MVC Design Pattern

The Model-View-Controller (MVC) design pattern is a widely used architectural approach for building user interfaces that separates the application logic (Model), data presentation (View), and user interaction handling (Controller).

In simple words we can say MVC is an architectural pattern that organizes an application's logic into distinct layers, each of which carries out a specific task.

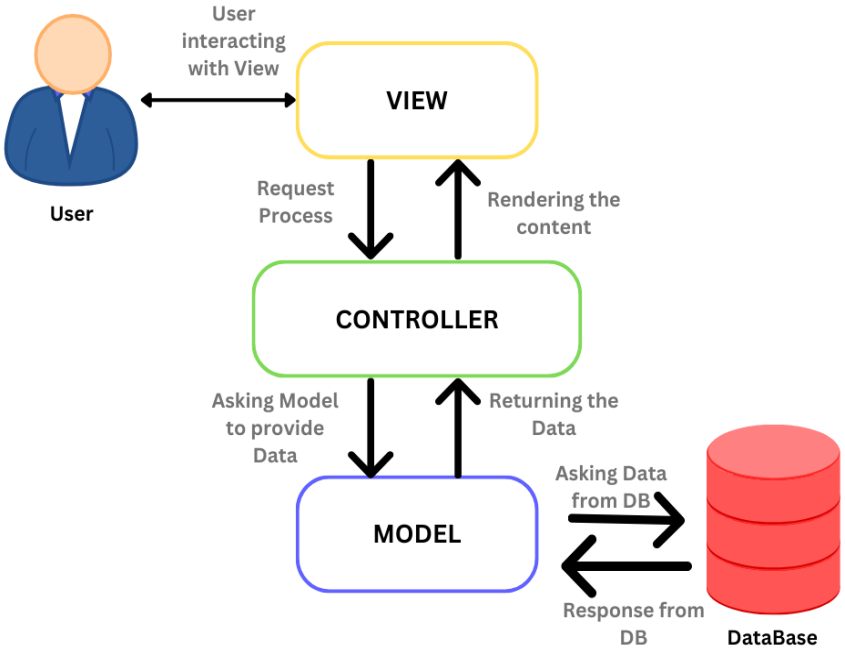
This separation promotes cleaner code, improved maintainability, and easier testing.

Components:

- **Model:** Represents the core data and business logic of the application. It encapsulates the data and provides methods to manipulate it.
- **View:** Responsible for presenting the data to the user in a specific format. It receives data from the Model and updates itself accordingly.
- **Controller:** Handles user interactions (like button clicks, form submissions). It communicates with both the Model and the View :
    1. Updates the Model based on user input.
    2. Retrieves data from the Model and instructs the View to update itself.

Benefits of MVC:

- Makes code modular and easier to understand.
- Changes to one component (model/view/controller) are less likely to impact others.
- Easier to test individual components and gives more flexibility.

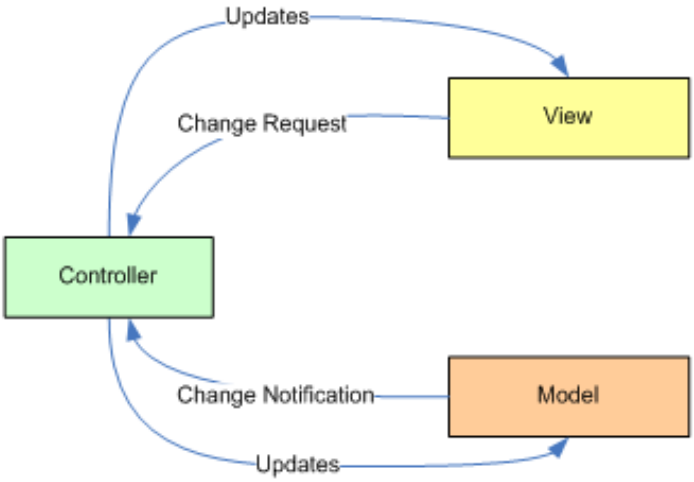


Types of MVC:

1. **Passive view MVC**
2. **Model view MVC**

Passive view MVC:

- The view is purely passive which means letting things happen in its way and simply displays data provided by controller.
- Advantages are simpler implementation and easier testing.
- While coming to Disadvantages this is less flexible and controller has more job to do.



Model View Presenter (MVP):

- Introduces a Presenter layer between the View and the Model.
- The Presenter receives data from the Model and prepares it for the View. The View notifies the Presenter about user interactions.
- Advantages are easier unit testing means we can check in-detail like individual components can be tested.
- Coming to disadvantages it is more complex structure and additional layers of abstraction.