

2CSDE86 Application Development Frameworks

Lab-4 Task

Submitted by: Labdhi Sheth 18BCE101

Aim:

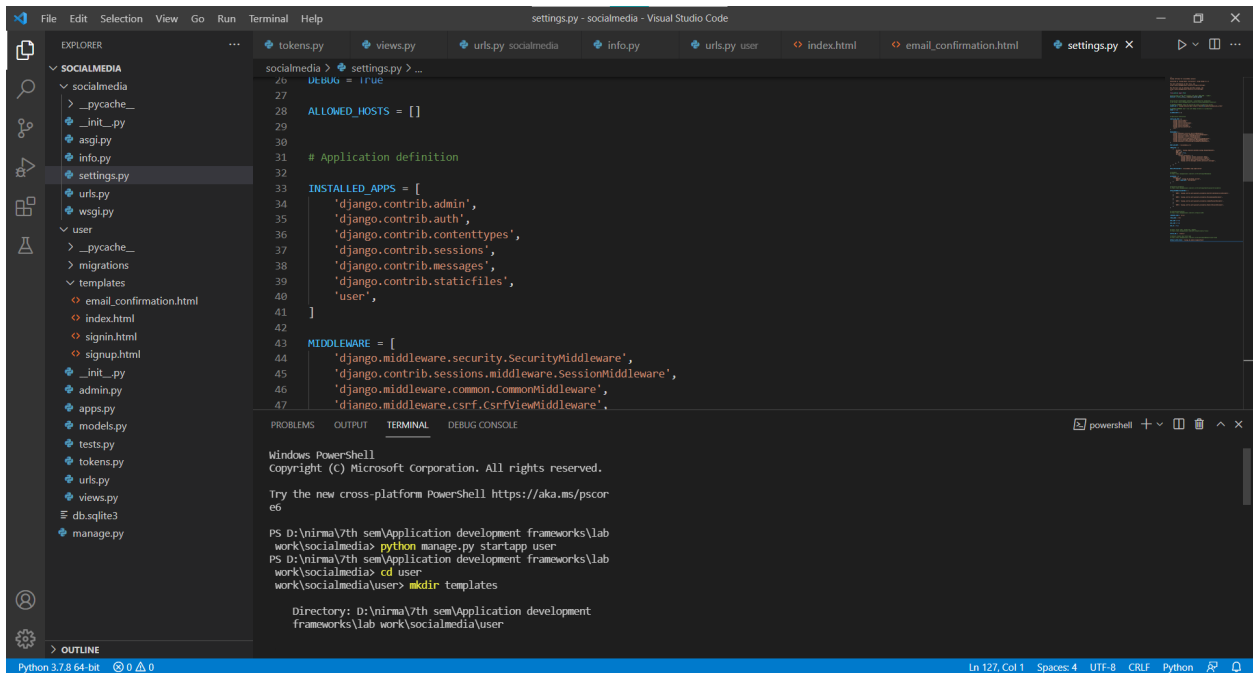
Perform CRUD operation and connect database with your social media website. Store all the data and retrieve the data according to the user Assignment

Methodology:

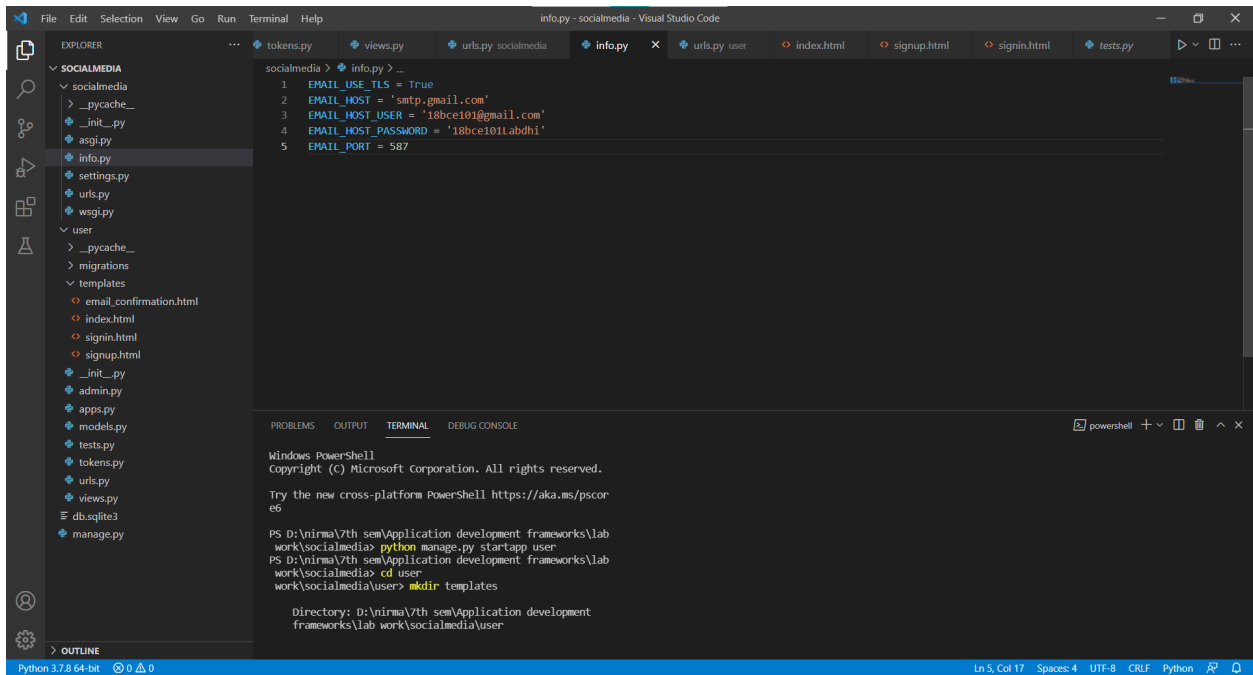
1. Open the anaconda terminal and write the command
`django-admin.py startproject socialmedia`
2. Open the project in vscode and open new terminal.
In the terminal write the commands to start a new app:
`python manage.py startapp user`

After making the apps make the templates folder as:
`cd user`
`mkdir templates`

3. `settings.py`
Register the app under the installed apps as:



4. info.py



Add the email configurations to the settings.py file:

```
12
13 from pathlib import Path
14 from .info import *
15 # Build paths inside the project like this: BASE_DIR / 'subdir'.
16 BASE_DIR = Path(__file__).resolve().parent.parent
17
18 EMAIL_USE_TLS = EMAIL_USE_TLS
19 EMAIL_HOST = EMAIL_HOST
20 EMAIL_HOST_USER = EMAIL_HOST_USER
21 EMAIL_HOST_PASSWORD = EMAIL_HOST_PASSWORD
22 EMAIL_PORT = EMAIL_PORT
23
24 # Quick-start development settings - unsuitable for production
25 # See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/
26
27 # SECURITY WARNING: keep the secret key used in production secret!
28 SECRET_KEY = 'django-insecure-$u7-*ltgr=n-^m4y7u#%2azsay5kif)&c#ewvxxhi_xf(8u7'
29
30 # SECURITY WARNING: don't run with debug turned on in production!
31 DEBUG = True
32
```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\nirma\7th sem\Application development frameworks\lab work\socialmedia> python manage.py startapp user
PS D:\nirma\7th sem\Application development frameworks\lab work\socialmedia> cd user
work\socialmedia\user> mkdir templates
Directory: D:\nirma\7th sem\Application development frameworks\lab work\socialmedia\user

5. User app:

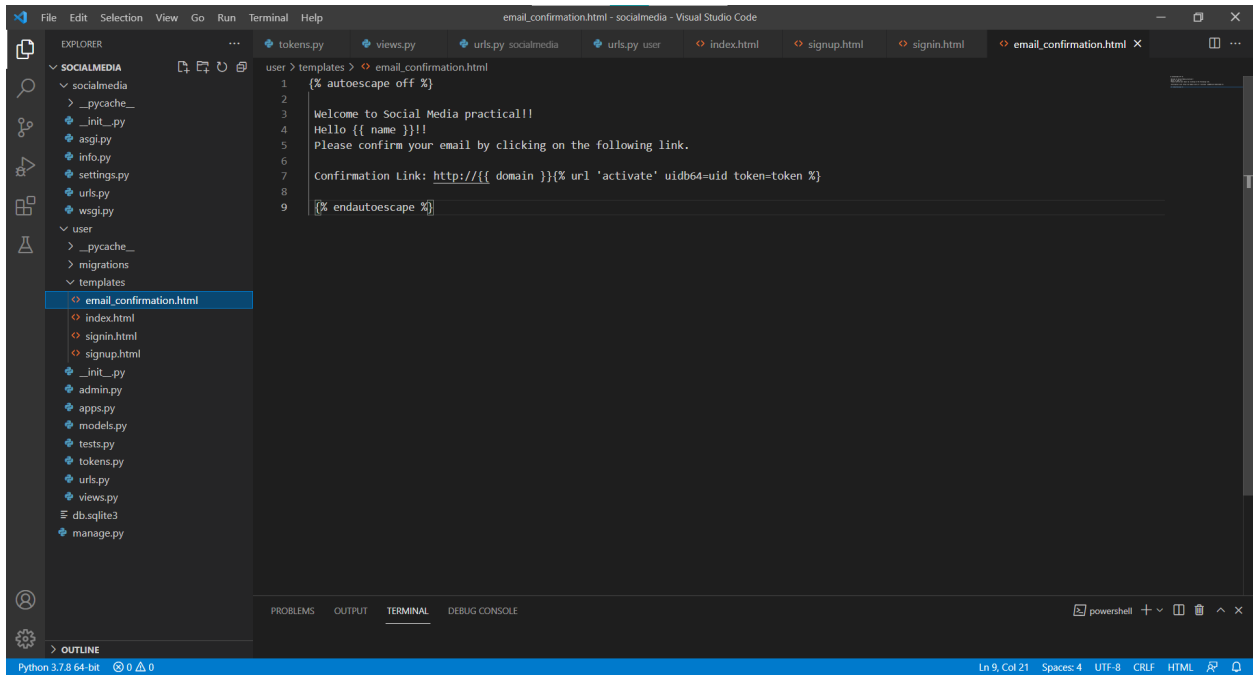
- admin.py, apps.py, models.py, tests.py files are left untouched.
- Create a new file called tokens.py

```
1 from django.contrib.auth.tokens import PasswordResetTokenGenerator
2
3 from six import text_type
4
5 class TokenGenerator(PasswordResetTokenGenerator):
6     def _make_hash_value(self, user, timestamp):
7         return (
8             text_type(user.pk) + text_type(timestamp)
9         )
10
11 generate_token = TokenGenerator()
```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\nirma\7th sem\Application development frameworks\lab work\socialmedia> python manage.py startapp user
PS D:\nirma\7th sem\Application development frameworks\lab work\socialmedia> cd user
work\socialmedia\user> mkdir templates
Directory: D:\nirma\7th sem\Application development frameworks\lab work\socialmedia\user

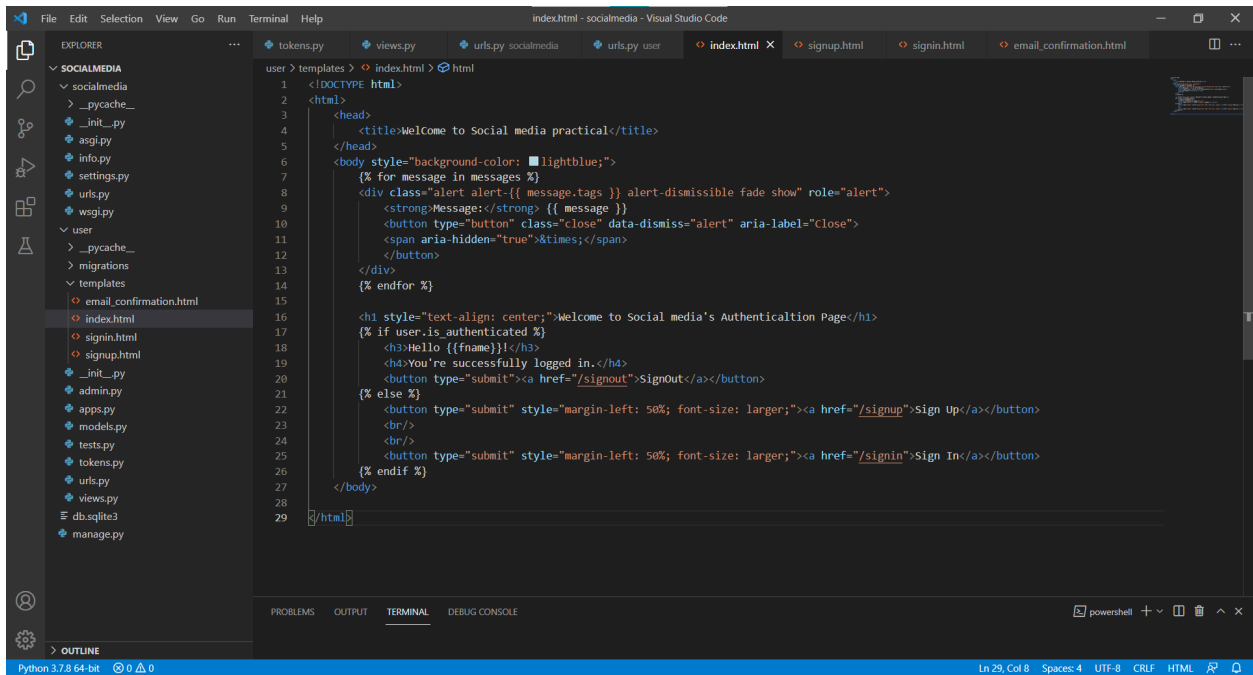
c. templates folder:

I. email_confirmation.html



```
1 {% autoescape off %}
2
3 Welcome to Social Media practical!!
4 Hello {{ name }}!!
5 Please confirm your email by clicking on the following link.
6
7 Confirmation Link: http://{{ domain }}{% url 'activate' uidb64-uid token-token %}
8
9 {% endautoescape %}
```

Ii. index.html



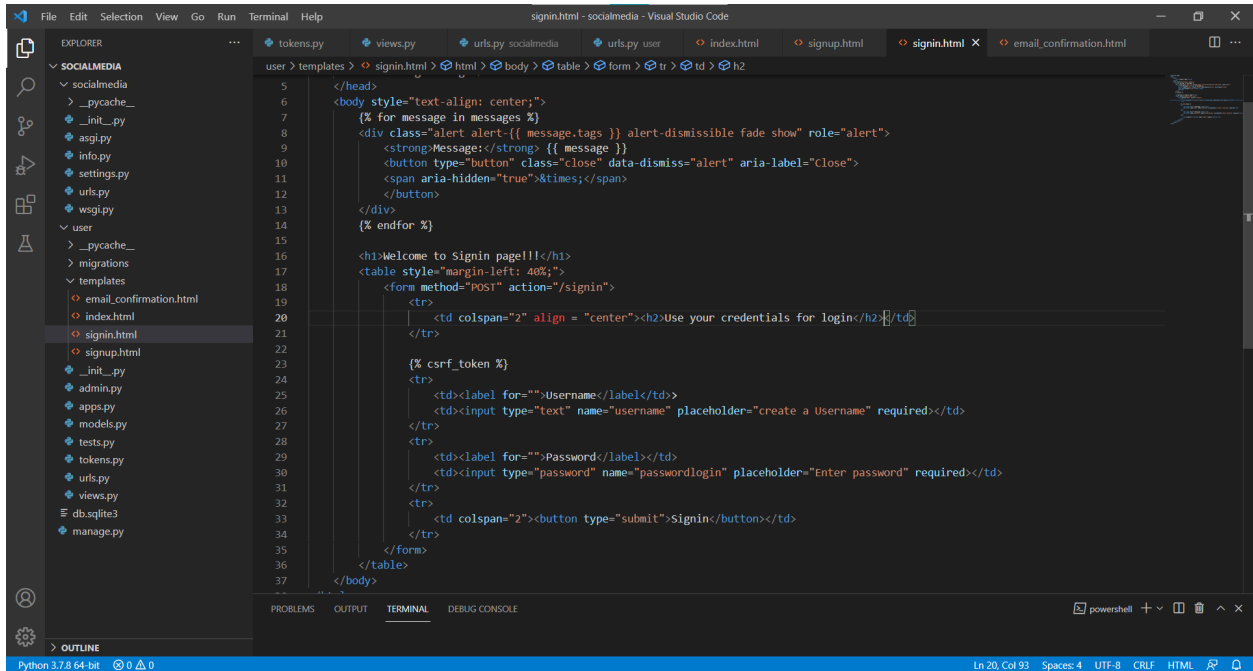
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Welcome to Social media practical</title>
5 </head>
6 <body style="background-color: #lightblue;">
7 <{% for message in messages %}
8 <div class="alert alert-{{ message.tags }}" alert-dismissible fade show" role="alert">
9 <strong>Messages</strong> {{ message }}
10 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
11 <span aria-hidden="true">&times;</span>
12 </button>
13 </div>
14 <{% endfor %}
15
16 <h1 style="text-align: center;">Welcome to Social media's Authenticaltion Page</h1>
17 <{% if user.is_authenticated %}
18 <h3>Hello {{fname}}!</h3>
19 <h4>You're successfully logged in.</h4>
20 <button type="submit"><a href="/signout">SignOut</a></button>
21
22 <{% else %}
23 <button type="submit" style="margin-left: 50%; font-size: larger;"><a href="/signup">Sign Up</a></button>
24 <br/>
25 <button type="submit" style="margin-left: 50%; font-size: larger;"><a href="/signin">Sign In</a></button>
26 <{% endif %}
27 </body>
28
29 </html>
```

Iii. signup.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <title>Signup Page</title>
6 </head>
7 <body style="text-align: center;">
8   {% for message in messages %}
9   <div class="alert alert-{{ message.tags }} alert-dismissible fade show" role="alert">
10     <strong>Message:</strong> {{ message }}
11     <button type="button" class="close" data-dismiss="alert" aria-label="close">
12       <span aria-hidden="true">&times;</span>
13     </button>
14   </div>
15   {% endfor %}
16
17 <h1>Welcome to Signup page!!!</h1>
18 <br>
19
20 <table style="margin-left: 40%;">
21   <form method="POST" action="/signup">
22     <tr>
23       <td colspan="2" align = "center"><h2>Create User</h2></td>
24     </tr>
25
26     {% csrf_token %}
27     <tr>
28       <td><label for="">Username</label></td>
29       <td><input type="text" name="username" placeholder="create a Username (only charcter and numbers are allowed)" requi
30     </tr>
31     <tr>
32       <td><label for="">First Name</label></td>
33       <td><input type="text" name="fname" placeholder="Enter your first name" required></td>
34     </tr>
35   </form>
36 </table>
37 </body>
38 </html>
```

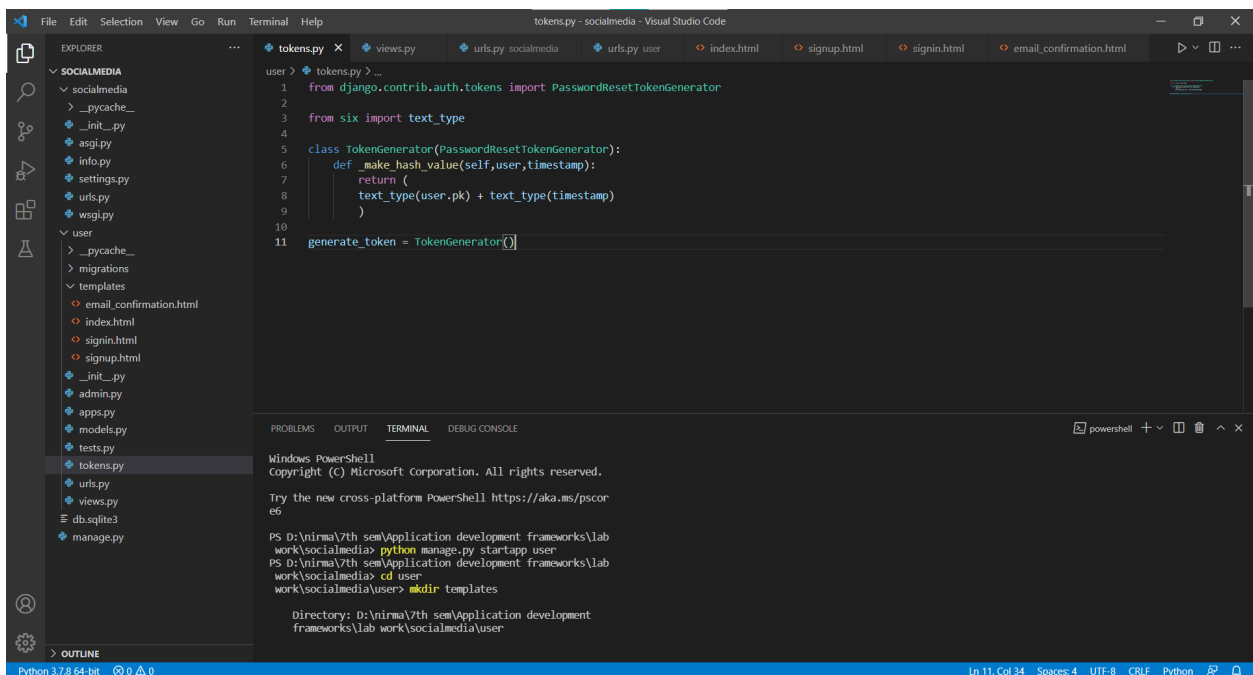
```
26   {% csrf_token %}
27   <td><label for="">Username</label></td>
28   <td><input type="text" name="username" placeholder="create a Username (only charcter and numbers are allowed)" requi
29   </tr>
30   <tr>
31     <td><label for="">First Name</label></td>
32     <td><input type="text" name="fname" placeholder="Enter your first name" required></td>
33   </tr>
34   <tr>
35     <td><label for="">Last Names</label></td>
36     <td><input type="text" name="lname" placeholder="Enter your last name" required></td>
37   </tr>
38   <tr>
39     <td><label for="">Email Address</label></td>
40     <td><input type="email" name="email" placeholder="Enter your email address" required></td>
41   </tr>
42   <tr>
43     <td><label for="">Password</label></td>
44     <td><input type="password" name="pass1" placeholder="Enter your password" required></td>
45   </tr>
46   <tr>
47     <td><label for="">Confirm Password</label></td>
48     <td><input type="password" name="pass2" placeholder="confirm your password" required></td>
49   </tr>
50   <tr>
51     <td colspan="2" align="center"><button type="submit">Sign up</button></td>
52   </tr>
53 </form>
54 </table>
55 </body>
56 </html>
```

Iv. signin.html



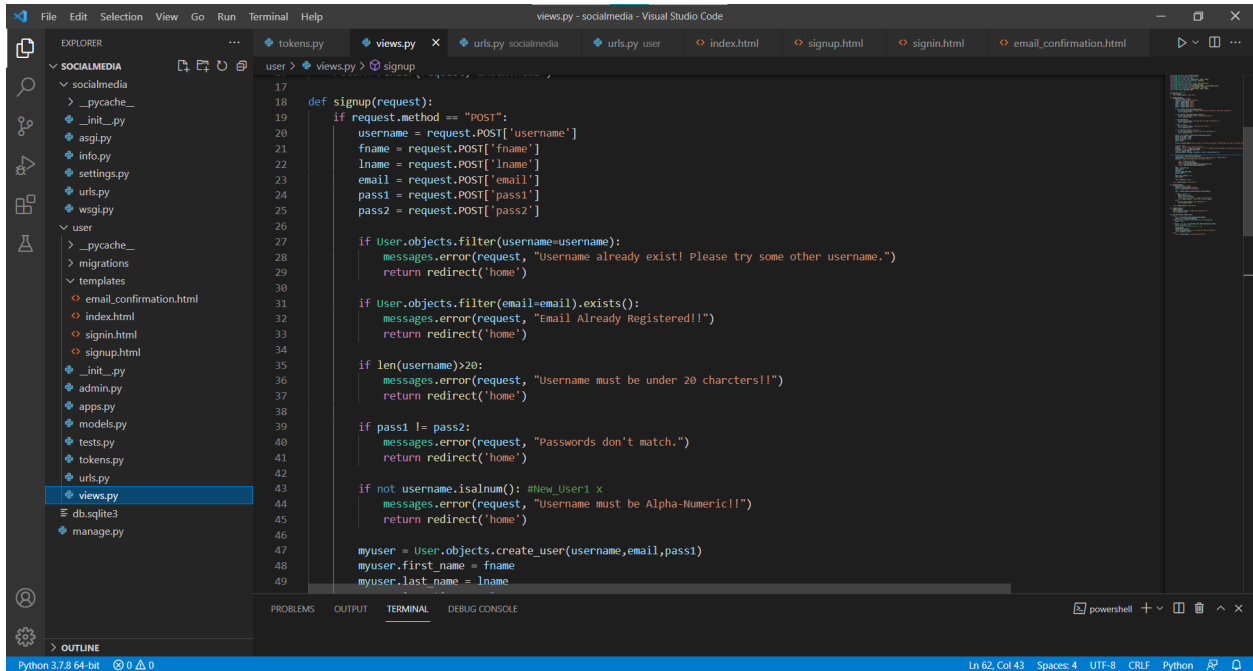
```
5 </head>
6 <body style="text-align: center;">
7     {% for message in messages %}
8     <div class="alert alert-{% message.tags %} alert-dismissible fade show" role="alert">
9         <strong>Message:</strong> {{ message }}
10        <button type="button" class="close" data-dismiss="alert" aria-label="Close">
11            <span aria-hidden="true">&times;</span>
12        </button>
13    </div>
14    {% endfor %}
15
16    <h1>Welcome to Signin page!!!</h1>
17    <table style="margin-left: 40%;>
18        <form method="POST" action="/signin">
19            <tr>
20                <td colspan="2" align = "center"><h2>Use your credentials for login</h2></td>
21            </tr>
22
23            {% csrf_token %}
24            <tr>
25                <td><label for="">Username</label></td>
26                <td><input type="text" name="username" placeholder="create a Username" required</td>
27            </tr>
28            <tr>
29                <td><label for="">Password</label></td>
30                <td><input type="password" name="passwordlogin" placeholder="Enter password" required</td>
31            </tr>
32            <tr>
33                <td colspan="2"><button type="submit">Signin</button></td>
34            </tr>
35        </form>
36    </table>
37 </body>
```

d. views.py file:

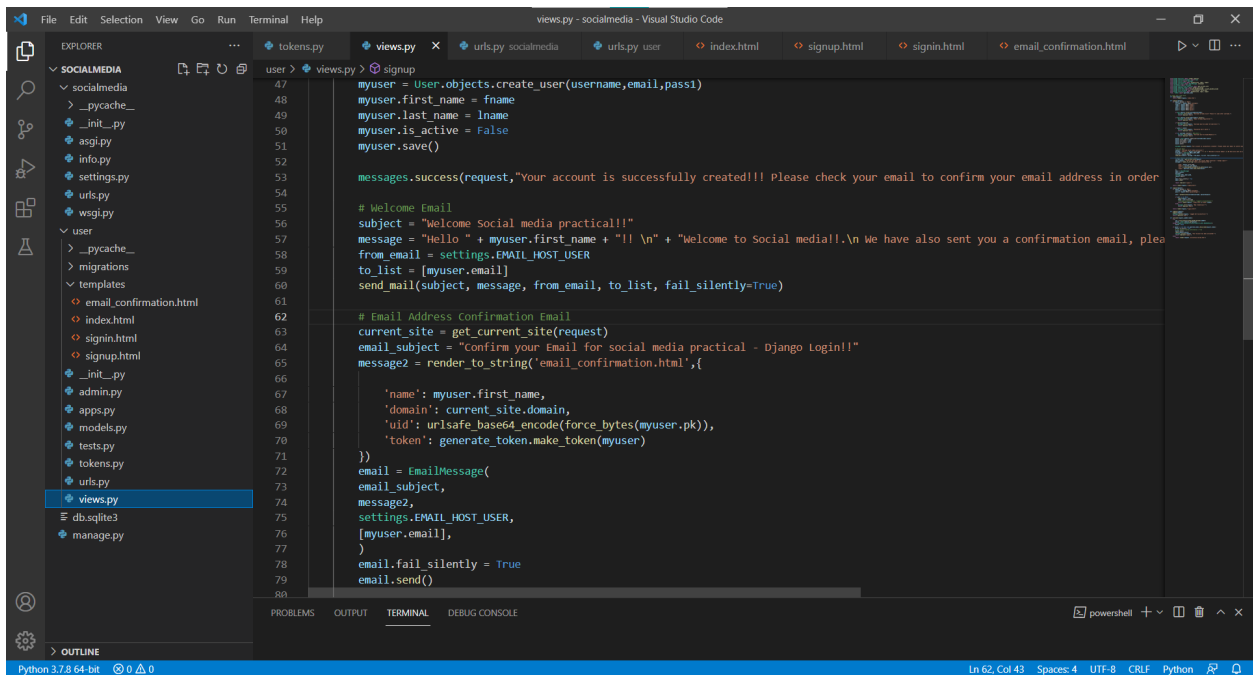


```
1 from django.contrib.auth.tokens import PasswordResetTokenGenerator
2
3 from six import text_type
4
5 class TokenGenerator(PasswordResetTokenGenerator):
6     def _make_hash_value(self, user, timestamp):
7         return (
8             text_type(user.pk) + text_type(timestamp)
9         )
10
11 generate_token = TokenGenerator()
```

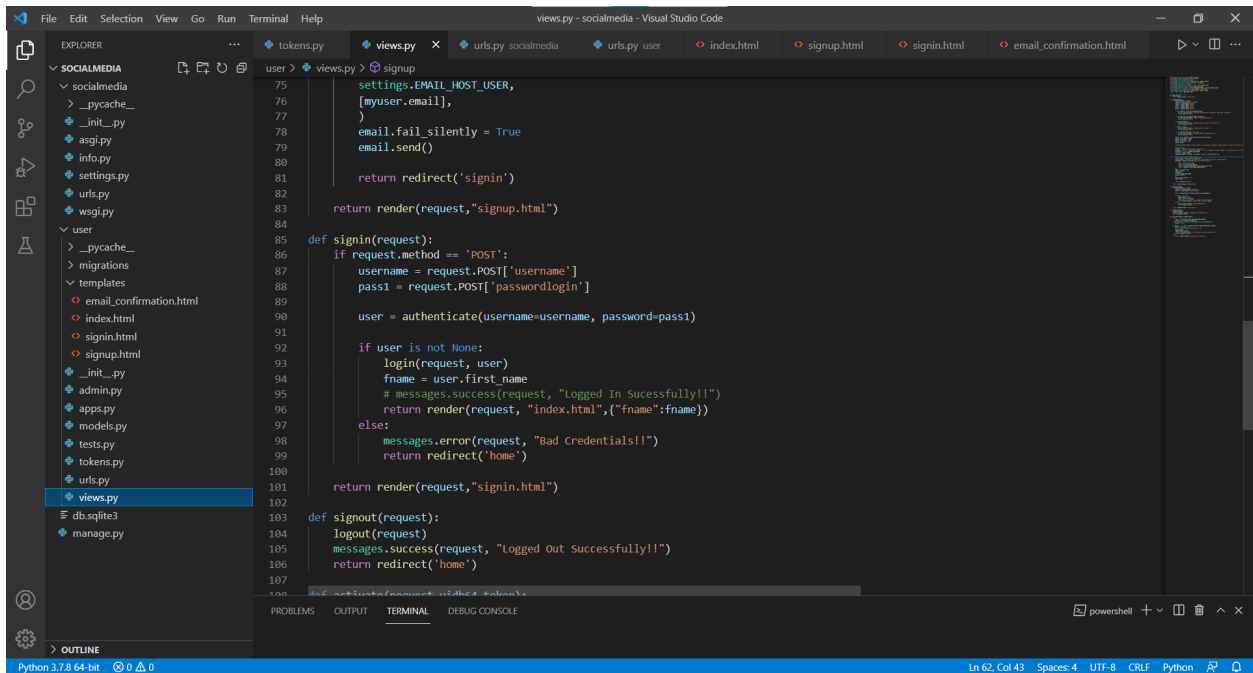
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS D:\nirma\7th sem\Application development frameworks\lab work\socialmedia> python manage.py startapp user
PS D:\nirma\7th sem\Application development frameworks\lab work\socialmedia> cd user
work\socialmedia\user> mkdir templates
Directory: D:\nirma\7th sem\Application development frameworks\lab work\socialmedia\user



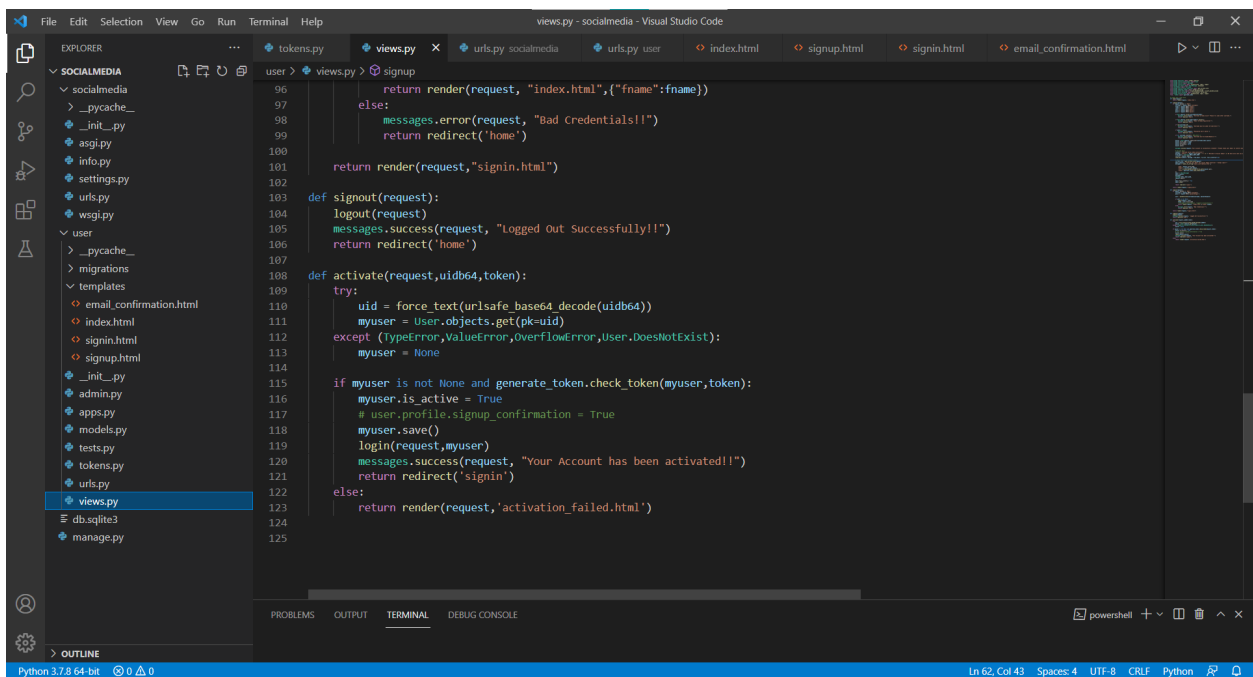
```
17
18 def signup(request):
19     if request.method == "POST":
20         username = request.POST['username']
21         fname = request.POST['fname']
22         lname = request.POST['lname']
23         email = request.POST['email']
24         pass1 = request.POST['pass1']
25         pass2 = request.POST['pass2']
26
27         if User.objects.filter(username=username):
28             messages.error(request, "Username already exist! Please try some other username.")
29             return redirect('home')
30
31         if User.objects.filter(email=email).exists():
32             messages.error(request, "Email Already Registered!!")
33             return redirect('home')
34
35         if len(username)>20:
36             messages.error(request, "Username must be under 20 charcters!!")
37             return redirect('home')
38
39         if pass1 != pass2:
40             messages.error(request, "Passwords don't match.")
41             return redirect('home')
42
43         if not username.isalnum(): #New User 1 x
44             messages.error(request, "Username must be Alpha-Numeric!!")
45             return redirect('home')
46
47         myuser = User.objects.create_user(username,email,pass1)
48         myuser.first_name = fname
49         myuser.last_name = lname
```



```
47         myuser = User.objects.create_user(username,email,pass1)
48         myuser.first_name = fname
49         myuser.last_name = lname
50         myuser.is_active = False
51         myuser.save()
52
53         messages.success(request,"Your account is successfully created!!! Please check your email to confirm your email address in order
54
55         # Welcome Email
56         subject = "Welcome Social media practical!!"
57         message = "Hello " + myuser.first_name + "!! \n" + "Welcome to Social media!!.\n We have also sent you a confirmation email, please
58         from_email = settings.EMAIL_HOST_USER
59         to_list = [myuser.email]
60         send_mail(subject, message, from_email, to_list, fail_silently=True)
61
62         # Email Address Confirmation Email
63         current_site = get_current_site(request)
64         email_subject = "Confirm your Email for social media practical - Django Login!!"
65         message2 = render_to_string('email_confirmation.html',{
66
67             'name': myuser.first_name,
68             'domain': current_site.domain,
69             'uid': urlsafe_base64_encode(force_bytes(myuser.pk)),
70             'token': generate_token.make_token(myuser)
71         })
72         email = EmailMessage(
73             email_subject,
74             message2,
75             settings.EMAIL_HOST_USER,
76             [myuser.email],
77         )
78         email.fail_silently = True
79         email.send()
80
```



```
75     settings.EMAIL_HOST_USER,
76     [myuser.email],
77 )
78     email.fail_silently = True
79     email.send()
80
81     return redirect('signin')
82
83     return render(request,"signup.html")
84
85 def signin(request):
86     if request.method == 'POST':
87         username = request.POST['username']
88         pass1 = request.POST['passwordlogin']
89
90         user = authenticate(username=username, password=pass1)
91
92         if user is not None:
93             login(request, user)
94             fname = user.first_name
95             # messages.success(request, "logged In Successfully!!")
96             return render(request, "index.html", {"fname":fname})
97         else:
98             messages.error(request, "Bad Credentials!!")
99             return redirect('home')
100
101     return render(request,"signin.html")
102
103 def signout(request):
104     logout(request)
105     messages.success(request, "Logged Out Successfully!!")
106     return redirect('home')
107
108
```



```
96     return render(request, "index.html", {"fname":fname})
97     else:
98         messages.error(request, "Bad Credentials!!")
99         return redirect('home')
100
101     return render(request,"signin.html")
102
103 def signout(request):
104     logout(request)
105     messages.success(request, "Logged Out Successfully!!")
106     return redirect('home')
107
108 def activate(request,uidb64,token):
109     try:
110         uid = force_text(urlsafe_base64_decode(uidb64))
111         myuser = User.objects.get(pk=uid)
112     except (TypeError,ValueError,OverflowError,User.DoesNotExist):
113         myuser = None
114
115     if myuser is not None and generate_token.check_token(myuser,token):
116         myuser.is_active = True
117         # user.profile.signup_confirmation = True
118         myuser.save()
119         login(request,myuser)
120         messages.success(request, "Your Account has been activated!!")
121         return redirect('signin')
122     else:
123         return render(request,'activation_failed.html')
124
125
```

e. urls.py file and created and:


```
1 from django.contrib import admin
2 from django.urls import path, include
3 from . import views
4
5
6 urlpatterns = []
7 path('', views.home, name='home'),
8 path('signup', views.signup, name='signup'),
9 path('signin', views.signin, name='signin'),
10 path('signout', views.signout, name='signout'),
11 path('activate/<uidb64>/<token>', views.activate, name='activate'),
12
```

6. urls.py (project level)

```
1 """socialmedia URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/3.2/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = []
20 path('admin/', admin.site.urls),
21 path('', include('user.urls')),
22
23
```

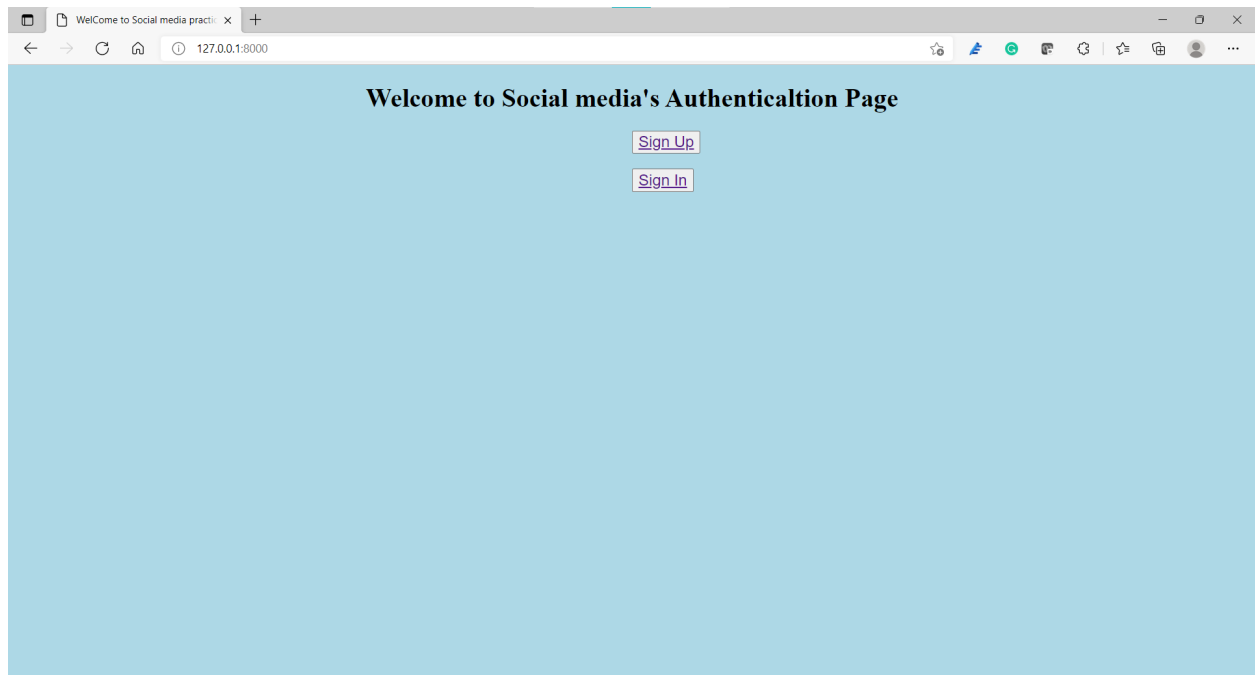
Output:

In the terminal run the following commands:
python manage.py makemigrations

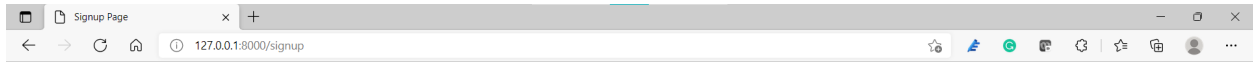
```
python manage.py migrate  
python manage.py runserver
```

Click on 127.0.0.1:8000

1. Welcome page:



2. Clicking on sign up:



Welcome to Signup page!!!

Create User

Username	<input type="text" value="create a Username (only G"/>
First Name	<input type="text" value="Enter your first name"/>
Last Name	<input type="text" value="Enter your last name"/>
Email Address	<input type="text" value="Enter your email address"/>
Password	<input type="text" value="Enter your password"/>
Confirm Password	<input type="text" value="Confirm your password"/>
<input type="button" value="Sign up"/>	

3. Filling in the details:

Welcome to Social media!! Inbox x



madtclass2017@gmail.com
to me ▾

10:22 PM (0 minutes ago) ☆ ↶ ⋮

Hello Labdhil!
Welcome to Social media!!
Thank you for visiting our website
. We have also sent you a confirmation email, please confirm your email address.

Thanking You
Developer Team

↶ Reply

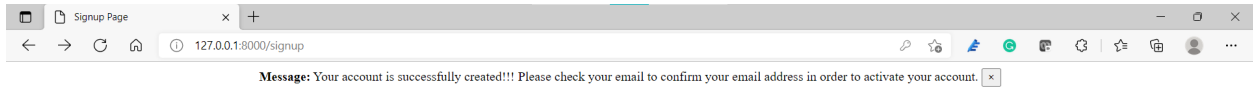
➦ Forward



Welcome to Signup page!!!

Create User

Username	<input type="text" value="world"/>
First Name	<input type="text" value="hello"/>
Last Name	<input type="text" value="world"/>
Email Address	<input type="text" value="codinggenz@gmail.com"/>
Password	<input type="password" value="*****"/>
Confirm Password	<input type="password" value="*****"/>
<input type="button" value="Sign up"/>	

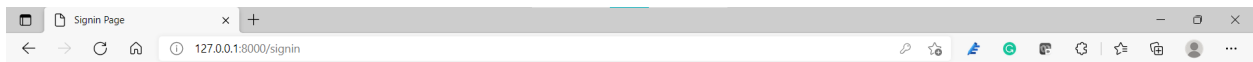


Welcome to Signup page!!!

Create User

Username	<input type="text" value="worlds"/>
First Name	<input type="text" value="hello"/>
Last Name	<input type="text" value="world"/>
Email Address	<input type="text" value="codinggenz@gmail.com"/>
Password	<input type="password" value="Enter your password"/>
Confirm Password	<input type="password" value="Confirm your password"/>
<input type="button" value="Sign up"/>	

4. Sign-in page:

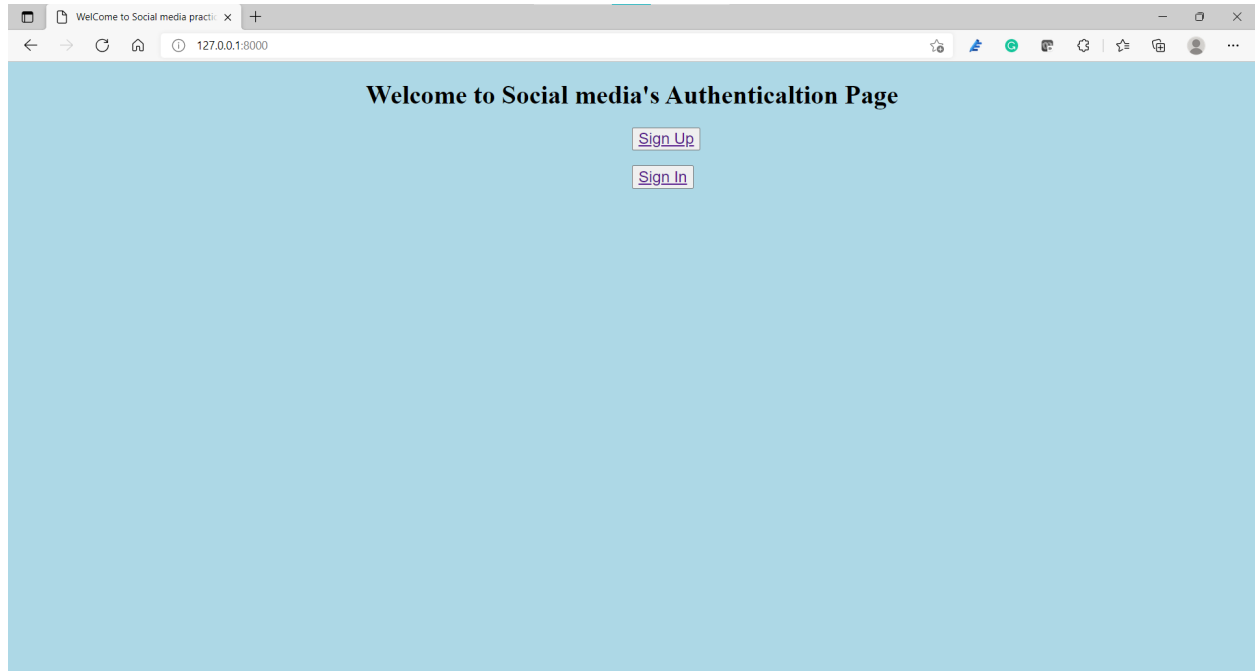


Welcome to Signin page!!!

Use your credentials for login

Username>	<input type="text" value="nirma"/>
Password	<input type="password" value="*****"/>
<input type="button" value="Signin"/>	

Successful sign in.



Conclusion:

We learned how to send emails through the Django app and make sign-up and sign-in pages.