Models

Go to Models.py and create a class as Below

**from django.db import models  
class RegisterdUser(models.Model):  
 name = models.CharField(max\_length=100)  
 emailid = models.CharField(max\_length=100)  
 phonenumber = models.CharField(blank=True,null=True,max\_length=13)  
 password = models.CharField(max\_length=30)**

Register this Class in admin.py

**from django.contrib import admin  
from .models import RegisterdUser,CreateTestsuite  
  
admin.site.register(RegisterdUser)**

Go to forms.py create a class with that model

**from django.forms import ModelForm  
from django import forms  
from .models import RegisterdUser,CreateTestsuite  
  
class RegisterForm(ModelForm):  
 password = forms.CharField(widget=forms.PasswordInput)  
  
 class Meta:  
 model = RegisterdUser  
 fields = '\_\_all\_\_'  
 labels = {  
 "name":"Username",  
 "emailid":"Email Id",  
 "phonenumber":"Phone Number",  
 "password":"password"  
 }**

in order to View the html we should use as forms.as\_p

**{% extends "base.html" %}  
  
<body>  
 {% block content %}  
 <form action="" method="post">  
 {% csrf\_token %}  
 {{form.as\_p}}  
 <input type="submit" value="Submit">  
 </form>  
 {% endblock %}  
</body>**

Once Registered user class in Go to Views.py and add the redirect because once sigin is done it should redirect

Import forms in Views.py

from .forms import RegisterdUser

from django.contrib import messages

We need to create a function called Register it should be accessed via post method

It will have two method one for creation one for Getting the data

**python manage.py sqlmigrate IIS\_APP 0001** to Create table in admin

Argument to the redirect is an URL which is in strings

**def registartion(request):**

**if request.method == "POST":**

**form = RegisterForm(request.POST)**

**if form.is\_valid():**

**form.save()**

**messages.success(request,"Account created successful")**

**return redirect("signin")**

**else:**

**form = RegisterForm()**

**user\_info = {'form':form}**

**return render(request,"register.html",user\_info)**

we can get data from registered users using the below commands

python manage.py shell

**>>from groceries\_app.models import RegisteredUser**

**>>RegisteredUser.objects.all()**

we can single data using get command

**RegisteredUser.objects.get(id=1)**

we can get inside data also

**RegisteredUser.objects.get(id=1).id**

In order to get the latest created object we use

**>>> RegisteredUser.objects.last()**

we use filter in oreder to filter out the data

**>>> RegisteredUser.objects.filter(id=2).first()**

we can create users through shell

**>>> post\_data = RegisteredUser(name="alex",emailid="alex@gmail.com",phoneNum="9229292",password="alex")**

**>>> post\_data.save() --> we can create user**

we can interact with the created users by using Views functuion we have to create class with inheratince of List View we need to write three steps

1. which model we need to import

2.Template name that is HTML

**ListView**

3.how much data need to be accessed

views.py

**from django.views.generic import ListView**

**class UserListView(ListView):  
 model = RegisteredUser  
 template\_name = "user\_data.html"  
 context\_object\_name = "alldata"**

the template should like this

**{% extends "base.html" %}  
<body>  
{% block content %}  
{% for user in alldata %}  
 Id:{{user.id}}<br>  
 Username:{{user.name}}<br>  
 Email ID:{{user.emailid}}<br><br>  
 {% endfor %}  
{% endblock %}  
</body>**

we need to add path min Urls.py

urls.py

**path('userList',views.UserListView.as\_view(), name="userList")**

**>>Createjob.objects.all()[0].jobname**

**DetailView:**

List view is used to diplay data of all instances but Detail View is of indivual instances

Views.py

**class UserDetailView(DetailView):  
 model = RegisteredUser**

**HTML**

**{% extends "base.html" %}  
<body>  
{% block content %}  
Id : {{object.id}}<br>  
UserName: {{object.name}}<br>  
Email ID: {{object.emailid}}<br>  
Phone Number: {{object.phoneNum}}<br><br>  
{% endblock %}  
</body>**

Add the path asa below here pk represents Primary key

**path('userdetail/<int:pk>',views.UserDetailView.as\_view(template\_name="user\_details.html"),name='userdetail')**

**In oreder to Access we should use the uRL in Chrome Tab we will be getting the first ID Data**

[**http://127.0.0.1:8000/userdetail/1**](http://127.0.0.1:8000/userdetail/1)

**Create View:**

**class UserCreateView(CreateView):  
 model = RegisteredUser  
 form\_class = RegisterForm**

IN HTML

**{% extends "base.html" %}  
<body>  
 {% block content %}  
 <form method="post">  
 {% csrf\_token %}  
 {{form.as\_p}}  
 <input type="submit" value="Submit">  
 </form>  
 {% endblock %}  
</body>**

**URL**

**path('usercreate/', views.UserCreateView.as\_view(template\_name="user\_create.html"), name='usercreate')**

we can redirect to another userdetails by adding get\_absloutefunction in models

models.py

**from django.db import models  
from django.urls import reverse  
  
class RegisteredUser(models.Model):  
 name = models.CharField(max\_length=100)  
 emailid = models.CharField(max\_length=100)  
 phoneNum = models.CharField(blank=True, null=True, max\_length = 20)  
 password = models.CharField(max\_length=30)  
 def get\_absolute\_url(self):  
 return reverse('userdetail',kwargs={'pk':self.pk})**

UpdateView

**Add the Module**

**class UserUpdateView(UpdateView):  
 model = RegisteredUser  
 form\_class = RegisterForm**

**ADD**

**path('userupdate/<int:pk>',views.UserUpdateView.as\_view(template\_name="user\_create.html"),name='userupdate')**

**redirect to URL:http://127.0.0.1:8000/userupdate/3**

**Authorization**

in order to not give direct access to the users we use

**Views.py**

**from django.contrib.auth.mixins import UserPassesTestMixin**

**class UserUpdateView(UserPassesTestMixin,UpdateView):  
 model = RegisteredUser  
 form\_class = RegisterForm  
 def test\_func(self):  
 if self.request.user.is\_active:  
 return True  
 else:  
 return False**

**user\_create.html**

**{% extends "base.html" %}  
  
<body>  
 {% block content %}  
 <form method="post">  
 {% csrf\_token %}  
 {% if request.user.is\_superuser %}  
 {{form.as\_p}}  
 <button type="submit">Submit data</button>  
 {% else %}  
 the logged user is not a super user  
 {% endif %}  
 </form>  
 {% endblock %}  
</body>**