Course Module 1: Course Unit 12: Django (Authentications)

Week 15

Django (Authentications)

Objectives:

1. Explain Django authentications.
2. Complete the program to manipulate Django authentications.
3. Devise adding authentications on the previous programs.

Django

Django

- high-level Python web application framework (enables the rapid development of web applications)
- backend framework used to resolve problems of connectivity with databases, other server problems, SEO (Search Engine Optimization) solutions

There are two types of websites: **static** and **dynamic**. **Django** is a framework for developing *dynamic websites*.
While a static website is one that solely presents information, there is no interaction (beyond simple page requests) that gets registered to a server. In a static website, the server sends HTML, CSS, and JavaScript to a client and that's it.
More capabilities require a dynamic website, where the server stores information and responds to user interaction beyond just serving pages. One major reason to develop a dynamic site is to **authenticate users** and **restrict content**.

Django Authentications

Django Authentications

- Django comes with a user authentication system. It handles user accounts, groups, permissions and cookie-based user sessions.
- The Django authentication system handles both authentication and authorization. Briefly, **authentication** verifies a user is who they claim to be, and **authorization** determines what an authenticated user is allowed to do. Here the term authentication is used to refer to both tasks.

Django Authentications

- Django provides an authentication and authorization ("permission") system, built on top of the session framework that allows you to verify user credentials and define what actions each user is allowed to perform. The framework includes built-in models for Users and Groups (a generic way of applying permissions to more than one user at a time), permissions/flags that designate whether a user may perform a task, forms and views for logging in users, and view tools for restricting content.

Django Authentications

The auth system consists of:

- **Users**
- **Permissions**: Binary (yes/no) flags designating whether a user may perform a certain task.
- **Groups**: A generic way of applying labels and permissions to more than one user.
- A **configurable password hashing system**
- **Forms and view tools** for logging in users, or restricting content
- A **pluggable backend system**

Django Authentications

The authentication system in Django aims to be very generic and doesn’t provide some features commonly found in web authentication systems. Solutions for some of these common problems have been implemented in third-party packages:

- Password strength checking
- Throttling of login attempts
- Authentication against third-parties (OAuth, for example)
- Object-level permissions

Django Authentications

Installation

Authentication support is bundled as a Django contrib module in **django.contrib.auth**. By default, the required configuration is already included in the *settings.py* generated by *django-admin startproject*, these consist of two items listed in your *INSTALLED_APPS* setting:

Django Authentications

Installation

- '**django.contrib.auth**' contains the core of the authentication framework, and its default models.
- '**django.contrib.contenttypes**' is the Django content type system, which allows permissions to be associated with models you create.

Django Authentications

Installation

and these items in your MIDDLEWARE setting:

- **SessionMiddleware** manages sessions across requests.
- **AuthenticationMiddleware** associates users with requests using sessions.

With these settings in place, running the command manage.py migrate creates the necessary database tables for auth related models and permissions for any models defined in your installed apps.

```
INSTALLED_APPS = [
    ...
    'django.contrib.auth',  #Core authentication
framework and its default models.
    'django.contrib.contenttypes',  #Django content
type system (allows permissions to be associated with
models).
    ....

MIDDLEWARE = [
MIDDLEWARE = [
    ...

'django.contrib.sessions.middleware.SessionMiddleware',
#Manages sessions across requests
    ...

'django.contrib.auth.middleware.AuthenticationMiddlewar
e',  #Associates users with requests using sessions.
    ....
```

Django Authentications

Usage

Using Django's default implementation

- ➢ Working with User objects
- ➢ Permissions and authorization
- ➢ Authentication in web requests
- ➢ Managing users in the admin

User authentication in Django | Django documentation | Django (djangoproject.com)

Django Tutorial Part 8: User authentication and permissions - Learn web development | MDN (mozilla.org)

Applications???

Any questions???

REFERENCES:

- Django Tutorial. (2022). https://data-flair.training/blogs/django-tutorial/
- Django Tutorial. (2020). https://www.geeksforgeeks.org/django-tutorial/
- Django Tutorial. (2022). https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction
- Django Tutorial. (n.d.). https://www.tutorialspoint.com/django/index.htm
- Django Tutorial. (n.d.). https://www.w3schools.com/django/index.php
- Getting started with Django. (n.d.). https://www.djangoproject.com/start/