Our application uses a Model-View Controller (MVC) architecture. In this architecture pattern, the application is separated into three main components, which are a model that stores calendar events in a local database and schedules them appropriately, a view that displays the contents of the model in an understandable manner to the user, and a controller that reads user input in order to add, modify, or delete events, or change the view of the data. The frontend includes both view and controller components This architecture is effective for this application because it allows for multiple views of the events, which in this case includes weekly/monthly views of events, as well as displaying events part of a specific category. Another advantage of this architectural model is independence between different parts of the system, as If small changes need to be made to the database or the internal logic of the event scheduling (Model), then changing the view or input validation (View and Controller) is not needed.