAME | SALARY | CITY |
|-----|-------|--------|---------|
| 101 | ARJUN | 50000 | TRICHY |
| 102 | LEO | 50000 | CHENNAI |
| 103 | ABDUL | 20000 | TRICHY |

TABLES & RECORDS**PYTHON****OBJECTS****QUERYSET**

- ✓ A Query set is a **list of objects of a model**.
- ✓ **To filter and arrange our data.**
- ✓ Use queries to retrieve data from the database.

----- WEB DEVELOPMENT -----

1. Create a Project with Forms & Tables.

Select student to change | Djani | X | Student Informations | X | INSERT RECORD | X | +

← → C ① 127.0.0.1:8000 insert

INSERT STUDENT INFORMATION

Sno:

Sname:

Salary:

Saddress:

SUBMIT

2. Verify the data in admin panel.

Select student to change | Djani | X | Student Informations | X | INSERT RECORD | X | +

← → C ① 127.0.0.1:8000/admin/apps/student/

Django administration

WELCOME, ADMIN | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home | Apps | Students

Start typing to filter

| Action | SNO | SNAME | SSALARY | SADDRESS |
|--------------------------|-----|-------|---------|----------|
| <input type="checkbox"/> | 104 | d | 30000.0 | chennai |
| <input type="checkbox"/> | 103 | c | 20000.0 | TRICHY |
| <input type="checkbox"/> | 102 | raj | 30000.0 | chennai |
| <input type="checkbox"/> | 101 | arjun | 20000.0 | trichy |

5 students

ADD STUDENT

WEB DEVELOPMENT

3. Verify the data.

The screenshot shows a web browser window with the title "Select student to change | Django". Below it is a "Student Information" link and an "INSERT RECORD" button. The URL is "127.0.0.1:8000/home". The main content is a table titled "STUDENTS INFORMATION" with the following data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 102 | raj | 30000.0 | chennai |
| 103 | c | 20000.0 | TRICHY |
| 103 | c | 20000.0 | TRICHY |
| 104 | d | 30000.0 | chennai |

4. Enter the **greater than** command in View file

The code editor shows a file named "views.py" with the following content:

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(ssalary__gt=20000)
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

5. Verify the result



6. Enter **Greater than and Equal** command in View file.

views.py

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(ssalary__gte=20000)
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

WEB DEVELOPMENT

7. Verify the result.

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 102 | raj | 30000.0 | chennai |
| 103 | c | 20000.0 | TRICHY |
| 104 | c | 20000.0 | TRICHY |
| 104 | d | 30000.0 | chennai |

8. Enter the **Less than** command in View file

```
views.py

from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(ssalary__lt=30000)
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

9. Verify the result.

The screenshot shows a Django admin interface titled "STUDENTS INFORMATION". At the top, there are tabs for "Select student to change (Django)", "Student Informations", "INSERT RECORD", and a plus sign icon. Below the tabs, the URL "127.0.0.1:8000/home/" is visible. The main content area displays a table with four columns: "STUDENT NO", "STUDENT NAME", "STUDENT SALARY", and "STUDENT ADDRESS". The table contains three rows of data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 103 | c | 20000 0 | TRICHY |
| 103 | c | 20000 0 | TRICHY |

10. Enter the **Less than and Equal** command in View file

The screenshot shows a Python code editor with a file named "views.py". The code defines a view function "home" that filters student records based on salary. It also defines form handling and final output rendering functions.

```
views.py
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(ssalary__lte=30000)
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

11. Verify the result.

The screenshot shows a Django admin interface with the title "STUDENTS INFORMATION". There are two tabs at the top: "Select student to change" and "Student Information". Below the tabs, there is a URL bar with "127.0.0.1:8000/home". The main content area displays a table with the following data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 102 | raj | 30000.0 | chennai |
| 103 | c | 20000.0 | TRICHY |
| 103 | c | 20000.0 | TRICHY |
| 104 | d | 30000.0 | chennai |

12. Enter the **Exact** command in View file

The screenshot shows a code editor with a file named "views.py". The code contains Python code for a Django application, specifically for a "Student" model. The code includes functions for displaying student information and saving new student forms. The "exact" command is used in the filter method to search for students with a specific address.

```
views.py
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(saddress__exact='trichy')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

13. Verify the result.

The screenshot shows a Django admin interface titled "STUDENTS INFORMATION". At the top, there are tabs for "Select student to change" (highlighted), "Student Informations", "INSERT RECORD", and "127.0.0.1:9000/home". Below the tabs is a table with four columns: "STUDENT NO", "STUDENT NAME", "STUDENT SALARY", and "STUDENT ADDRESS". A single row is displayed with the values: 101, arjun, 20000.0, trichy.

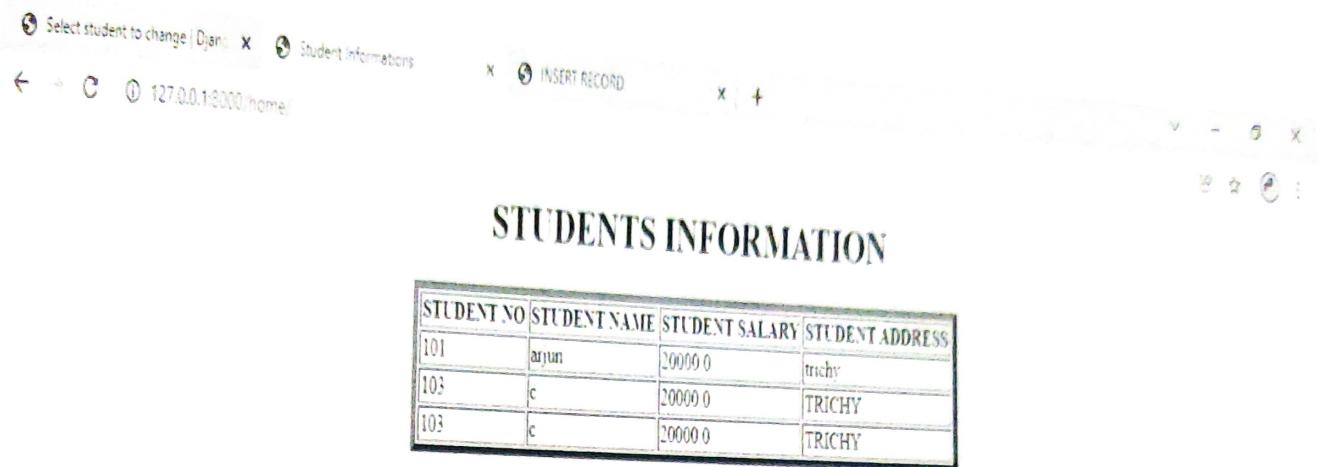
| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |

14. Enter the **iexact** command in View file

The screenshot shows a code editor with a file named "views.py". The code defines a view function "home" that filters students by address using the "iexact" lookup. It also defines a "forms" function and two helper functions, "final" and "render".

```
views.py
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(saddress__iexact='trichy')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

15. Verify the result.



The screenshot shows a browser window with the URL `127.0.0.1:8000/home/`. The page title is "STUDENTS INFORMATION". It displays a table with four columns: STUDENT NO, STUDENT NAME, STUDENT SALARY, and STUDENT ADDRESS. The data in the table is as follows:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 103 | c | 20000 0 | TRICHY |
| 103 | c | 20000 0 | TRICHY |

16. Enter the **Contains** command in View file

```
views.py
```

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sname__contains='ar')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

17. Verify the result.

Select student to change | Dan X Student Information
← C ① 127.0.0.1:8000/home X +

STUDENTS INFORMATION

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 | trichy |

18. Enter the **icontains** command in View file

```
views.py X
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sname_icontains='ar')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

19. Verify the result.

The screenshot shows a Django admin interface for 'Student informations'. At the top, there are tabs for 'Select student to change' (Django), 'Student informations', and 'INSERT RECORD'. Below the tabs, there are navigation icons: back, forward, search, and refresh. The URL '127.0.0.1:8000/home' is visible. The main title is 'STUDENTS INFORMATION'. A table displays student data with columns: STUDENT NO, STUDENT NAME, STUDENT SALARY, and STUDENT ADDRESS. One row is shown: STUDENT NO 101, STUDENT NAME arjun, STUDENT SALARY 20000.0, and STUDENT ADDRESS trichy.

20. Enter the **filter** command in View file

```
views.py
```

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sno_in=(101,102))
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

21. Verify the result.

The screenshot shows a web application interface. At the top, there are two tabs: "Select student to change: Djano" and "Student information". Below the tabs, there are navigation icons: back, forward, search, and refresh. The URL in the address bar is "127.0.0.1:8000/home". The main content area is titled "STUDENTS INFORMATION" and contains a table with four columns: "STUDENT NO", "STUDENT NAME", "STUDENT SALARY", and "STUDENT ADDRESS". The table has two rows of data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 102 | raj | 30000.0 | chennai |

22. Enter the **Startwith** command in View file

```
views.py
```

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sname_startswith='a')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

23. Verify the result.

The screenshot shows a web browser window titled "Student informations". The address bar indicates the URL is "127.0.0.1:8000/home". The main content area is titled "STUDENTS INFORMATION" and displays a table with four columns: STUDENT NO, STUDENT NAME, STUDENT SALARY, and STUDENT ADDRESS. The table contains two rows of data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 105 | abdul | 10000 0 | combatore |

24. Enter the **startswith** command in View file

The screenshot shows a code editor with a file named "views.py". The code defines a view function "home" that filters students whose name starts with 'a'. It also defines a form "StudentForm" and a final view "final".

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sname__startswith='a')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

25. Verify the result.

The screenshot shows a Django admin interface titled "STUDENTS INFORMATION". At the top, there are tabs for "Select student to change (Django)", "Student informations", "INSERT RECORD", and a "+" button. Below the tabs, the URL "127.0.0.1:8000/home/" is visible. The main content area displays a table with four columns: STUDENT NO, STUDENT NAME, STUDENT SALARY, and STUDENT ADDRESS. The table contains three rows of data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 105 | abdul | 10000 0 | coombatore |
| 106 | ABC | 30000 0 | coombatore |

26. Enter the **endswith** command in View file

The screenshot shows a code editor window with a file named "views.py". The code contains Python functions for handling requests and rendering templates. The "home" function filters students by name ending with 'n' and returns a rendered template. The "forms" function handles POST requests, creates a form, validates it, and saves the data if valid. The "final" function renders a template after saving. The code uses Django's shortcuts and models.

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sname__endswith='n')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

27. Verify the result.

The screenshot shows the Django admin interface for a 'Student' model. At the top, there are tabs for 'Select student to change', 'Django', 'Student informations', and 'INSERT RECORD'. Below the tabs, there are buttons for back, forward, search, and refresh. The URL '127.0.0.1:8000/admin/' is visible. The main title is 'STUDENTS INFORMATION'. A table displays student data with columns: STUDENT NO, STUDENT NAME, STUDENT SALARY, and STUDENT ADDRESS. One row is present with values: 101, arjun, 20000.0, trichy.

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |

28. Enter the **iendswith** command in View file

```
views.py
```

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(sname__iendswith='n')
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

29. Verify the result.

The screenshot shows a Django admin interface for 'Student Information'. At the top, there are tabs for 'Select student to change' (active), 'Django', 'Student Information', and 'INSERT RECORD'. Below the tabs, there's a URL bar with '127.0.0.1:8000/home/'. The main title is 'STUDENTS INFORMATION'. A table with four columns ('STUDENT NO', 'STUDENT NAME', 'STUDENT SALARY', 'STUDENT ADDRESS') displays one row: '101' in STUDENT NO, 'arjun' in STUDENT NAME, '20000.0' in STUDENT SALARY, and 'Inchy' in STUDENT ADDRESS.

30. Enter the **range** command in View file

views.py

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.filter(ssalary__range=(10000,25000))
    return render(request,'appfile/home.html',{'s':stud})
def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})
def final(request):
    return render(request,'appfile/final.html')
```

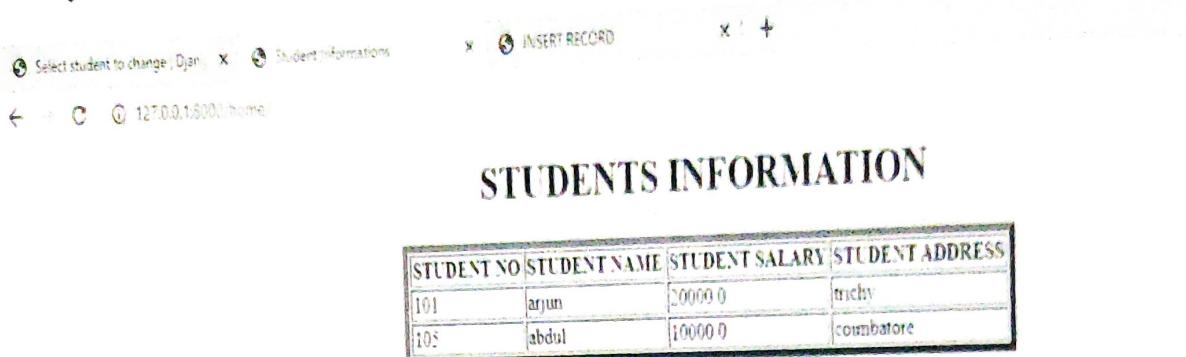
31. Verify the result.

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 103 | c | 20000.0 | TRICHY |
| 103 | c | 20000.0 | TRICHY |
| 105 | abdul | 10000.0 | combatore |

32. Enter the **and (&)** command in View file

```
views.py  
from django.shortcuts import render  
from apps.models import Student  
from apps.forms import StudentForm  
def home(request):  
    stud=Student.objects.filter(ssalary__range=(10000,25000))&Student.objects.filter(sname__startswith='a')  
    return render(request,'appfile/home.html',{'s':stud})  
def forms(request):  
    form=StudentForm()  
    if request.method=="POST":  
        form=StudentForm(request.POST)  
        if form.is_valid():  
            form.save()  
        return final(request)  
    return render(request,'appfile/form.html',{'form':form})  
  
def final(request):  
    return render(request,'appfile/final.html')
```

33. Verify the result.



The screenshot shows a web browser window titled "Student Informations". The address bar indicates the URL is "127.0.0.1:800/home". The main content area displays a table titled "STUDENTS INFORMATION" with four columns: STUDENT NO, STUDENT NAME, STUDENT SALARY, and STUDENT ADDRESS. The table contains two rows of data:

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 102 | abdul | 10000 0 | combatore |

34. Enter the **orderby** command in View file

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    stud=Student.objects.all().order_by('-ssalary')
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

35. Verify the result.



| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 110 | leo | 50000 0 | coimbator |
| 102 | raj | 30000 0 | chennai |
| 101 | arjun | 30000 0 | trichy |
| 104 | abdul | 10000 0 | trichy |

36. Enter the **union** command in View file

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm
def home(request):
    qs1=Student.objects.filter(ssalary__lt=25000)
    qs2=Student.objects.filter(sname__startswith='a')
    stud=qs1.union(qs2)
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

37. Verify the result.

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | michy |
| 104 | abdul | 10000 0 | michy |

38. Enter the **inserting** command in View file

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm

def home(request):
    s=Student(sname='murugesh',ssalary=15000,sno=105,saddress='chennai')
    s.save()
    stud=Student.objects.all()
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

39. Verify the result.

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 102 | raj | 30000 0 | chennai |
| 110 | leo | 50000 0 | coimbatore |
| 104 | abdul | 10000 0 | trichy |
| 105 | murugesh | 15000 0 | chennai |
| 105 | murugesh | 15000 0 | chennai |

40. Enter the **create** command in View file

```
views.py           *  models.py           *
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm

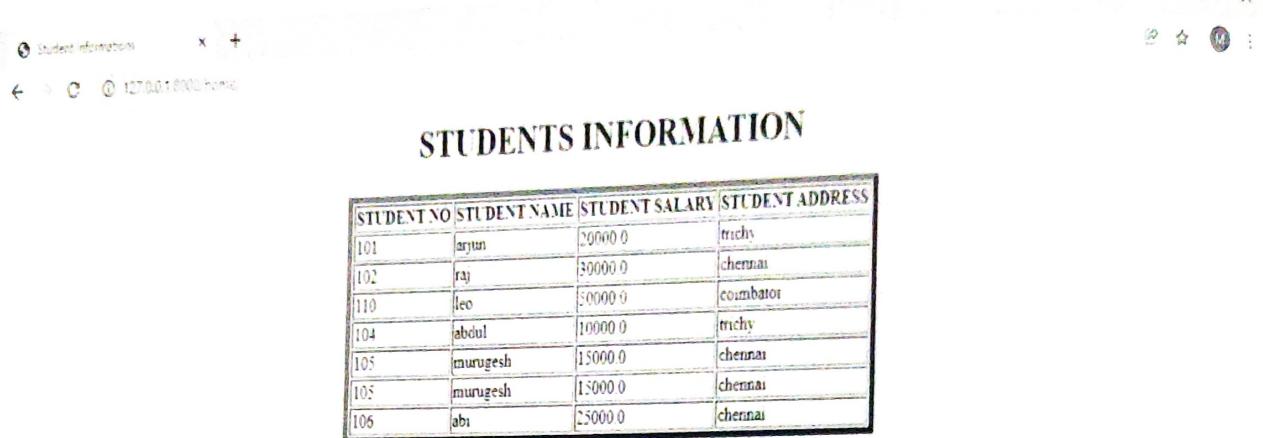
def home(request):
    s=Student.objects.create(sname='abi',ssalary=25000,sno=106,saddress='chennai')
    stud=Student.objects.all()
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

WEB DEVELOPMENT

41. Verify the result.



| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 102 | raj | 30000.0 | chennai |
| 110 | leo | 50000.0 | coimbatore |
| 104 | abdul | 10000.0 | trichy |
| 105 | murugesh | 15000.0 | chennai |
| 105 | murugesh | 15000.0 | chennai |
| 106 | abi | 25000.0 | chennai |

42. Enter the **bulk create** command in View file

```
views.py          *  models.py
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm

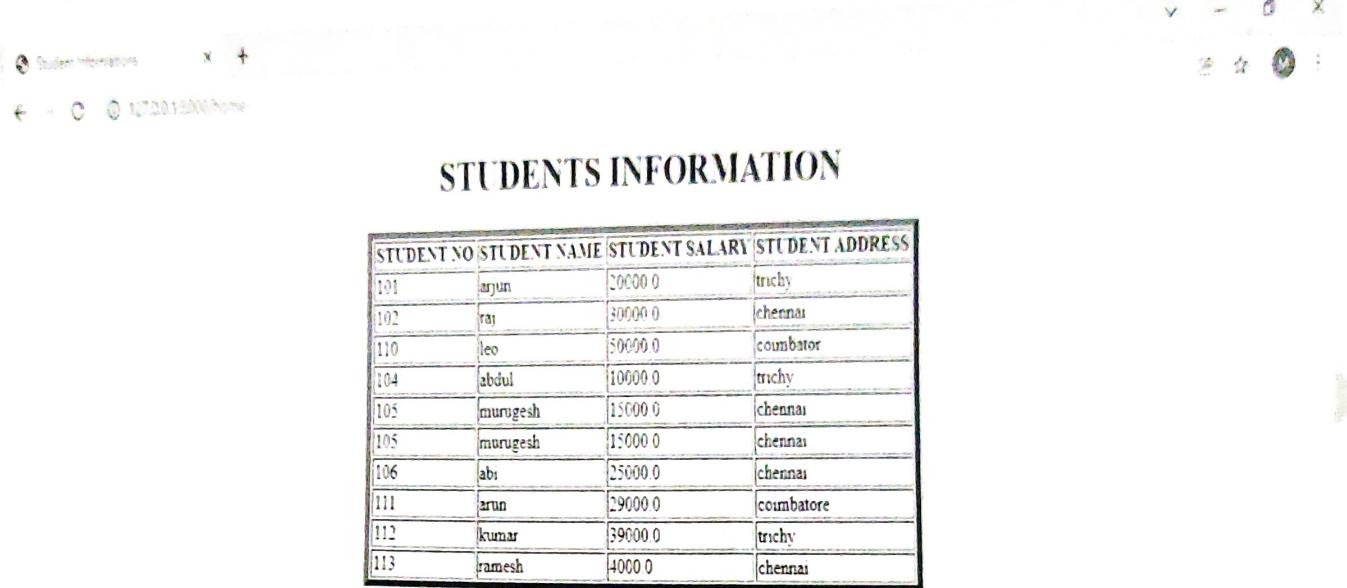
def home(request):
    s1=Student(sname='arun',ssalary=29000,sno=111,saddress='coimbatore')
    s2=Student(sname='kumar',ssalary=39000,sno=112,saddress='trichy')
    s3=Student(sname='ramesh',ssalary=40000,sno=113,saddress='chennai')
    Student.objects.bulk_create([s1,s2,s3])
    stud=Student.objects.all()
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

WEB DEVELOPMENT

43. Verify the result.



| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 102 | raj | 30000.0 | chennai |
| 110 | leo | 50000.0 | cumbator |
| 104 | abdul | 10000.0 | trichy |
| 105 | murugesh | 15000.0 | chennai |
| 105 | murugesh | 15000.0 | chennai |
| 106 | abi | 25000.0 | chennai |
| 111 | arun | 29000.0 | cumbatore |
| 112 | kumar | 39000.0 | trichy |
| 113 | ramesh | 40000.0 | chennai |

44. Enter the **get** command in View file

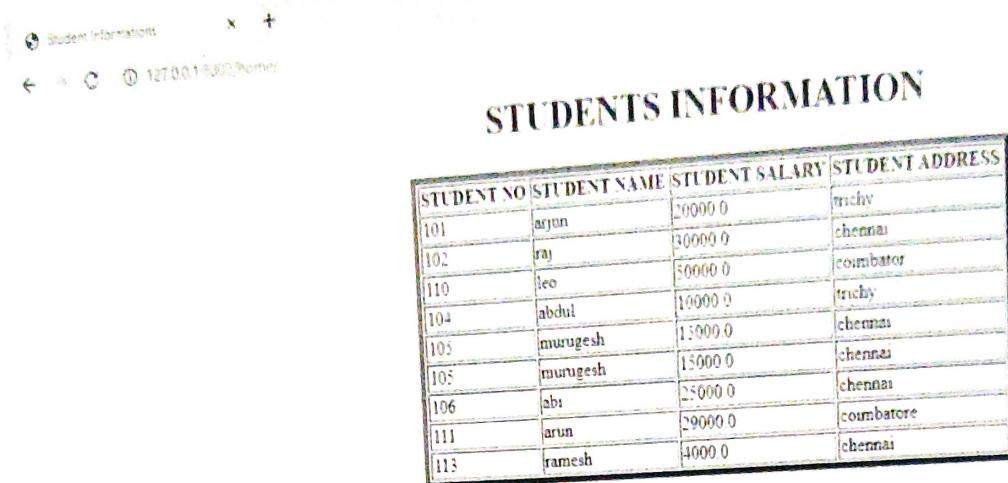
```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm

def home(request):
    s=Student.objects.get(sname='kumar')
    s.delete()
    stud=Student.objects.all()
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

45. Verify the result.



| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000 0 | trichy |
| 102 | raj | 30000 0 | chennai |
| 110 | leo | 50000 0 | combator |
| 104 | abdul | 19000 0 | trichy |
| 105 | murugesh | 15000 0 | chennai |
| 106 | murugesh | 15000 0 | chennai |
| 106 | abi | 25000 0 | chennai |
| 111 | arun | 29000 0 | combatore |
| 113 | ramesh | 40000 0 | chennai |

46. Enter the **delete** command in View file

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm

def home(request):
    s=Student.objects.filter(ssalary__gt=20000)
    s.delete()
    stud=Student.objects.all()
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
    return final(request)
return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 104 | abdul | 10000.0 | trichy |
| 105 | murugesh | 15000.0 | chennai |
| 105 | murugesh | 15000.0 | chennai |
| 113 | ramesh | 4000.0 | chennai |

48. Enter the **change** command in View file

```
from django.shortcuts import render
from apps.models import Student
from apps.forms import StudentForm

def home(request):
    s=Student.objects.get(sno=104)
    print(s.sname)
    s.sname='rajesh'
    s.save()
    stud=Student.objects.all()
    return render(request,'appfile/home.html',{'s':stud})

def forms(request):
    form=StudentForm()
    if request.method=="POST":
        form=StudentForm(request.POST)
        if form.is_valid():
            form.save()
        return final(request)
    return render(request,'appfile/form.html',{'form':form})

def final(request):
    return render(request,'appfile/final.html')
```

49. Verify the result.



STUDENTS INFORMATION

| STUDENT NO | STUDENT NAME | STUDENT SALARY | STUDENT ADDRESS |
|------------|--------------|----------------|-----------------|
| 101 | arjun | 20000.0 | trichy |
| 104 | rajesh | 10000.0 | trichy |
| 105 | murugesh | 15000.0 | chennai |
| 105 | murugesh | 15000.0 | chennai |
| 113 | ramesh | 4000.0 | chennai |