**Django Web Framework – Report**

**What is Django?**

Django is a high-level Python web framework that enables rapid development and clean, pragmatic design. It follows the MVT (Model-View-Template) architecture, making it easy to build and scale web applications.

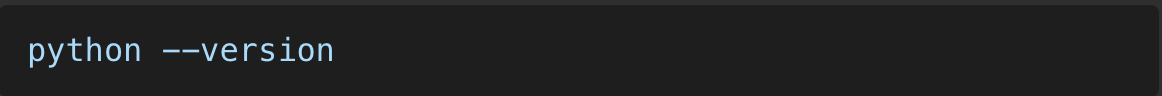
**Features of Django:**

* Batteries-Included: Comes with authentication, ORM, admin panel, and more.
* Scalability: Used by large companies like Instagram, Pinterest, and Spotify.
* Security: Protects against SQL injection, XSS, CSRF, and other attacks.
* Rapid Development: Reduces boilerplate code and enhances productivity.

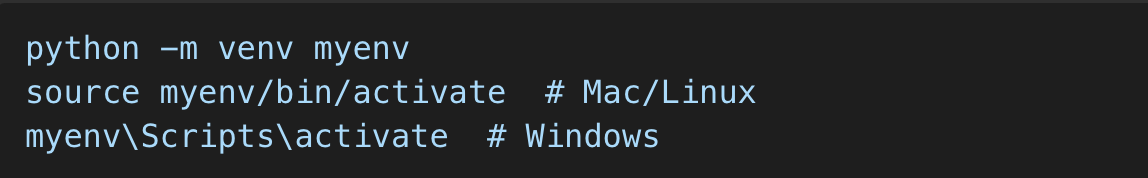
**Setting Up Django Development Environment**

Step 1: Install Python (If Not Installed)

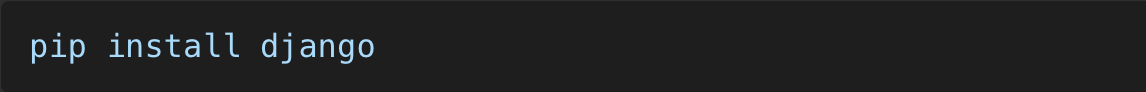
Ensure you have Python 3.x installed:



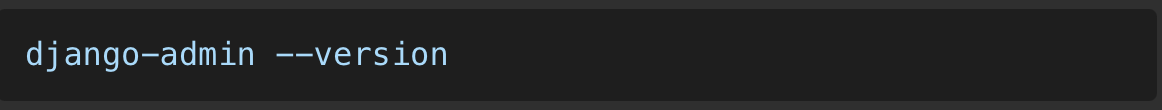
Step 2: Create a Virtual Environment



Step 3: Install Django



Check installations:

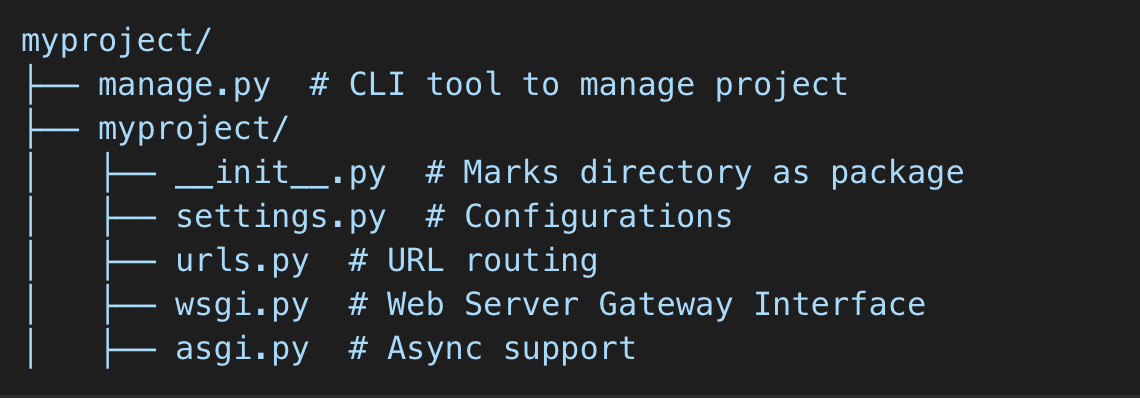


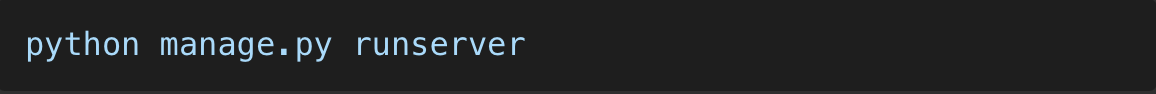
**Django Project and App Structure**

Creating a Django Project:

****

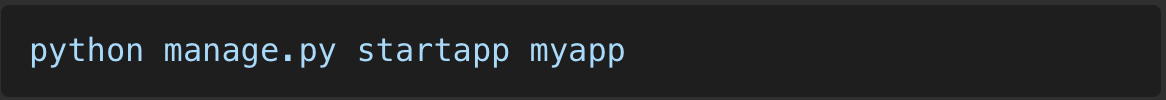
This creates:



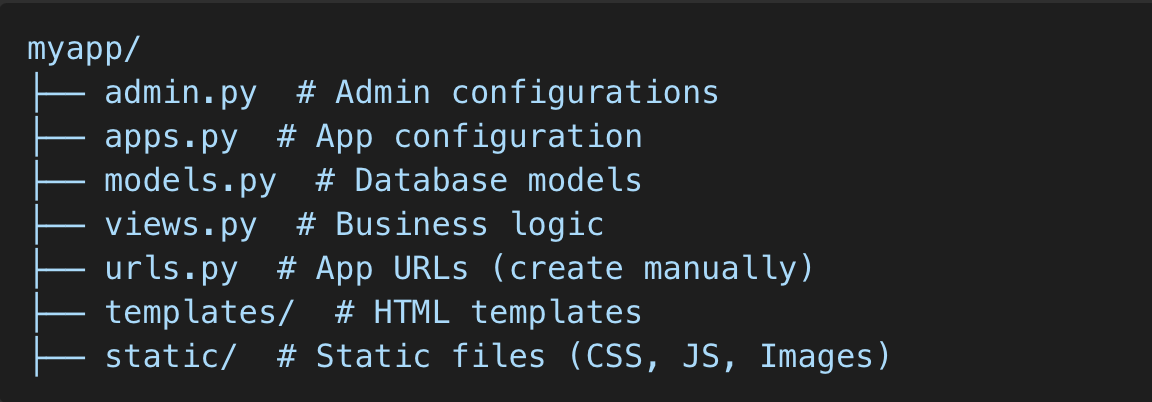
Run the server:

Visit http://127.0.0.1:8000/ to see the Django welcome page.

**Creating an App:**



This creates:

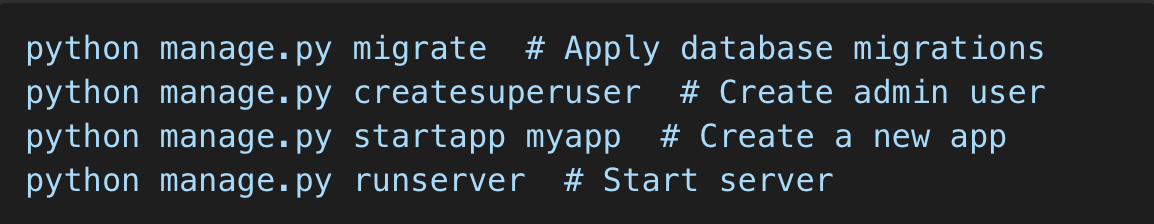


Register the app in settings.py:



Django Admin & Manage.py Commands:

Common commands:



**Understanding MVT (Model-View-Template) Architecture:**

Django follows MVT (Model-View-Template):

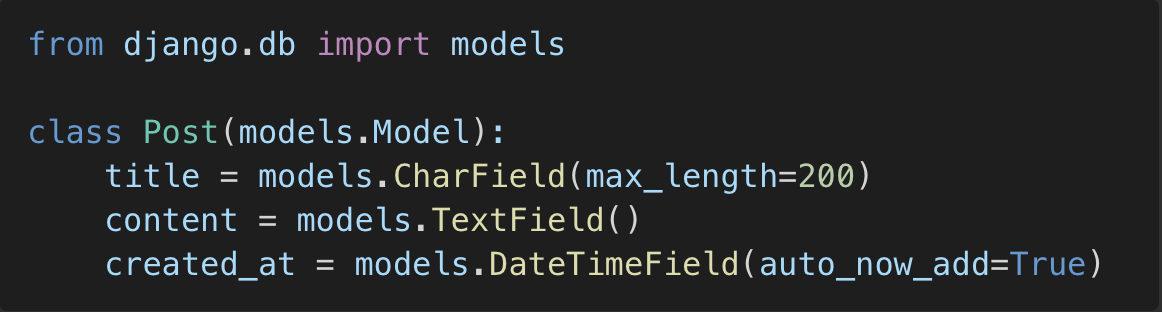
**Model (M) – Manages database interactions.**

**View (V) – Handles business logic.**

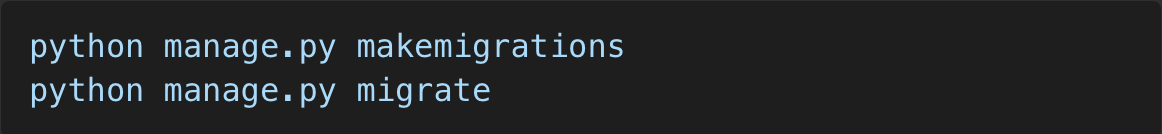
**Template (T) – Renders UI using HTML.**

Example: Implementing MVT in Django

Step 1: Define a Model (Database Table):



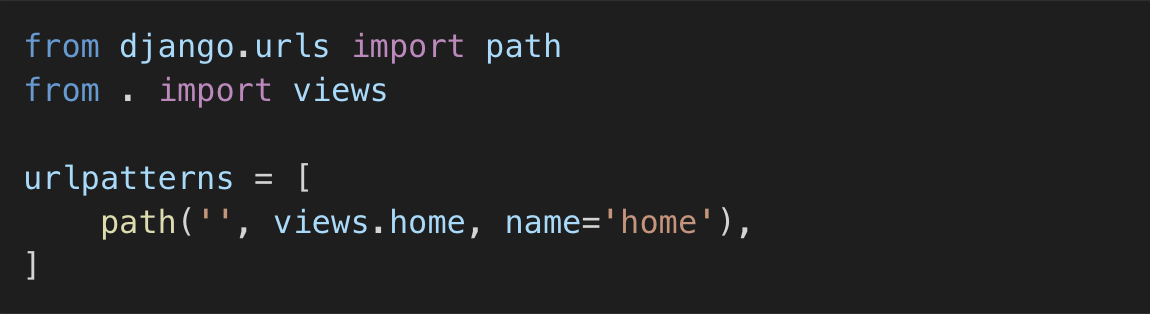
Run migrations:



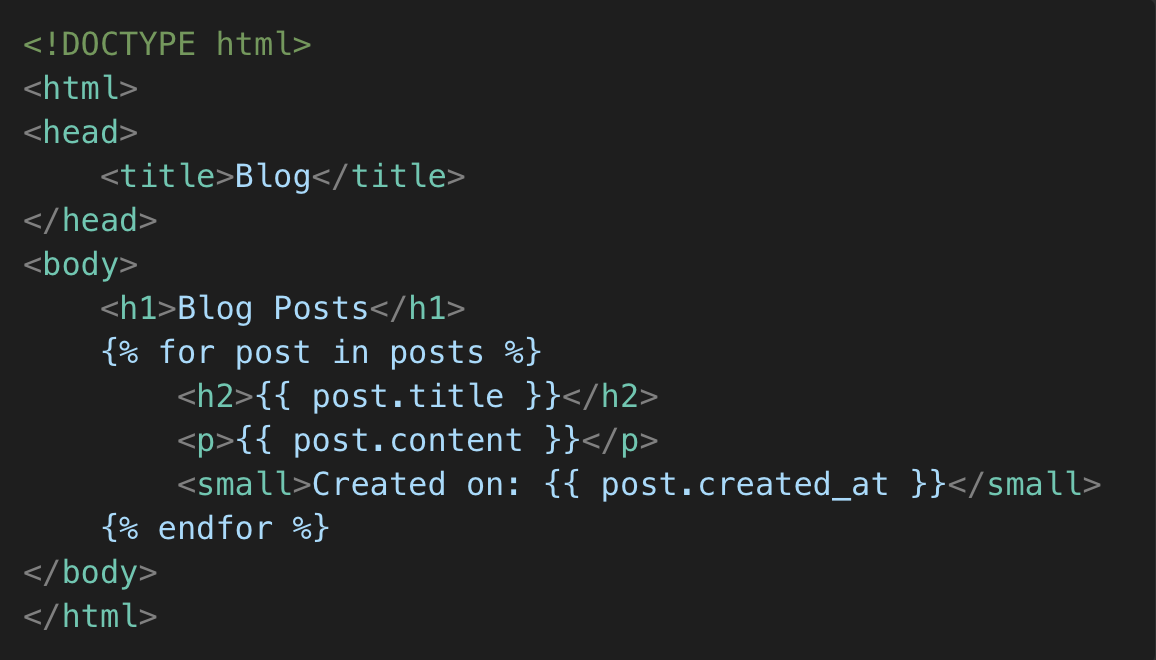
#### Step 2: Create a View



#### Step 3: Define URL Patterns



#### Step 4: Create a Template



**Essential Concepts:**

* **Django ORM** – Interact with databases using Python instead of SQL.
* **Django Admin Panel** – Built-in admin interface to manage models.
* **Migrations** – Used to apply changes to the database schema.
* **Static & Media Files Handling** – For serving CSS, JS, and images.
* **Authentication System** – Built-in user management (login, logout, permissions).
* **Middleware** – Functions that process requests before reaching the view.