**MVC (Model-View-Controller):**

**Model:** Represents the data and the business logic. It directly interacts with the database or other data sources.

**View:** Represents the UI and is responsible for rendering the data from the model. It typically doesn't contain business logic.

**Controller:** Acts as an intermediary between the Model and the View. It processes user input, updates the Model, and changes the View accordingly.

**Pros:**

* **Separation**: Clear separation between data, UI, and user interaction logic.
* **Simplicity**: Easy to understand and implement for small to medium-sized applications.

**Cons:**

* **Controller cause difficulty**: In large applications, Controllers can become complex and handle too much logic.
* **Tight Coupling**: Views and Controllers can become tightly coupled, making it harder to test and maintain.

**Usage in Flutter:**

* Flutter doesn’t have a built-in MVC framework, but you can implement it by creating different classes for Models, Views (Widgets), and Controllers (Classes that handle user input and interact with Models).