



**CODING  
CLUB**

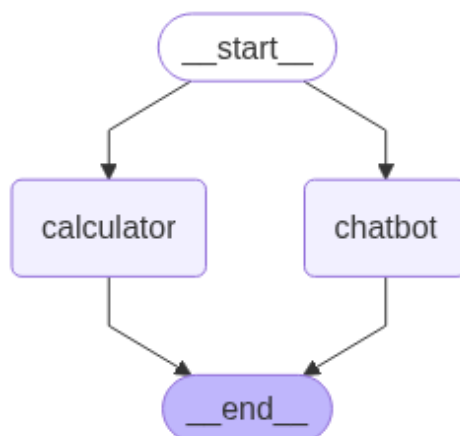
# **TASK 2 REPORT**

**PREPARED BY:  
RAM TEJ DUVVURI**

# Level - 1 (Core Activation)

1. I started off the task with working on node called chatbot which answer basic queries of user.
2. To start things off I had register google cloud project to access Gemini.
3. And I Build a node named chatbot which answer user basics question.
4. In this process I understood that answer given by chatbot is in dictionary format so i write a small function Query which take string as input and give to chatbot function and take answer as output.
5. And Next I worked on Calculator , I wrote a function which can do calculation with BODMAS rule in mind
- 6 . Then for to visualize graph i used code given in langgraph website.

## **Graph**



# Level - 2 ( Senses of the World )

1.To integrate Weather update into functionality I used openWeather API

2.Then I wrote a function which take city as string as return a string contains of temperature and weather of that city.

3.To integrate fashion recommendations into functionality I used one of the API of rapidapi .

4.Then I wrote a function which take command as string as return a string contains of recommendation and price of it .

# Level - 3 (Judgement and Memory)

1. At this point, I added a judgement element to Weathermind so it could understand user queries better.
2. First, when I tried adding judgement, things got a bit messy. I was trying to write a function called `determine_tool` that takes keywords from the user's question—like "Weather" or "Temperature"—and calls the `get_weather` node to answer. But it didn't work as expected, 'cause my `get_weather` node returns a string, but my function expects a dict. Also, I kinda messed up the logic for routing responses, so it wasn't handling things properly.
3. So I broke the problem into parts and came up with a solution. First, I wrote a function that detects the city in the sentence, then depending on the keywords, it sends this string to either the weather function, fashion function, or just the basic chatbot.
4. This time it worked! After that, I jumped onto the memory part, followed the LangGraph documentation, and it worked fine