CRUD PROJECT

- **python -m django startproject crud**
- **cd crud**
- **python manage.py startapp database**
- **install in setting.py**
- **create templates**

  database>templates>base.html, addshow.html, updatestudent.html

  base.html

```
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" contenr="width=device-width,initial-scale=1.0">
        <title> crud - VISHAL</title>

    </head>
    <body>
    <div>
        <h1 style="text-align:center">Function based View Model Form Crud Project_VHA</h1>
        {% block content %}

        {% endblock content %}
    </div>
    </body>
</html>
```

Addshow.html

```
{% extends 'database/base.html' %}
{% block content %}
        <h4>add NEW STUDENT</h4>
{% endblock content %}
```

Updatestudent.html

```
{% extends 'database/base.html' %}
{% block content %}
        <h4>update NEW STUDENT</h4>
{% endblock content %}
```

- **For template render write views.py(database)**

```
from django.shortcuts import render
# Create your views here.
def add_show(request):
    return render(request,'database/addshow.html')
```

- **For server write in urls.py (crud)**

```
rom django.contrib import admin
from django.urls import path
from database import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views.add_show,name="addandshow")
]
```

- **python manage.py runserver**

- **For making a table in admin make a class in models.py(database)**

```
from django.db import models

# Create your models here.
class User(models.Model):
    roll=models.IntegerField()
    name=models.CharField(max_length=70)
    division=models.CharField(max_length=70)
```

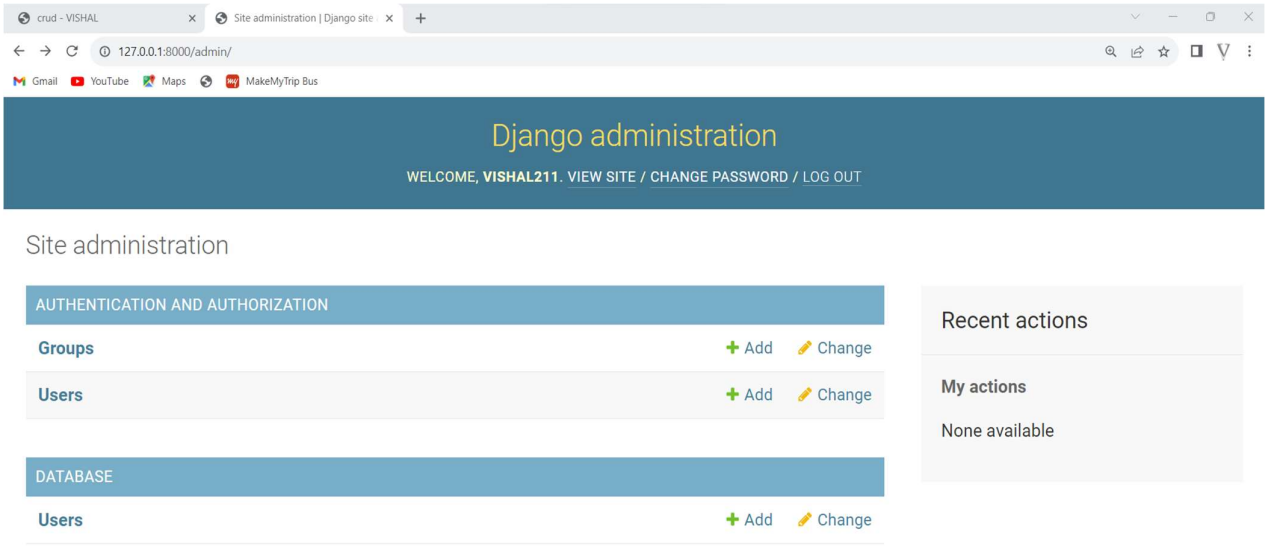
- **For Register your models in admin.py (database)**

```
from django.contrib import admin
from .models import User
# Register your models here.
@admin.register(User)
class UserAdmin(admin.ModelAdmin):
    list_display = ('id','roll','name','division')
```

- **For query generate**
  python manage.py makemigrations
- **For query execute (making table)**
  Python manage.py migrate
- **For admin login createsuperuser**
  Python manage.py createsuperuser

```
(env) D:\project\CRUD\crud>python manage.py createsuperuser
Username (leave blank to use 'vishal'): vishal211
Email address: v@gmail.com
Password:
Password (again):
The password is too similar to the username.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

- **python manage.py runserver**



- **for data entry make a form in forms.py**

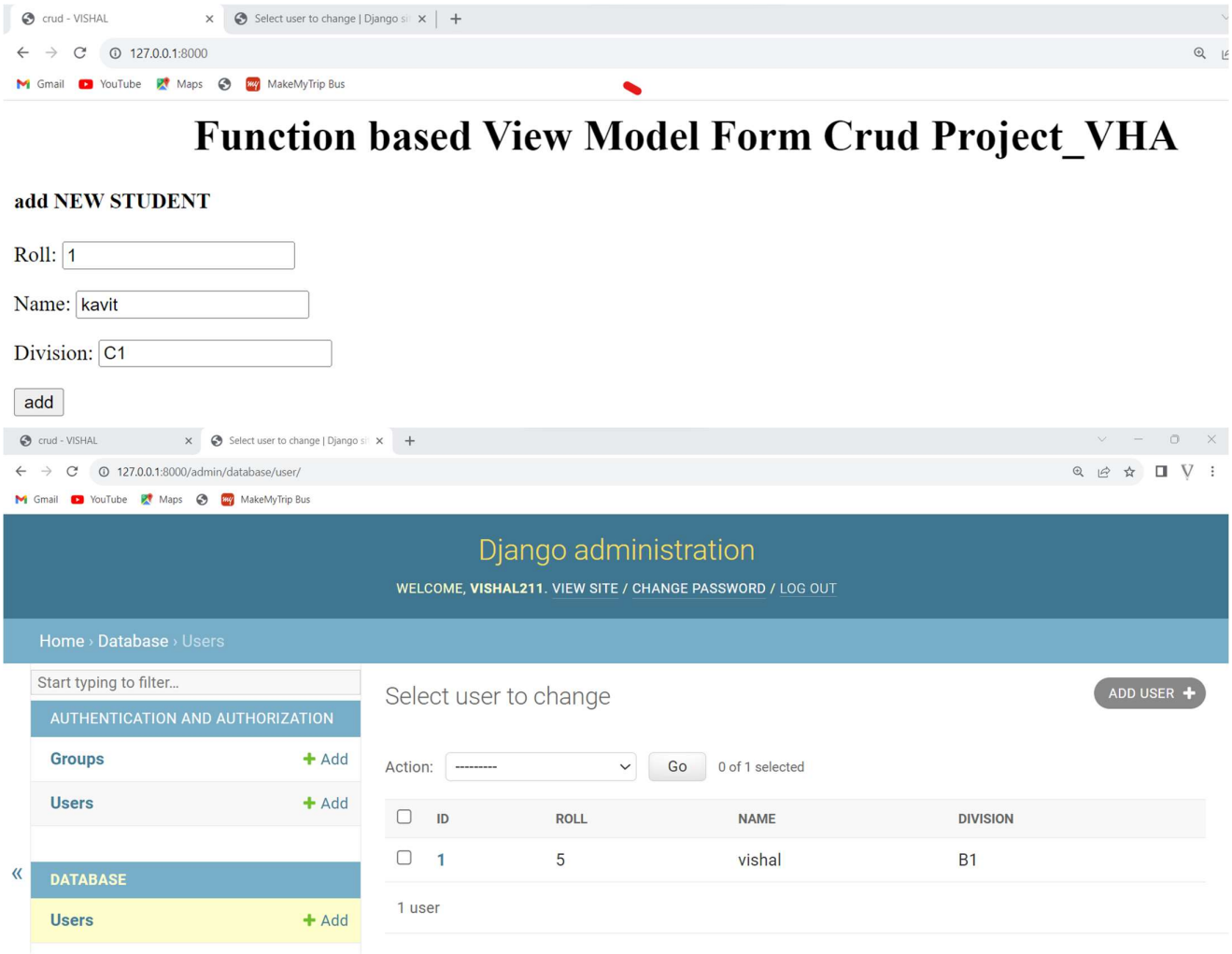
```python
from django.core import validators
from django import forms
from .models import User

class StudentRegistration(forms.ModelForm):
 class Meta:
  model = User
  fields = ['roll', 'name', 'division']
```

- **For User model and StudentRegistration connectivity in views.py**

```python
from django.shortcuts import render, HttpResponseRedirect
from .forms import StudentRegistration
from .models import User
# Create your views here.
def add_show(request):
 if request.method == 'POST':
  fm = StudentRegistration(request.POST)
  if fm.is_valid():
   r = fm.cleaned_data['roll']
   n = fm.cleaned_data['name']
   d = fm.cleaned_data['division']
```

```
  reg = User(roll=r, name=n, division=d)
  reg.save()
  fm = StudentRegistration()
else:
 fm = StudentRegistration()
stud = User.objects.all()
return render(request, 'database/addshow.html', {'form':fm, 'stu':stud})
```

➢ Output

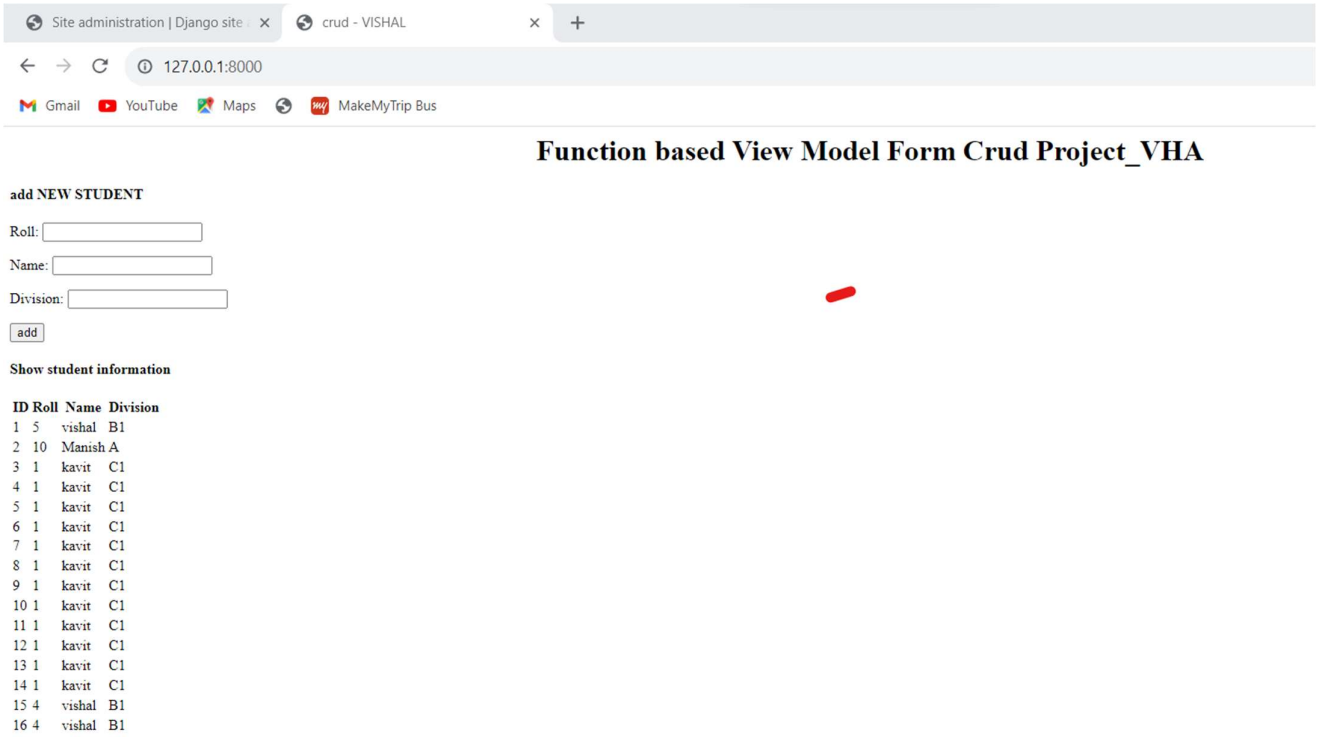


Changing adduser.html code for show add list

```
{% extends 'database/base.html' %}
{% block content %}
<div>
<div align="left">
        <h4>add NEW STUDENT</h4>
        <form action="" method="post" >
        {% csrf_token %}
```

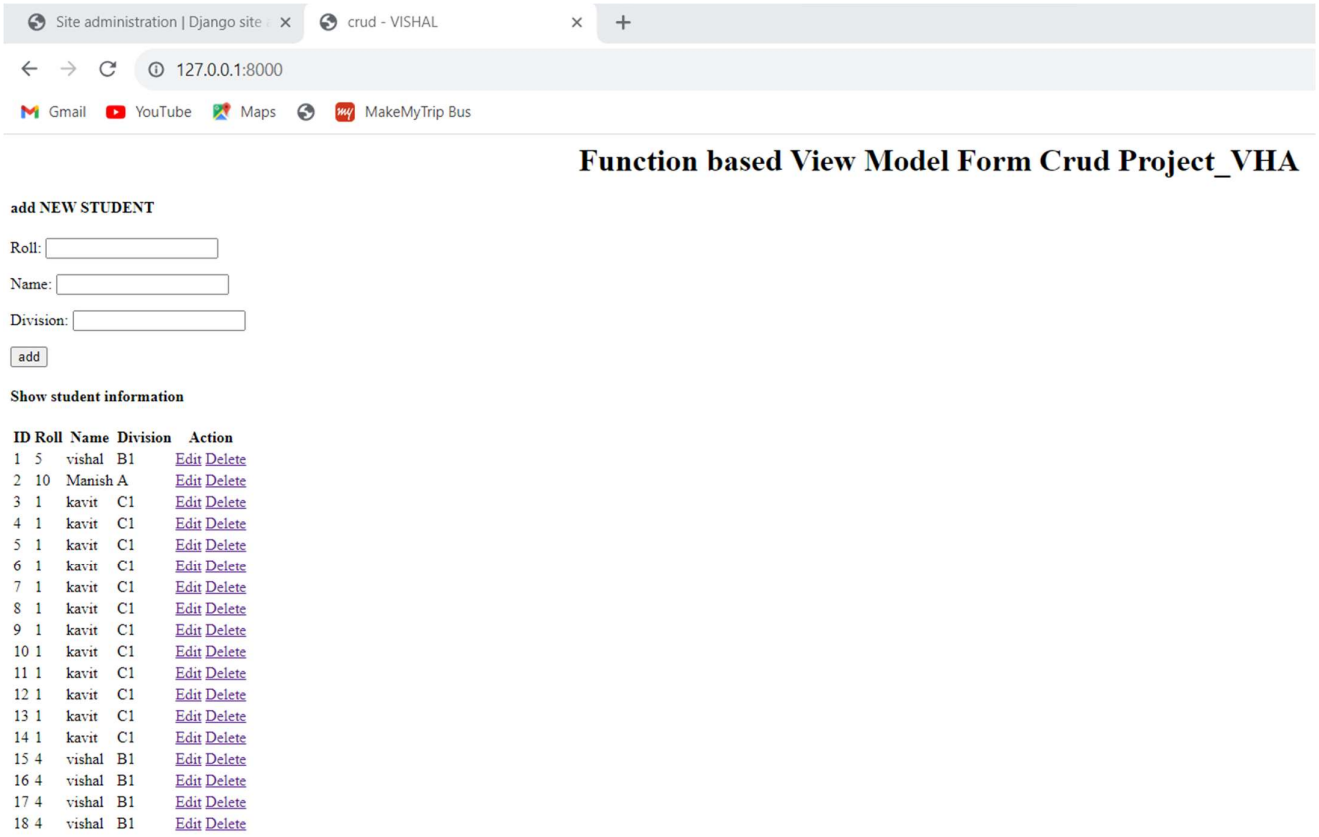
```
        {{form.as_p}}
        <input type="submit" value="add">
        </form>
</div>
<div>
        <h4>Show student information</h4>
        {% if stu %}
        <table>
                <tr>
                <th>ID</th>
                <th>Roll</th>
                <th>Name</th>
                <th>Division</th>
        </tr>
        {% for st in stu %}
        <tr>
        <td>{{st.id}}</td>
        <td>{{st.roll}}</td>
        <td>{{st.name}}</td>
        <td>{{st.division}}</td>
        </tr>
        {% endfor %}
        </table>
        {% else %}
        <h4> No Data </h4>
        {% endif %}
</div>
</div>
{% endblock content %}
```

Output:



For add two button edit and delete for each user change in addshow.html(only add one column action)

```
<table>
                <tr>
                <th>ID</th>
                <th>Roll</th>
                <th>Name</th>
                <th>Division</th>
                <th>Action<th>
        </tr>
        {% for st in stu %}
        <tr>
        <td>{{st.id}}</td>
        <td>{{st.roll}}</td>
        <td>{{st.name}}</td>
        <td>{{st.division}}</td>
        <td>
                <a href="#">Edit</a>
                <a href="#">Delete</a>
        </td>
        </tr>
        {% endfor %}
</table>
```



- **After that first write delete code**

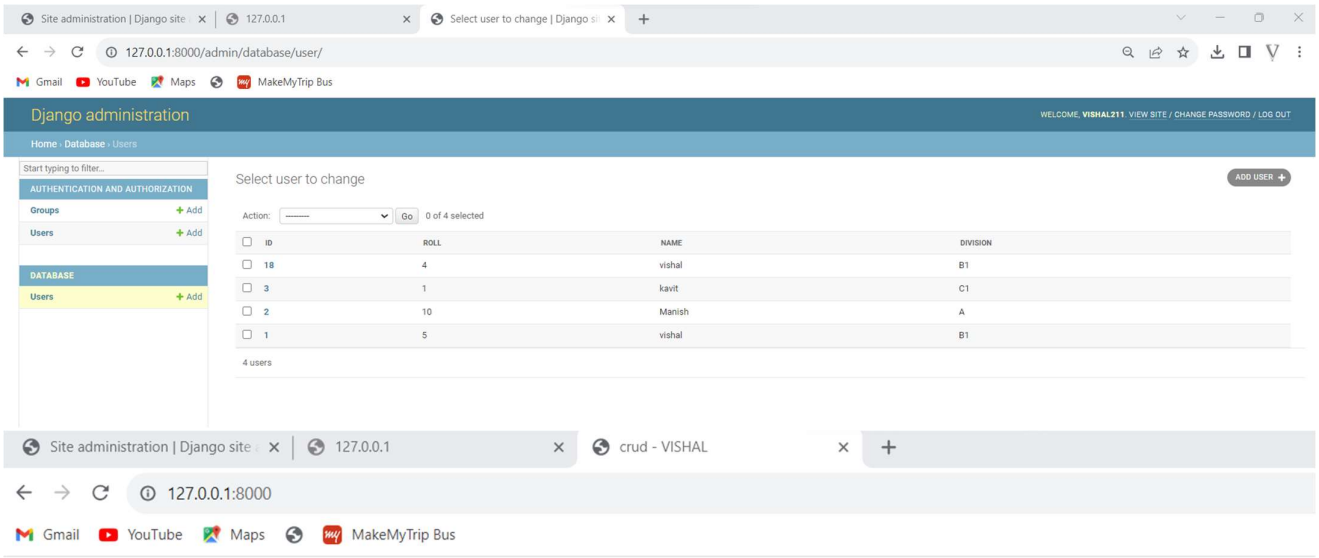
```
# This Function will Delete
def delete_data(request, id):
 if request.method == 'POST':
  pi = User.objects.get(pk=id)
  pi.delete()
  return HttpResponseRedirect('/')
```

- **Give the url of delete_data function:(make a dynamic url)**

```
from django.contrib import admin
from django.urls import path
from database import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views.add_show,name="addandshow"),
    path('delete/<int:id>/',views.delete_data,name='deletedata')

]
```

- **How to get the URL using the delete button for that make a form instead of <a> (post method is more secure than get) in addshow.html**

```
<form action="{% url 'deletedata' st.id %}" method="post">
                        {% csrf_token %}
                        <input type="submit" value="Delete">
```



Function based View Model Form Crud Project_VHA

add NEW STUDENT

Roll:

Name:

Division:

add

Show student information

ID	Roll	Name	Division	Action
1	5	vishal	B1	Edit Delete
2	10	Manish	A	Edit Delete
3	1	kavit	C1	Edit Delete
18	4	vishal	B1	Edit Delete

For update make views.py

```
 This Function will Update/Edit
def update_data(request, id):
 if request.method == 'POST':
  pi = User.objects.get(pk=id)
  fm = StudentRegistration(request.POST, instance=pi)
  if fm.is_valid():
   fm.save()
 else:
  pi = User.objects.get(pk=id)
```

```
  fm = StudentRegistration(instance=pi)
 return render(request, 'database/updatestudent.html', {'form':fm})
```

for url

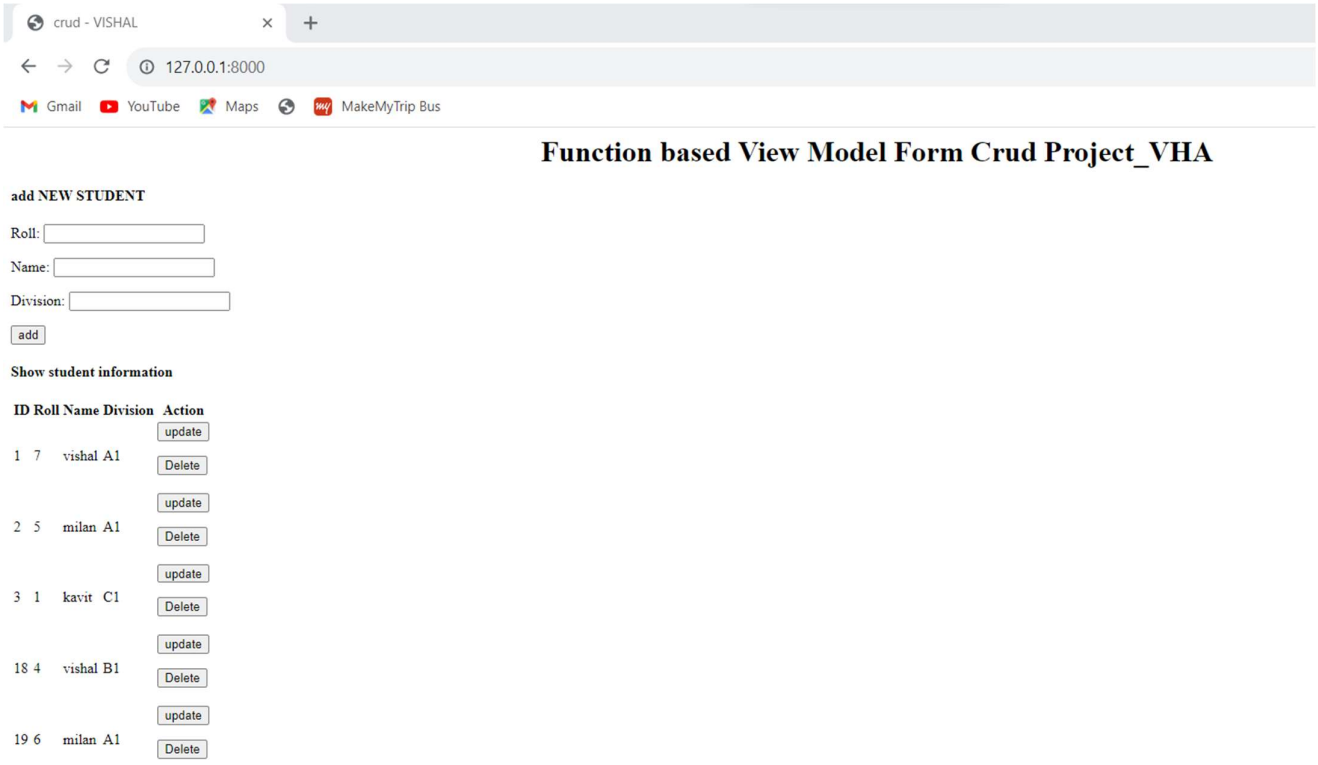
```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views.add_show,name="addshow"),
    path('update/<int:id>/',views.update_data,name='updatedata'),
    path('delete/<int:id>/',views.delete_data,name='deletedata')

]
```

For change in htmlfile

```
 <form action="{% url 'updatedata' st.id %}" method="post">
                        {% csrf_token %}
                        <input type="submit" value="update">
```

For updatestudent html file

```
{% extends 'database/base.html' %}
{% block content %}

 <div>
        <h4>update STUDENT information</h4>
        <form method="post">
                {% csrf_token %}
                {{form.as_p}}
                <input type="submit" value="update">

        </form>
        <a href="{% url 'addshow' %}"> back to home </a>
 </div>
{% endblock content %}
```


crud - VISHAL
127.0.0.1:8000
Gmail
YouTube
Maps
MakeMyTrip Bus
Function based View Model Form Crud Project_VHA
add NEW STUDENT
Roll:
Name:
Division:
add
Show student information
ID Roll Name Division Action
1 7 vishal A1
update
Delete
2 5 milan A1
update
Delete
3 1 kavit C1
update
Delete
18 4 vishal B1
update
Delete
19 6 milan A1
update
Delete