Practical-1

P-1: Introduction to MVC and MVT architecture. Comparative Study on MVC and MVT.

MVC (Model-View-Controller) and MVT (Model-View-Template) are both architectural patterns used in software development, particularly in web application frameworks. While they share some similarities, they have distinct differences in how they organize the components of an application.

1. MVC (Model-View-Controller):
- Model: Represents the data and business logic of the application. It handles data manipulation and communicates with the database or external data sources.
- View: Represents the user interface (UI) of the application. It displays the data from the Model and interacts with the user.
- Controller: Acts as an intermediary between the Model and the View. It processes user input, updates the Model, and renders the appropriate View.

In MVC, the Controller manages the flow of data between the Model and the View. When a user interacts with the View, the Controller processes the input, updates the Model, and then instructs the View to update accordingly. MVC provides a clear separation of concerns and makes it easier to maintain and modify individual components.

2. MVT (Model-View-Template):
- Model: Similar to MVC, it represents the data and business logic of the application.
- View: Represents the user interface, just like in MVC.
- Template: This is the key difference from MVC. The Template is a presentation layer that defines how the data from the Model is displayed in the View. It is responsible for rendering the HTML or other markup code.

In MVT, the Template plays a significant role in the architecture. When a user interacts with the View, the data is sent to the corresponding Template, which processes the data and renders the final output displayed to the user. The Template is responsible for the presentation logic, making it a mixture of both View and Controller responsibilities.

The main distinction between MVC and MVT is the role of the Template in MVT. While MVC has a clear separation of concerns between the Model, View, and Controller, MVT combines some aspects of the View and Controller into the Template. Django, a popular web framework, follows the MVT architecture.

In summary, MVC and MVT are both architectural patterns used in web development, but they differ in how they handle the presentation logic and the responsibilities of the components involved. MVC has a strict separation of concerns, while MVT combines aspects of the View and Controller into the Template.