

# Django Important Topics - Notes

## 1. Basics & Setup

- Installing Django
- Project vs App (difference & structure)
- manage.py commands (runserver, migrate, createsuperuser, etc.)

## 2. Django Project Structure

- settings.py → configurations (databases, installed apps, middleware)
- urls.py → URL routing
- wsgi.py & asgi.py basics

## 3. Apps & Views

- Function-based views (FBV)
- Class-based views (CBV)
- render(), redirect()
- HttpResponse & JsonResponse

## 4. Templates

- Template rendering (.html files)
- Template tags & filters ({% if %}, {% for %}, {{ variable }})
- Template inheritance (base.html)

## 5. Models & Database

- Defining models (models.py)
- Model fields (CharField, IntegerField, ForeignKey, etc.)
- ORM basics & QuerySets (filter, get, all, order\_by)
- Migrations (makemigrations, migrate)

## 6. Admin Panel

- Creating superuser
- Registering models in admin.py
- Customizing admin

## 7. Forms & User Input

- Django Forms
- ModelForms
- Handling POST/GET requests

- Validation & CSRF protection

## **8. Authentication & Authorization**

- User model & authentication system
- Login, Logout, Signup
- Permissions & groups
- @login\_required decorator

## **9. Middleware**

- Definition and role of middleware
- Examples: AuthenticationMiddleware, CSRF middleware

## **10. Static & Media Files**

- Static files (CSS, JS, images)
- Media files (user uploads)

## **11. Error Handling**

- 404 & 500 error pages
- Custom error views

## **12. Advanced (Optional but Useful)**

- Class-Based Views in depth (ListView, DetailView, CreateView, UpdateView, DeleteView)
- Django Rest Framework (DRF) basics (for APIs)
- Signals (post\_save, pre\_delete)
- Using Postman for API testing