Django Models:

-Django framework has builtin **ORM mechanism** (Object Relational Mapping)

-According to ORM django framework transfers objects between django application and a database

-Developers are not required to type sql queries in a django project

- A developer has to create a model class in a django project, for each table required in the database

- Django framework creates a table for a model class with syntax

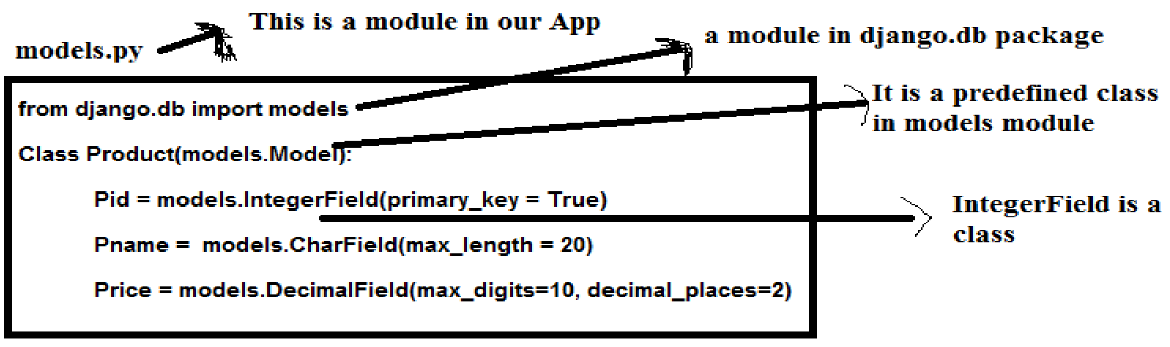
- ***Appname_Modelname*** and also creates columns for each variable created in a model class

-Model classes of an application must be defined in **models.py** file

Imp points to remember:

1. We should create model class
2. By seeing your model class jango framework automatically creates the database table as per model how u defined
3. To do step1 and step2 first we need to install a db in our machine
4. Jango framework automatically maps model class with database table
5. As a developer we should only create model class rest all work will be taken care by jango framework
6. Whatever variables we create in model class those will become fields in table

<u>**models.py**</u>

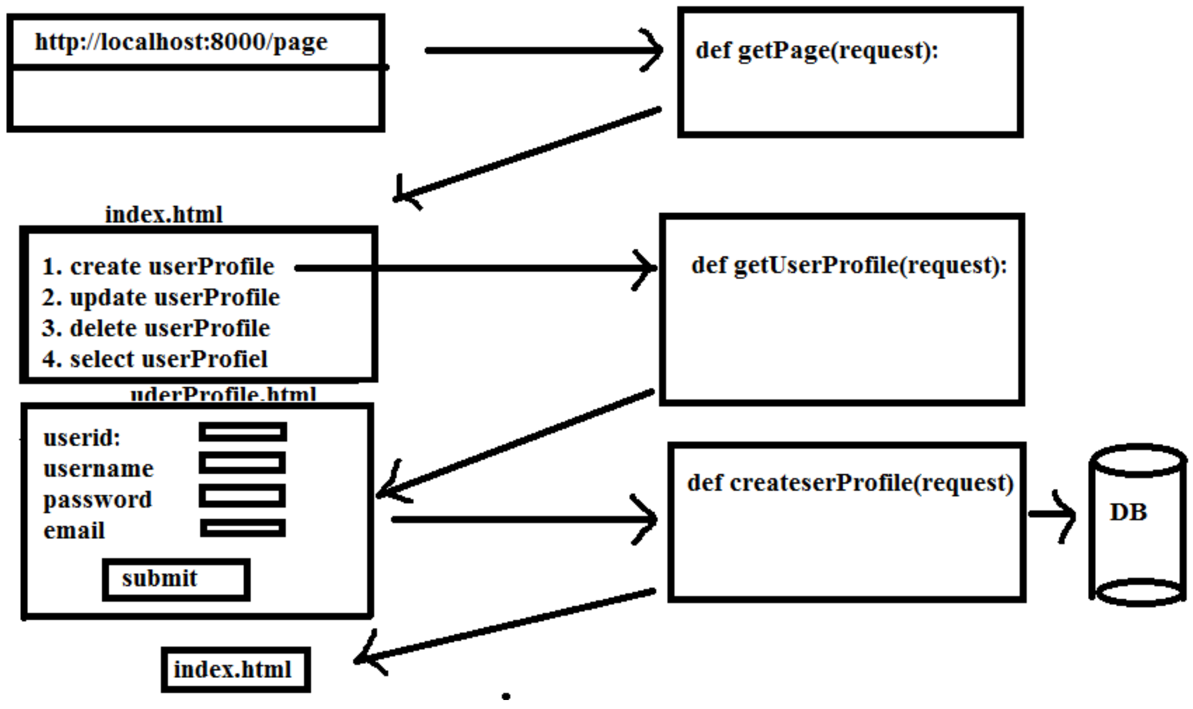

models.py
This is a module in our App
a module in django.db package
It is a predefined class in models module
IntegerField is a class

```
from django.db import models
Class Product(models.Model):
        Pid = models.IntegerField(primary_key = True)
        Pname =  models.CharField(max_length = 20)
        Price = models.DecimalField(max_digits=10, decimal_places=2)
```

Django framework creates following table for the above model class
ProductApp_Product   (table name)

User Profile to perform CRUD operations:

Flow of creating a user profile



Step1: Create a django project with name ***UserProfileProject***

Step2: Create an app with name ***ProfileApp***

Step3: Open ***settings.py*** and configure database properties

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.oracle',
        'NAME': 'orcl',
        'HOST':'localhost',
        'PORT':'1521',
        'USER':'pythondev1',
        'PASSWORD':'pythondev12345',
    }
}
```

Also add ***ProfileApp*** in **INSTALLED_APPS** list

&

Add template directory in settings.py

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTem
        'DIRS': ['C:/mydjango/UserProfileProject/templates'],
```

Step 4: Open ***models.py*** and create ***UserProfile*** class

```
from django.db import models

# Create your models here.
class UserProfile(models.Model):
    userid   = models.IntegerField(primary_key=True)
    username = models.CharField(max_length=20)
    password = models.CharField(max_length=20)
    email    = models.EmailField()
```

Step 5: In cmd> ***run manage.py*** and execute the following 2 commands in the below window

(i) Makemigrations ProfileApp

(ii) migrate ProfileApp

Step 6: Verify table is created in database or not

You should see a table got created with name ***ProfileApp_UserProfile***

Note: Table gets generated with Appname_Modelclassname

Step 7: Open ***views.py*** and define the following view function.

```
from django.shortcuts import render

# This view function returns index.html to the browser
def getPage(request):
    return render(request, 'index.html')
```

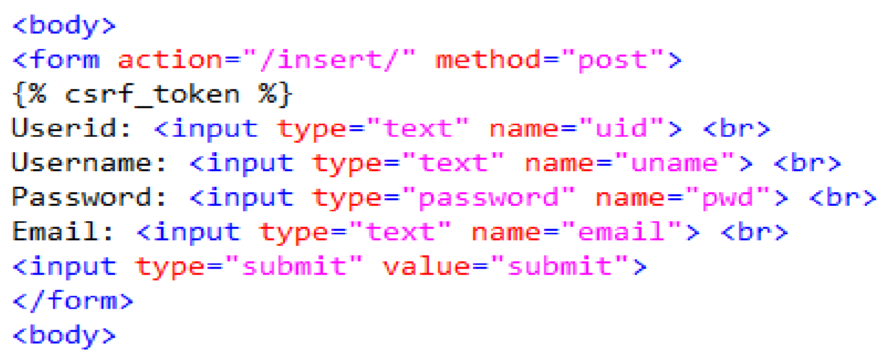
Step 8: Create***index.html*** under **templates** directory with the following code which generates 4 hyperlink

```
<html>
<body>
<a href="/getProfilePage/"> create a new UserProfile </a> <br>
<a href="/getUpdatePage/"> update a UserProfile </a> <br>
<a href="/getDeletePage/">delete a UserProfile</a><br>
<a href="/getUserProfiles/">Display UserProfiles</a><br>
</body>
<html>
```

Step 9: Open **views.py** and define the following view function (2nd)

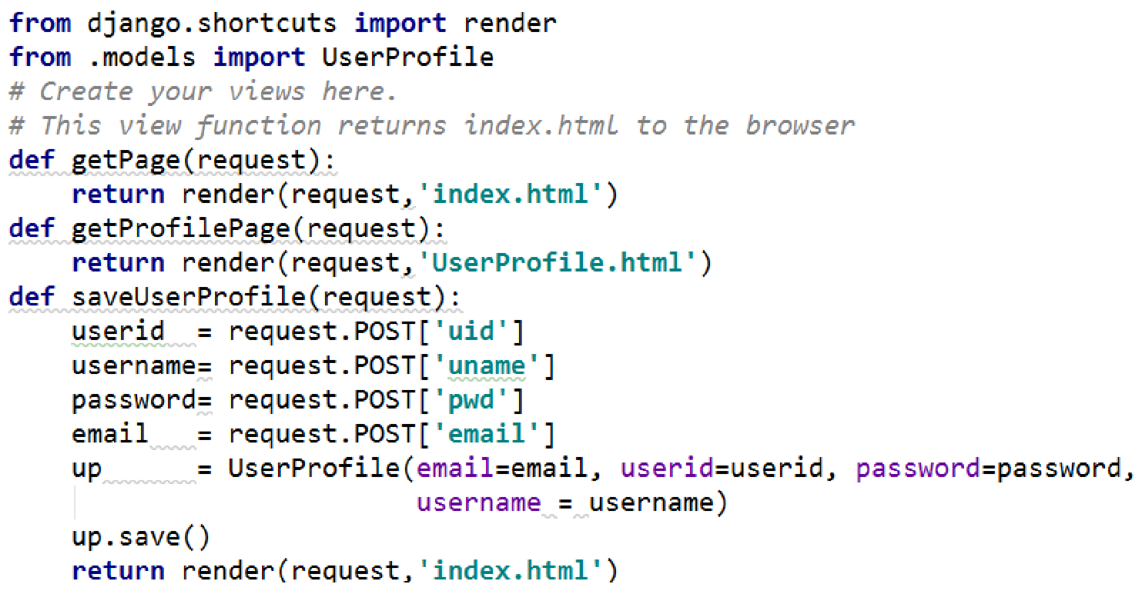
```
from django.shortcuts import render

# This view function returns index.html to the browser
def getPage(request):
    return render(request, 'index.html')

#This view function returns UserProfile.html to browser
def getProfilePage(request):
    return render(request,'UserProfile.html')
```

Step 10: Create ***UserProfile.html*** under ***templates*** directory



```
<body>
<form action="/insert/" method="post">
{% csrf_token %}
Userid: <input type="text" name="uid"> <br>
Username: <input type="text" name="uname"> <br>
Password: <input type="password" name="pwd"> <br>
Email: <input type="text" name="email"> <br>
<input type="submit" value="submit">
</form>
<body>
```

Step 11: Open ***views.py*** and define the following new function (3rdfunc)

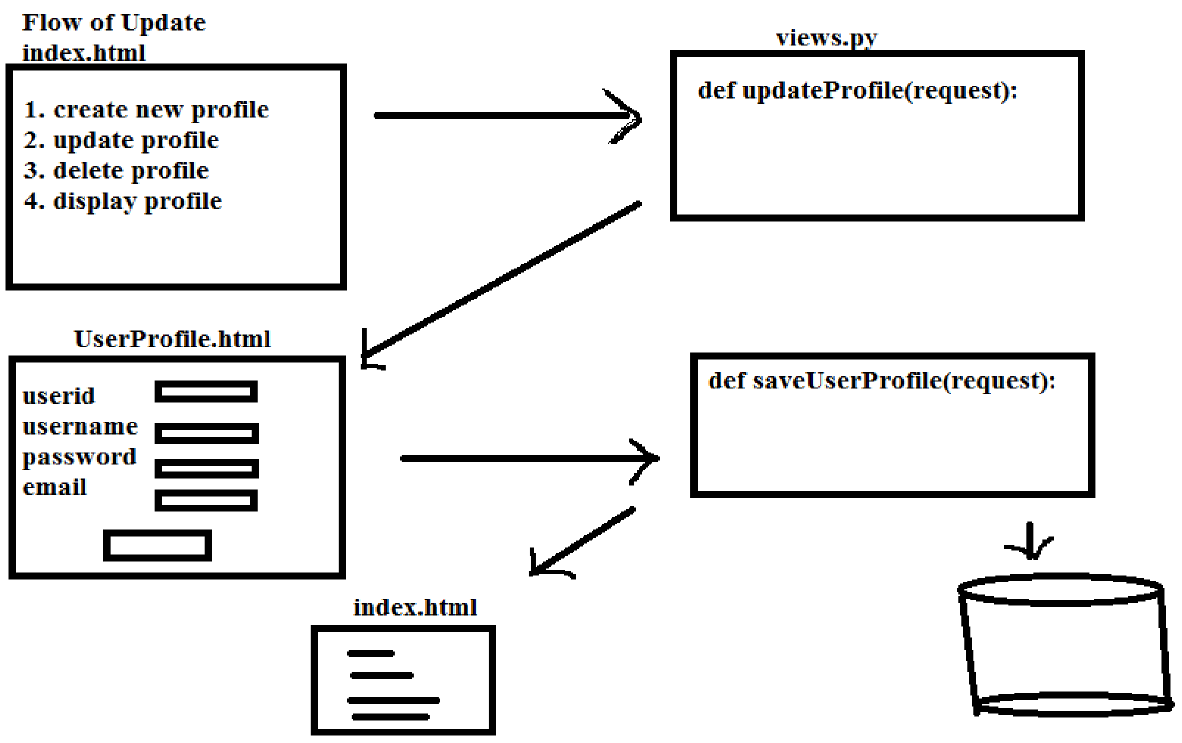


```
from django.shortcuts import render
from .models import UserProfile
# Create your views here.
# This view function returns index.html to the browser
def getPage(request):
    return render(request,'index.html')
def getProfilePage(request):
    return render(request,'UserProfile.html')
def saveUserProfile(request):
    userid  = request.POST['uid']
    username= request.POST['uname']
    password= request.POST['pwd']
    email   = request.POST['email']
    up      = UserProfile(email=email, userid=userid, password=password,
                          username = username)
    up.save()
    return render(request,'index.html')
```

Step 12: Open ***urls.py***and add the following code

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

urlpatterns = [
    path('admin/',admin.site.urls),
    path('page/',views.getPage),
    path('getProfilePage/',views.getProfilePage),
    path('insert/',views.saveUserProfile)
```

Flow of UpdateProfile



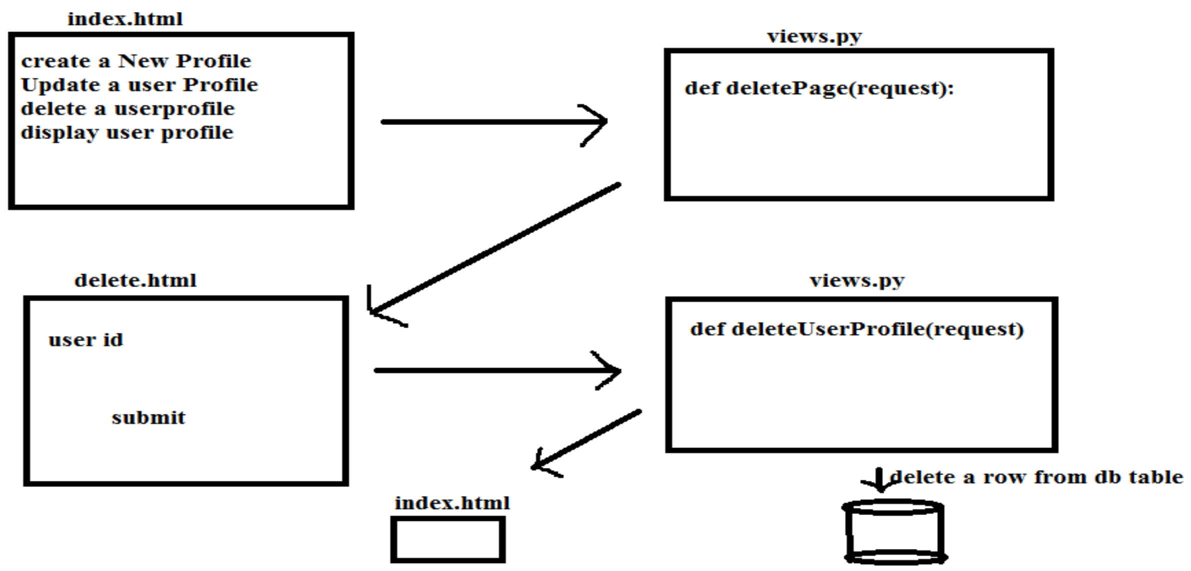
Step 12: Open ***views.py*** and add the following function (4th)

```
#This view func returns userProfile.html to update a userprofile
def updateProfile(request):
    return render(request,'UserProfile.html')
```

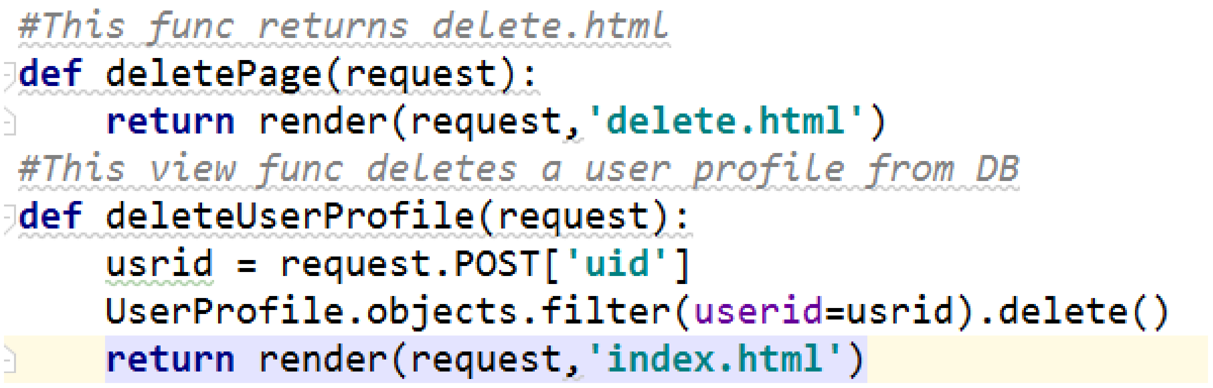
Step 13: Open***urls.py*** and append the following path

```
path('getUpdatePage/',views.updateProfile)
```

Flow of Delete Profile:

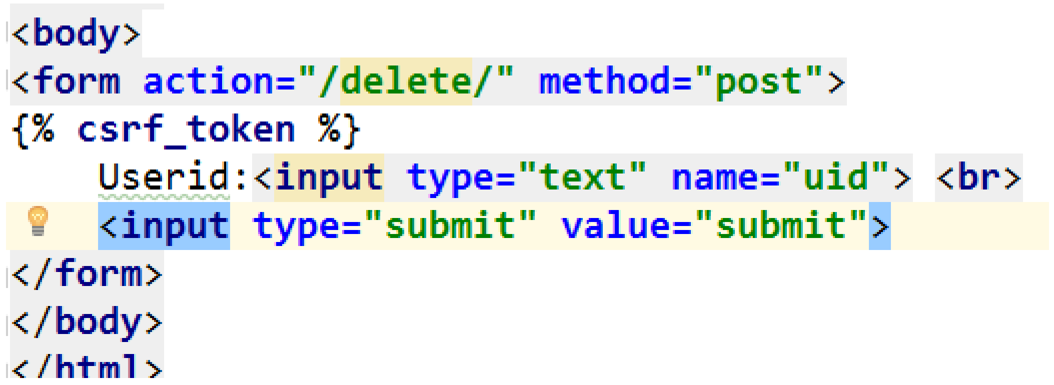


Step 14: Open ***views.py*** and define the following two functions (5th and 6th)



```
#This func returns delete.html
def deletePage(request):
    return render(request,'delete.html')
#This view func deletes a user profile from DB
def deleteUserProfile(request):
    usrid = request.POST['uid']
    UserProfile.objects.filter(userid=usrid).delete()
    return render(request,'index.html')
```

Step 15: Create a ***delete.html*** under ***templates*** directory like the following

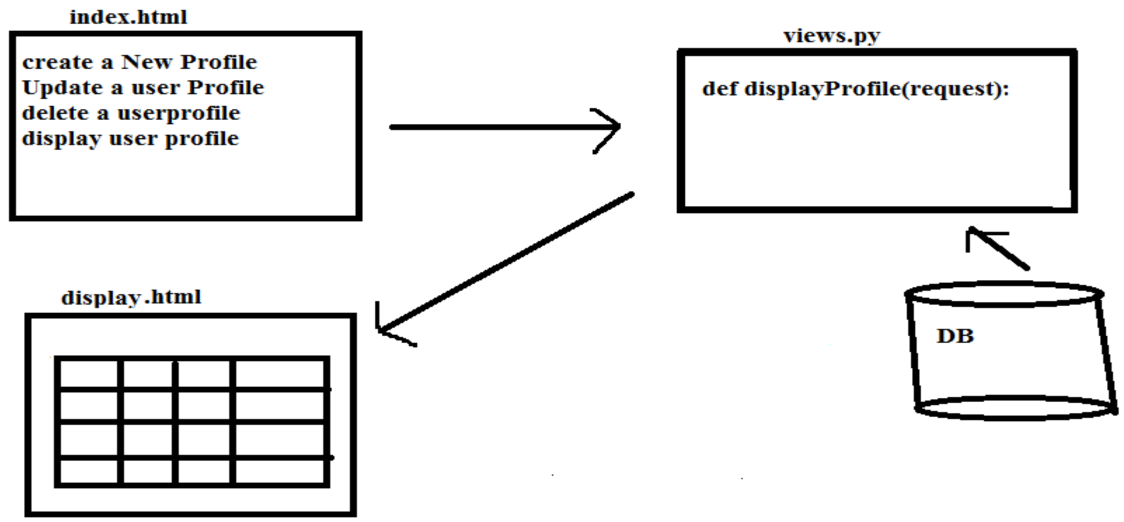


```
<body>
<form action="/delete/" method="post">
{% csrf_token %}
    Userid:<input type="text" name="uid"> <br>
    <input type="submit" value="submit">
</form>
</body>
</html>
```

Step 16: Open **urls.py** and append the following 2 path statements.

```
path('getDeletePage/',views.deletePage),
path('delete/',views.deleteUserProfile),
```

Flow of Display:



Step 17: Open ***views.py*** and add the following view function (7th)

```
#This view function display all userprofiles
def displayUserProfiles(request):
    UserProfiles = UserProfile.objects.all()
    return render(request,'display.html',{'usersprofiles':UserProfiles})
```

Step 18: Create ***display.html*** under ***templates*** directory and define the following code

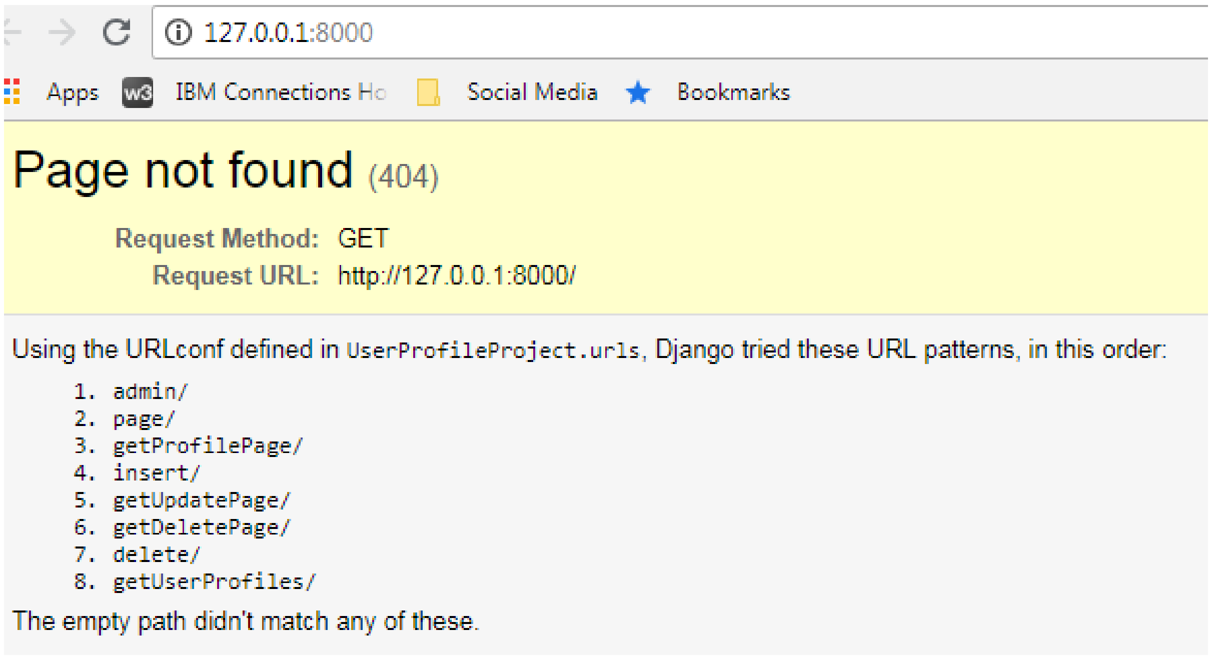
```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<table border="1">
    <tr>
        <th>userid</th>
        <th>username</th>
        <th>password</th>
        <th>email1</th>
    </tr>
    {% for up in usersprofiles %}
    <tr>
        <td> {{ up.userid }}</td>
        <td> {{ up.username }}</td>
        <td> {{ up.password }}</td>
        <td> {{ up.email }}</td>
    </tr>
    {% endfor %}
</table>
</body>
</html>
```

Step 19: In **urls.py** add below code as last line

```
path('getUserProfiles/',views.displayUserProfiles)
```

Step 20: Execute project by clicking **on Run->Run UserProfileProject**

Step 21: Check the output in browser



Page not found (404)

Request Method: GET
Request URL: http://127.0.0.1:8000/

Using the URLconf defined in `UserProfileProject.urls`, Django tried these URL patterns, in this order:

```
1. admin/
2. page/
3. getProfilePage/
4. insert/
5. getUpdatePage/
6. getDeletePage/
7. delete/
8. getUserProfiles/
```

The empty path didn't match any of these.