**Django Web Framework Course - Setup and Initial Workflow**

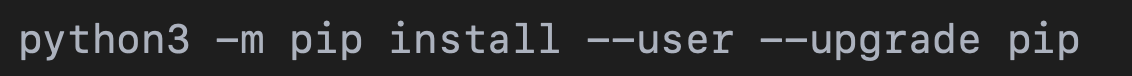
**Objective:** To set up a development environment for working with Django, including creating virtual environments, managing Python packages, and understanding the basic structure of a Django project.

**1. Setting Up Virtual Environment:**

A virtual environment helps isolate dependencies for each project, preventing package conflicts with the global Python installation.

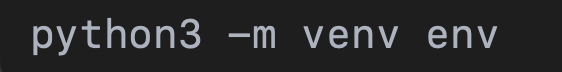
• **Installing pip:**

• Ensure the latest version of pip is installed using the command:



• **Creating a Virtual Environment:**

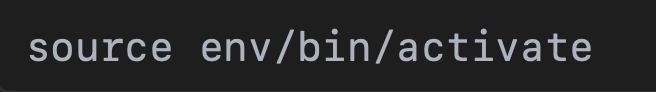
• The virtual environment is created using the following command:



• env is the name of the virtual environment, and it will store isolated Python installations and dependencies.

• **Activating the Virtual Environment:**

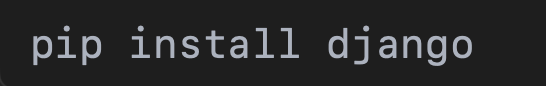
• On macOS, activate the virtual environment by running:



• The (env) prefix indicates the virtual environment is active.

• **Installing Django:**

• After activation, install Django within the environment:



• **Deactivating the Virtual Environment:**

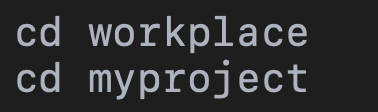
• To exit the virtual environment, run:



**2. Working with VS Code:**

• **Navigating to Project Directory:**

• Use the terminal to navigate to your project directory:

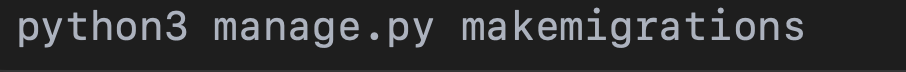


• **Running Django Commands:**

• To run the development server:



• To create migrations:



• To apply the migrations:



**Conclusion:**

The setup process for Django involves creating a virtual environment, activating it, installing Django, and then running essential Django commands to manage migrations and run the development server. VS Code layout tools, combined with SQLite extensions, streamline the workflow and make it easier to handle both the development and database management aspects of the project.

**Django Web Framework Course - Django Project Setup**

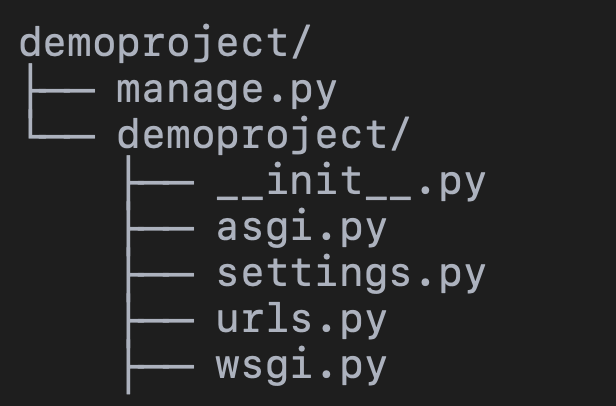
**1. Project Creation:**

• **Command Executed:**

django-admin startproject demoproject

• **Project Structure:**

The following structure was created:



• manage.py: A script to run administrative tasks (local to the project).

• demoproject/: A Python package that contains configuration files.

**Development Server Start:**

• **Command Executed:**

python manage.py runserver

• **Server Status:**

• The server is running on: http://127.0.0.1:8000/

• **Result:** The default Django welcome page was accessed successfully, confirming the server is working.

**3. Unapplied Migrations:**

• **Message:**

“You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.”

• **Solution:**

Run python manage.py migrate to apply the migrations and configure the database properly.

**4. Next Steps:**

• Apply the migrations with the command:

python manage.py migrate

• Restart the server and verify that the migrations are applied successfully.

**5. Conclusion:**

• The Django project setup was completed without conflicts.

• The next step is to apply the migrations to ensure the database is synchronized with the models.