

Django Crud with Modelform :-

models.py file code:-

```
from django.db import models

# Create your models here.

class ContactForm(models.Model):

    fullname= models.CharField(max_length=100)
    email= models.EmailField()
    contact= models.CharField(max_length=50)
    message= models.CharField(max_length=200)
```

forms.py file code:-

```
from django import forms
from .models import ContactForm
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User

class FormContactForm(forms.ModelForm):
    class Meta:
        model= ContactForm
        fields= ["fullname", "email", "contact", "message"]

class NewUserForm(UserCreationForm):
    email = forms.EmailField(required=True)

    class Meta:
        model = User
        fields = ["username", "email", "password1", "password2"]
```

urls.py file code:-

```
from django.urls import path

from . import views
```

```
urlpatterns=[
path('',views.index,name='index'),
path('home/',views.home,name='home'),
path('homepage/',views.homepage,name='homepage'),
path('about/',views.about,name='about'),

path('showform/', views.showform),
path('search/', views.search,name='search'),
path('search1/', views.search1,name='search1'),

path('editemp/<int:id>', views.editemp, name='editemp'),
path('updateemp/<str:id>', views.updateemp, name='updateemp'),

path("register", views.register_request, name="register"),
path("login", views.login_request, name="login"),
path("logout", views.logout_request, name= "logout"),
```

views.py file code:-

```
from urllib import request
from django.shortcuts import render,redirect
from hello.models import ContactForm,contactEnquiry
from hello.forms import FormContactForm
from .forms import NewUserForm

from django.http import HttpResponseRedirect
from django.contrib.auth import login, authenticate,logout #add this
from django.contrib import messages
from django.contrib.auth.forms import AuthenticationForm
from django.db.models import Q

def showform(request):
    form= FormContactForm(request.POST or None)
    if form.is_valid():
        form.save()

    context= {'form': form }
```

```

return render(request, 'contactform.html', context)

def home(request):
    serviceData=ContactForm.objects.all()
    data={'serviceData':serviceData
        }

    return render(request,"home.html",data)

def editemp(request,id):
    data=ContactForm.objects.get(id=id)
    return render(request, "edit.html",{ 'editupdaterecord':data})

def updateemp(request,id):
    if request.method=='POST':
        pi=ContactForm.objects.get(id=id)
        form= FormContactForm(request.POST ,instance=pi)
        if form.is_valid():

            form.save()
            msg="record updated"
            return render(request, "update.html",{ 'editupdaterecord':msg})

def search(request):
    if request.GET.get('myform'): # write your form name here
        book_name = request.GET.get('myform')

        try:

            status = ContactForm.objects.filter(fullname__icontains=book_name)
            return render(request,"search1.html",{ "books":status})
        except:
            return render(request,"search1.html",{ 'books':status})
    else:
        return render(request, 'search1.html', { 'books':status})

def search1(request):
    if 'q' in request.GET:

```

```

q=request.GET['q']
multiple_q=Q(Q(fullname=q)| Q(message=q))
data=ContactForm.objects.filter(multiple_q)

else:
    data =ContactForm.objects.all()

return render(request,'search1.html',{
'data':data
})

def login_request(request):
    if request.method == "POST":
        form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            username = form.cleaned_data.get('username')
            password = form.cleaned_data.get('password')
            user = authenticate(username=username, password=password)
            if user is not None:
                login(request, user)
                messages.info(request, f"You are now logged in as {username}.")
                return redirect("homepage")
            else:
                messages.error(request,"Invalid username or password.")
        else:
            messages.error(request,"Invalid username or password.")
    form = AuthenticationForm()
    return render(request, 'login.html', context={"login_form":form})

def register_request(request):
    if request.method == "POST":
        form = NewUserForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)
            messages.success(request, "Registration successful." )
            return redirect("home")
        messages.error(request, "Unsuccessful registration. Invalid
information.")
    form = NewUserForm()
    return render (request, 'register.html', context={"register_form":form})

def logout_request(request):

```

```
logout(request)
messages.info(request, "You have successfully logged out.")
return redirect("homepage")
```

under your templates folder /

home.html file code:-

```
<html>
  <head>
    {% load static %}

    <link rel="stylesheet" href="{% static 'ok.css' %}">

  </head>
  <body>

    <h1 class="myclass">hello world how are you
      {{data}}
    </h1>
    <div>
      <table>
        <tr><th> id </th> <th> full name </th><th> email </th><th>Contact
</th><th>Edit</th></tr>
```

```

        {% for n in serviceData %}
        <tr>
        <td> {{n.id}}</td>

        <td><span class="{{n.fullname}}"> {{n.fullname}}</span></td>

        <td><span class="{{n.email}}">{{n.email}} </span></td>

        <td> {{n.contact}} </td>

        <td><a href="{% url 'editemp' n.id %}">Edit</a></td>
</tr>

        {% endfor %}
    </table>
</div>
</body>
</html>

```

Edit.html file code:-

```

<form action="{% url 'updateemp' editupdaterecord.id %}" method="post">

    {% csrf_token %}

    name <input type="text" name="fullname" value={{editupdaterecord.fullname}}>

    email <input type="text" name="email" value={{editupdaterecord.email}}>
    Contact <input type="text" name="contact" value={{editupdaterecord.contact}}>
    message <input type="text" name="message" value={{editupdaterecord.message}}>

    <input type="submit" value="update">

</form>

```

update.html file code:-

```
{{editupdaterecord}}
```

search1.html file code:-

```
<form name="myform" method="GET" action="{% url 'search1' %}">
  <input type="text" name="q" placeholder="Search" />
<button type=submit> search</button></form>

{% for data in data %}
<td>{{data.fullname}}</td>
<td>{{data.email}}</td>
<td>{{data.contact}}</td>
{% endfor %}
```

register.html file code:-

```
<div class="container py-5">
  <h1>Register</h1>
  <form method="POST">
    {% csrf_token %}
    {{ register_form }}
    <button class="btn btn-primary" type="submit">Register</button>
  </form>
  <p class="text-center">If you already have an account, <a href="{% url
'login' %}">login</a> instead.</p>
</div>
```

login.html file code:-

```
<!--Login-->
<div class="container py-5">
  <h1>Login</h1>
  <form method="POST">
    {% csrf_token %}
    {{ login_form }}
    <button class="btn btn-primary" type="submit">Login</button>
  </form>
  <p class="text-center">Don't have an account? <a href="{% url 'register'
%}">Create an account</a>.</p>
</div>
```

homepage.html file code:-

```
<!--Navbar-->
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarText" aria-controls="navbarText" aria-expanded="False" aria-
label="Toggle navigation">
  <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarText">
  <ul class="navbar-nav mr-auto">
    {% if user.is_authenticated %}

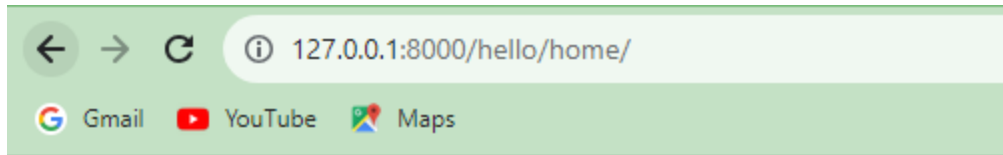
    <li class="nav-item">
      <a class="nav-link" href="{% url 'logout' %}">Logout</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Welcome, {{user.username}}</a>
    </li>

    {% else %}

    <li class="nav-item">
      <a class="nav-link" href="{% url 'login' %}">Login</a>
    </li>

    {% endif %}
  </ul>
</div>
</nav>
```

To make edit record :-



hello world how are you




| id | full name | email | Contact | Edit |
|----|--------------|---------------------------|-------------|----------------------|
| 1 | okji | okom@gmail.com | 8149996597 | Edit |
| 2 | yyyyyy | yy@gmalil.com | 28899 | Edit |
| 3 | Lokesh Singh | lokeshsingh9326@gmail.com | 8149996597 | Edit |
| 4 | apple111111 | apple@gmail.com | 8149996597 | Edit |
| 5 | omji | omji@gmail.com | 8999788999 | Edit |
| 6 | ankit sir | ankisir@gmail.com | 81499965987 | Edit |

And add new record :-

| | |
|---------------------------------------|----------------------|
| id | <input type="text"/> |
| Full Name | <input type="text"/> |
| Email | <input type="text"/> |
| Contact | <input type="text"/> |
| Message | <input type="text"/> |
| <input type="submit" value="Submit"/> | |

To check login and register :-

← → ↻ 127.0.0.1:8000/hello/register

 Gmail  YouTube  Maps

Register

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. Email: Password:




- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

If you already have an account, [login](#) instead.

← → ↻ 127.0.0.1:8000/hello/login

 Gmail  YouTube  Maps

Login

Username: Password:

Don't have an account? [Create an account](#).