TASK 03

1. Virtual Environment

A virtual environment is a self-contained directory with its own Python and package setup, isolated from the system's Python environment. It solves dependency conflicts between different projects by isolating packages.

Usage:

- Create: python -m venv myenv
- Activate:
    - Windows: myenv\Scripts\activate
    - macOS/Linux: source myenv/bin/activate
- Install packages: pip install <package_name>
- Deactivate: deactivate

2. What is Django?

Django is a Python web framework for building dynamic websites, emphasizing rapid development, security, and scalability. It follows the MTV (Model-Template-View) architecture.

3. Django vs. Django REST Framework (DRF)

- Django: Full-stack framework for web apps with HTML rendering.
- DRF: Extension for building RESTful APIs that return JSON responses, handling serialization and authentication.

4. Django Architecture

Follows the MTV (Model-Template-View) pattern:

- Model: Database structure.
- Template: HTML rendering.
- View: Handles requests and sends data to templates.

5. Django Code Flow

1. Request comes in.
2. URL routing matches the view.
3. View processes data and renders template.
4. Response is sent back to the user.

6. Django Request-Response Cycle

1. Request → URL routing → View execution.
2. Template renders → Response is sent to the client.

7. Django App vs. Project

- Project: A collection of settings and configurations, can contain multiple apps.
- App: A modular component of the project, each handling specific functionality (e.g., a blog).

8. Django File Structure

- Project: manage.py, settings.py, urls.py

- App: models.py, views.py, admin.py, migrations/