

Step 1:-

models.py file:-

```
from django.db import models

from django.contrib.auth.models import User

# Create your models here.

class Task(models.Model):

    title = models.CharField(max_length=100, null=True)

    content = models.CharField(max_length=1000, null=True, blank=True)

    date_posted = models.DateTimeField(auto_now_add=True, null=True)

    user = models.ForeignKey(User, max_length=10, on_delete=models.CASCADE, null=True)
```

Step 2:-

forms.py file code :-

```

> todo > forms.py > ...

from django.contrib.auth.models import User

from django.contrib.auth.forms import AuthenticationForm
from django.forms.widgets import PasswordInput, TextInput

from django import forms

# - Register a user

class CreateUserForm(UserCreationForm):

    class Meta:

        model = User
        fields = ['username', 'email', 'password1', 'password2']

# - Login a user

class LoginForm(AuthenticationForm):

    username = forms.CharField(widget=TextInput())
    password = forms.CharField(widget=PasswordInput())

```

```

# - Create a task

class CreateTask(forms.ModelForm):

    class Meta:

        model = Task
        fields = ['title', 'content',]
        exclude = ['user',]

```

Views.py file Code :-

> todo > views.py > ...

```
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect

from .forms import CreateUserForm, LoginForm, CreateTaskForm

from django.contrib.auth.models import auth
from django.contrib.auth import authenticate, login

from django.contrib.auth.decorators import login_required
```

- Registering / Creating a user

```
def register(request):

    form = CreateUserForm()

    if request.method == 'POST':

        form = CreateUserForm(request.POST)

        if form.is_valid():

            form.save()

            return redirect("/dashboard/")

    context = {'form':form}

    return render(request, 'register.html', context=context)
```

```
# - Login a user

def my_login(request):

    form = LoginForm

    if request.method == 'POST':

        form = LoginForm(request, data=request.POST)

        if form.is_valid():

            username = request.POST.get('username')
            password = request.POST.get('password')

            user = authenticate(request, username=username, password=password)

            if user is not None:

                auth.login(request, user)

                return redirect("dashboard")

    context = {'form':form}

    return render(request, 'my-login.html', context=context)
```

```
# - Create a task page

@login_required(login_url='my-login')
def createTask(request):

    form = CreateTaskForm()

    if request.method == 'POST':

        form = CreateTaskForm(request.POST)

        if form.is_valid():

            form.save()

            return redirect('')

    context = {'form':form}

    return render(request, 'profile/create-task.html', context=context)
```

```
| I
```

```
# - View all tasks page
```

```
@Login_required(login_url='my-login')
```

```
def viewTask(request):
```

```
    current_user = request.user.id
```

```
    task = Task.models.all().filter(user=current_user)
```

```
    context = {'task': task}
```

```
    return render(request, 'profile/view-tasks.html', context=context)
```

```
# - Logout a user
```

```
# - Update task page
```

```
@Login_required(login_url='my-login')
```

```
def updateTask(request, pk):
```

```
    task = Task.objects.get(id=pk)
```

```
    form = CreateTaskForm(instance=task)
```

```
    if request.method == 'POST':
```

```
        form = CreateTaskForm(request.POST, instance=task)
```

```
        if form.is_valid():
```

```
            form.save()
```

```
            return redirect('view-tasks')
```

```
I
```

```
# - Delete task page I
def deleteTask(request, pk):
    task = Task.objects.get(id=pk)

    if request.method == 'POST':
        task.delete()

        return redirect('view-tasks')

    return render(request, 'profile/delete-task.html')
```

urls.py file code :-

```
taskly > todo > urls.py > ...
39 # ----- CREATE TASK ----- #
40 # ----- #
41
42 path('create-task', views.createTask, name="create-task"),
43
44
45 # ----- READ TASK ----- #
46 # ----- #
47
48 path('view-tasks', views.viewTask, name="view-tasks"),
49
50 I
51 # ----- UPDATE TASK ----- #
52 # ----- #
53
54
55 path('update-task/<str:pk>/', views.updateTask, name="update-task"),
56
57
```

```
# ----- DELETE TASK ----- #  
# ----- #  
  
path('delete-task/<str:pk>/', views.delete_task, name="update-task"),
```

View-tasks.html file code:-



```
view-tasks.html X dashboard.html  
templates > profile > view-tasks.html > html > div.container.bg-white.shadow-md.p-5.form-layout  
  
<br>  
<div class="container bg-white shadow-md p-5 form-layout">  
  <h3> My tasks </h3>  
  <hr>  
  {% for tsk in task %}  
    {{ tsk.title }}  
    <br> <br>  
    {{ tsk.content }}  
  {% endfor %}
```

create-task.html file code :-

```
create-task.html X
taskly > templates > profile > create-task.html > html > div.container.bg-white.shadow-md.p-5.form-layout > form
27
28     <hr>
29
30     <br>
31
32     <form method="POST" autocomplete="off">
33
34         {% csrf_token %}
35
36
37         {{form.title|as_crispy_field}}
38
39         <br> <br>
40
41         {{form.content|as_crispy_field}}
42
43
44         <input class="btn btn-primary navbar-btn" type="submit" value="Submit"/>
45
46     </form>
47
48 </div>
49
50
```

delete-task.html file code :-

```
<form method="POST" autocomplete="off">
    {% csrf_token %}

    <p> Are your sure that you want to delete this task? </p>

    <br> <br>

    <a href="{% url 'view-tasks' %}"> Go back </a>
    |
    |
    <input class="btn btn-primary navbar-btn" type="submit" value="Delete"/>

</form>
```