Chapter-9

User Authentication And Authorization

User Authentication and Authorization

Authentication: The process of validating user is called authentication.

Authorization: The process of validating access permissions of user is called authorization.

Generally our web pages can be accessed by any person without having any restrictions. But some times our business requirement is to access a web page compulsory we have to register and login.Then only end user can able to access our page. To fulfill such of requirements we should go for Django authentication and authorization module. (auth application)

Django provides the following 2 in built applications for user authentication.

1) django.contrib.auth
2) django.contrib.contenttypes

auth application is authentication application of Django.
This auth application internally uses contenttypes application to track models installed in our database.

<u>Note:</u> To use Django in built authentication facility, compulsory these 2 applications should be in INSTALLED_APPS list. But from Django 1.10 onwards automatically these are available and we are not required to add explicitly.

Django uses PBKDF2_Sha256 algorithm to encrypt passwords and hence passwords won't be stored in plain text form and we can expect more security. Even superuser also cannot see any user's password.

Based on our requirement, we can use more secure hashing algorithms also like bcrypt and argon2. We can install with pip as follows.

pip install bcrypt
pip install django[argon2]

More secured algorithm is argon2 and then bcrypt followed PBKDF2.

In settings.py we have to configure password hashers as follows.

```
1)  PASSWORD_HASHERS=[
2)    'django.contrib.auth.hashers.Argon2PasswordHasher',
3)    'django.contrib.auth.hashers.BCryptSHA256PasswordHasher',
4)    'django.contrib.auth.hashers.BCryptPasswordHasher',
5)    'django.contrib.auth.hashers.PBKDF2PasswordHasher',
6)    'django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher',
7)  ]
```

Django will always consider from first to last. ie order is important.

Just like templates and static folder, we have to create media folder also.

Difference between Static and Media Folders:

- ☕ Static folder contains images, CSS files etc which are provided by application to the end user.
- ☕ But media folder contains the resources like images provided by end user to the application (like profile image etc)

How to Configure Media Folder in settings.py File:

MEDIA_DIR = os.path.join(BASE_DIR,'media')
MEDIA_ROOT = MEDIA_DIR
MEDIA_URL = '/media/'

from django.contrib.auth.decorators import login_required

@login_required
def java_exams_view(request):
    return render(request,'testapp/java.html')

If we use @login_required decorator for any view function,then auth application will check whether user login or not. If the user not login then the control will be forwarded to login page.

http://127.0.0.1:8000/accounts/login/?next=/java/

We have to configure auth application url patterns in urls.py file.

from django.conf.urls import url,include

url('accounts/', include('django.contrib.auth.urls')),
In our project auth application urls also included.

TemplateDoesNotExist at /accounts/login/registration/login.html

login link of auth application: /accounts/login/
logout link of auth application: /accounts/logout

After logout then Django default logout page will be displayed. Instead of this default target page we can configure our own target page inside settings.py as follows.
LOGOUT_REDIRECT_URL='/'

If we click login link explicitly and after login by default the control will goes to
http://127.0.0.1:8000/accounts/profile/
But we can configure our own target page after login inside settings.py as follows.

LOGIN_REDIRECT_URL='/'

Authentication Application:

views.py

```
1)  from django.shortcuts import render
2)  from django.contrib.auth.decorators import login_required
3)
4)  # Create your views here.
5)  def home_page_view(request):
6)     return render(request,'testapp/home.html')
7)
8)  @login_required
9)  def java_exams_view(request):
10)    return render(request,'testapp/javaexams.html')
11) @login_required
12) def python_exams_view(request):
13)    return render(request,'testapp/pythonexams.html')
14) @login_required
15) def aptitude_exams_view(request):
16)    return render(request,'testapp/aptitudeexams.html')
17)
18) def logout_view(request):
19)    return render(request,'testapp/logout.html')
```

urls.py

```
1)  from django.conf.urls import url,include
2)  from django.contrib import admin
3)  from testapp import views
4)
5)  urlpatterns = [
6)     url(r'^admin/', admin.site.urls),
7)     url('accounts/',include('django.contrib.auth.urls')),
8)     url(r'^$', views.home_page_view),
9)     url(r'^python/', views.python_exams_view),
10)    url(r'^java/', views.java_exams_view),
11)    url(r'^aptitude/', views.aptitude_exams_view),
12)    url(r'^logout/', views.logout_view),
13) ]
```

auth123.css

```
1)  body{
```

```
2)    background: #efb917;
3)    color:blue;
4)  }
5)  .jumbotron{
6)    background: green;
7)    color:white;
8)  }
```

base.html

```html
1)  <!DOCTYPE html>
2)  {%load staticfiles%}
3)  <html lang="en" dir="ltr">
4)    <head>
5)     <meta charset="utf-8">
6)     <title></title>
7)     <!-- Latest compiled and minified CSS -->
8)  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
9)  <link rel="stylesheet" href="{%static "css/auth123.css"%}">
10)  </head>
11)  <body>
12)   <nav class='navbar navbar-default navbar-inverse'>
13)    <div class="container">
14)     <div class="navbar-header">
15)      <a class="navbar-brand" href="/">DURGAEXAMS</a>
16)     </div>
17)     <ul class='nav navbar-nav'>
18)      <li> <a href="/java">Java Exams</a> </li>
19)      <li> <a href="/python">Python Exams</a> </li>
20)      <li> <a href="/aptitude">Aptitude Exams</a> </li>
21)     </ul>
22)     <ul class='nav navbar-nav navbar-right'>
23)      <li> <a href="#">Signup</a> </li>
24)      <li> <a href="/accounts/login/">Login</a> </li>
25)      <li> <a href="/accounts/logout/">Logout</a> </li>
26)     </ul>
27)    </div>
28)   </nav>
29)   {%block body_block%}
30)   {%endblock%}
31)  </body>
32) </html>
```

home.html

```
1)  <!DOCTYPE html>
2)  {%extends 'testapp/base.html'%}
3)     {%block body_block%}
4)     <div class="jumbotron" >
5)      <div class="container">
6)       <h1>Welcome to Durga Exams</h1>
7)       <h2>Rules:</h2>
8)       <ul>
9)        <li>Rule-1:You should write only one exam per day</li>
10)        <li>Rule-1:You should write only one exam per day</li>
11)        <li>Rule-1:You should write only one exam per day</li>
12)        <li>Rule-1:You should write only one exam per day</li>
13)        <li>Rule-1:You should write only one exam per day</li>
14)        <li>Rule-1:You should write only one exam per day</li>
15)        <li>Rule-1:You should write only one exam per day</li>
16)       </ul>
17)      </div>
18)     </div>
19)     {%endblock%}
```

javaexams.html

```
1)  <!DOCTYPE html>
2)  {%extends 'testapp/base.html'%}
3)     {%block body_block%}
4)     <div class="jumbotron" >
5)      <div class="container" align='center'>
6)       <h1>Welcome to Java Exams</h1>
7)      </div>
8)     </div>
9)     {%endblock%}
```

pythonexams.html

```
1)  <!DOCTYPE html>
2)  {%extends 'testapp/base.html'%}
3)     {%block body_block%}
4)     <div class="jumbotron" >
5)      <div class="container" align='center'>
6)       <h1>Welcome to Python Exams</h1>
```

```
7)        </div>
8)        </div>
9)     {%endblock%}
```

aptitudeexams.html

```
1)  <!DOCTYPE html>
2)  {%extends 'testapp/base.html'%}
3)     {%block body_block%}
4)     <div class="jumbotron" >
5)      <div class="container" align='center'>
6)       <h1>Welcome to Aptitude Exams</h1>
7)      </div>
8)     </div>
9)     {%endblock%}
```

registration/login.html

```
1)  <!DOCTYPE html>
2)  {%extends 'testapp/base.html'%}
3)     {%block body_block%}
4)      <div class="container" align='center'>
5)       <h1>Login to write Exams...</h1>
6)       <form  method="post">
7)        {{form.as_p}}
8)        {%csrf_token%}
9)        <button type="submit">Login</button>
10)       </form>
11)      </div>
12)     {%endblock%}
```

testapp/logout.html

```
1)  <!DOCTYPE html>
2)  {%extends 'testapp/base.html'%}
3)     {%block body_block%}
4)     <div class="jumbotron" >
5)      <div class="container">
6)       <h1>Thanks for visiting Durgaexams</h1>
7)       <p>Anyway we are feeling very sad b'z you logout atleast 24 hours per day
     you have to spend with our website like facebook</p>
8)       <h2>Please Login Again:</h2>
9)       <a href="/accounts/login" class="btn btn-primary btn-lg btn-success">
     Login</a>
10)       </div>
```

```
11)    </div>
12)    {%endblock%}
```

settings.py

```
1) LOGIN_REDIRECT_URL='/'
2) LOGOUT_REDIRECT_URL='/logout'
```

Configure Signup Form:

forms.py

```
1) from django import forms
2) from django.contrib.auth.models import User
3) class SignUpForm(forms.ModelForm):
4)    class Meta:
5)       model=User
6)       fields=['username','password','email','first_name','last_name']
```

views.py

```
1) def signup_view(request):
2)    form=SignUpForm()
3)    if request.method=='POST':
4)      form=SignUpForm(request.POST)
5)      user=form.save()
6)      user.set_password(user.password)
7)      user.save()
8)      return HttpResponseRedirect('/accounts/login')
9)   return render(request,'testapp/signup.html',{'form':form})
```

urls.py

```
1) urlpatterns = [
2)    ....
3)    url(r'^signup/', views.signup_view),
4) ]
```

Password Hashers in settings.py

```
1) PASSWORD_HASHERS=[
2) #'django.contrib.auth.hashers.Argon2PasswordHasher',
3) 'django.contrib.auth.hashers.BCryptSHA256PasswordHasher',
4) 'django.contrib.auth.hashers.BCryptPasswordHasher',
```

```
5)   'django.contrib.auth.hashers.PBKDF2PasswordHasher',
6)   'django.contrib.auth.hashers.PBKDF2SHA1PasswordHasher',
7)   ]
```

Future Assignments:

1. How to customize our own login form
2. How to use auth application provided default signup form
3. Social Login