the best architectural pattern that fit our system is Layered Architecture. This architecture is based on building a layer on top of the previous layers that are organized hierarchically, with each layer is consisting of related functionalities. The highest outer layer would service the user interface, while the lowest layer provides core functionality. each layer can use only the layers above or below it directly.

Layered Architecture Diagram:

GUI

Presentation Layer

Application Layer

Model

Controller

Data Layer

Firebase Database

We have used Model View Controller (MVC) pattern in our system. We chose to implement MVC pattern because it is more suitable for what we are trying to implement. The MVC pattern splits code into one of three MVC components.

* **Model**: contains data and business logic. It interacts with the controller and has no knowledge of the interface.
* **View**: contains what is visible to the user (or allows the user to interact with elements on the screen).
* **Controller**: interacts with the view and the model by responding to user input and retrieving data the user requests. It also manages the application logic.

GUI

Controller

Model

Advantage of using this pattern:

* Separation of Concerns:
* Re-usability and Flexibility:
* High Cohesion and Low Coupling: