

## **We used MVC design pattern in our rails application**

Model-View-Controller (MVC) is a design pattern for structuring software systems. The MVC pattern separates application logic from the user interface. MVC stands for Model, View, and Controller, the three main components of the MVC web application architecture. Each of these components should operate as discrete units. The discrete nature of MVC components is crucial for app maintenance and testing.

We often hear that the best way to write clear and concise code in Rails is by following the “Fat Models and Skinny Controllers” approach, which refers to how the Model and Controller in an MVC architecture should ideally work together. Having a “skinny” Controller (which is common practice) means that all non-critical logic moves to the Model; this simplifies testing and maintenance. However, we suggest taking the “skinny” approach a step further: put both your Models and your Controllers on a diet. Often we focus on Models and Controllers, but what about the View? The View can also be “fat,” and in fact, we believe that the View also should go on a diet.

we faced a problem when we pass a parameters from database to view so we used MVC which was perfect choice to resolve it .

### **Advantages of MVC architecture:**

- Development of the application becomes fast.
- Easy for multiple developers to collaborate and work together.
- Easier to Update the application.
- Easier to Debug as we have multiple levels properly written in the application.

### **Disadvantages of MVC architecture:**

- It is hard to understand the MVC architecture.
- Must have strict rules on methods.