

Task 5: User Authentication and Static Files Management

Objective

This task focuses on adding user authentication and managing static and media files in a Django project.

By completing this task, you will learn how to secure certain parts of your web app allow users to log in log out register and upload profile pictures, as well as properly handle and display static and uploaded media files.

Questions to Explore

User Authentication

1. What is Django's built-in authentication system, and what problems does it solve?
2. How web applications save the password and why?
3. What's the difference between hashing, encoding, and encryption?
4. What are sessions and cookies, and how do they help keep users logged in?
5. Where do sessions save?
6. What are the main authentication views Django provides (e.g., LoginView, LogoutView, PasswordChangeView)?
7. How can we restrict access to certain pages (for example, using @login_required or LoginRequiredMixin)?
8. How can we create a custom user registration form?
9. What is Django default user model?
10. How we can extend Django default user model?
11. How can we connect the User model to other models (like linking a Developer to a User)?
12. Should we extend user model or use one to one relationship?
13. Should we use Django default model without extending?
14. What is different method of authentication (look for differences)? (jwt, session)

Static & Media Files

7. What are static files and media files in Django?
 8. How do we configure STATIC_URL, STATICFILES_DIRS, MEDIA_URL, and MEDIA_ROOT in settings.py?
 9. How can we use the {% static %} tag to display CSS or images in templates?
 10. How do we allow users to upload and display profile pictures or files?
 11. What's the difference between serving static and media files during development?
-

Deliverables

You will extend your previous Django project by adding authentication and static/media file handling.

1. Authentication System

- Add pages for user registration, login, and logout.
- Restrict all CRUD pages (Developers/Projects) to logged-in users only.
- Use @login_required or LoginRequiredMixin in your views.
- Add a simple navigation bar showing the logged-in username and a “Logout” button.

2. Profile and Media Upload

- Create a Profile model that links to Django’s User model using a OneToOneField.
- Add fields like bio and profile_picture.
- Allow users to upload and update their profile pictures through a form.
- Display profile details (username, bio, picture) on a “My Profile” page.

3. Static Files Setup

- Add at least one custom CSS file and connect it in your templates using {% static %}.
 - Use your own logo or header image stored in the static directory.
 - Make sure uploaded profile pictures are stored and displayed correctly from the /media/ folder.
-

Bonus (Extra 5 Marks)

- Add password reset functionality using Django's built-in auth views.
 - Create a simple API endpoint /api/developers/ that returns all developers in JSON format using Django REST Framework.
-

References

- Django Authentication: <https://docs.djangoproject.com/en/5.0/topics/auth/>
 - Django Static & Media Files: <https://docs.djangoproject.com/en/5.0/howto/static-files/>
 - Django REST Framework: <https://www.django-rest-framework.org/>
-