**1. What is MVT in Django?**

Django is built on the **MVT (Model-View-Template)** design pattern.  
It is very similar to MVC (Model-View-Controller) but Django has its own way of handling the controller part internally.

**MVT stands for:**

1. **Model** – Manages the data (database).
2. **View** – Handles the business logic and communicates with the model.
3. **Template** – Deals with the presentation layer (HTML/CSS for users).

**2. How MVT Works in Django (Step-by-Step)**

**Request Flow:**

1. A user sends a **request** by entering a URL.
2. **URL Dispatcher** checks the urls.py and sends the request to the correct **View** function/class.
3. **View**:
   * Interacts with the **Model** (if data is needed).
   * Selects the appropriate **Template**.
   * Sends back a **Response (HTML)** to the user.

**3. Components Explained**

**A. Model (models.py)**

* It is a **Python class** that represents a database table.
* Each attribute is a field in the table.
* Django ORM (Object Relational Mapper) is used to interact with the database without writing raw SQL.

**Example:**

*# models.py*

**from** django**.**db **import** models

class **Student**(*models***.***Model*)**:**

    name **=** models**.**CharField(**max\_length=**100)

    roll\_no **=** models**.**IntegerField()

    city **=** models**.**CharField(**max\_length=**50)

**def** \_\_str\_\_(*self*)**:**

**return** *self***.**name

**B. View (views.py)**

* Contains the **logic** for handling requests.
* Fetches data from the Model and sends it to the Template.

**Example:**

*#b*

*# views.py*

**from** django**.**shortcuts **import** render

**from** **.**models **import** Student

**def** student\_list(**request**)**:**

    students **=** Student**.**objects**.**all()  *# Get all students*

**return** render(request**,** 'student\_list.html'**,** {'students'**:** students})

**C. Template (HTML files in templates folder)**

* Defines **how data will be displayed**.
* Uses Django Template Language (DTL) to insert variables dynamically.

**Example:**

*#c*

**<**!-- student\_list**.**html --**>**

**<**h2**>**Student List**</**h2**>**

**<**ul**>**

  {**%** **for** student **in** students **%**}

**<**li**>**{{ student**.**name }} **-** {{ student**.**city }}**</**li**>**

  {**%** endfor **%**}

**</**ul**>**

**4. Difference Between MVT and MVC**

* **MVC (Model-View-Controller)**: You write your own Controller.
* **MVT (Model-View-Template)**: Django has a built-in controller (URL Dispatcher).
  + **View in MVC = Template in MVT**
  + **Controller in MVC = View in MVT**

**5. Complete Flow Diagram (Simplified)**

* User -> URL -> urls**.**py -> View -> Model -> Template -> Response (HTML)
* urls.py = Routes request
* views.py = Handles request
* models.py = Data
* templates/ = Presentation

**6. Key Points**

* MVT separates **data, logic, and presentation** clearly.
* Django's **View** acts as a Controller + Business Logic.
* Templates use **dynamic variables** and tags.
* Database operations are abstracted using Django ORM.

**Example Mini Project Folder**

project**/**

    app\_name**/**

        models**.**py

        views**.**py

        urls**.**py

        templates**/**

            student\_list**.**html

