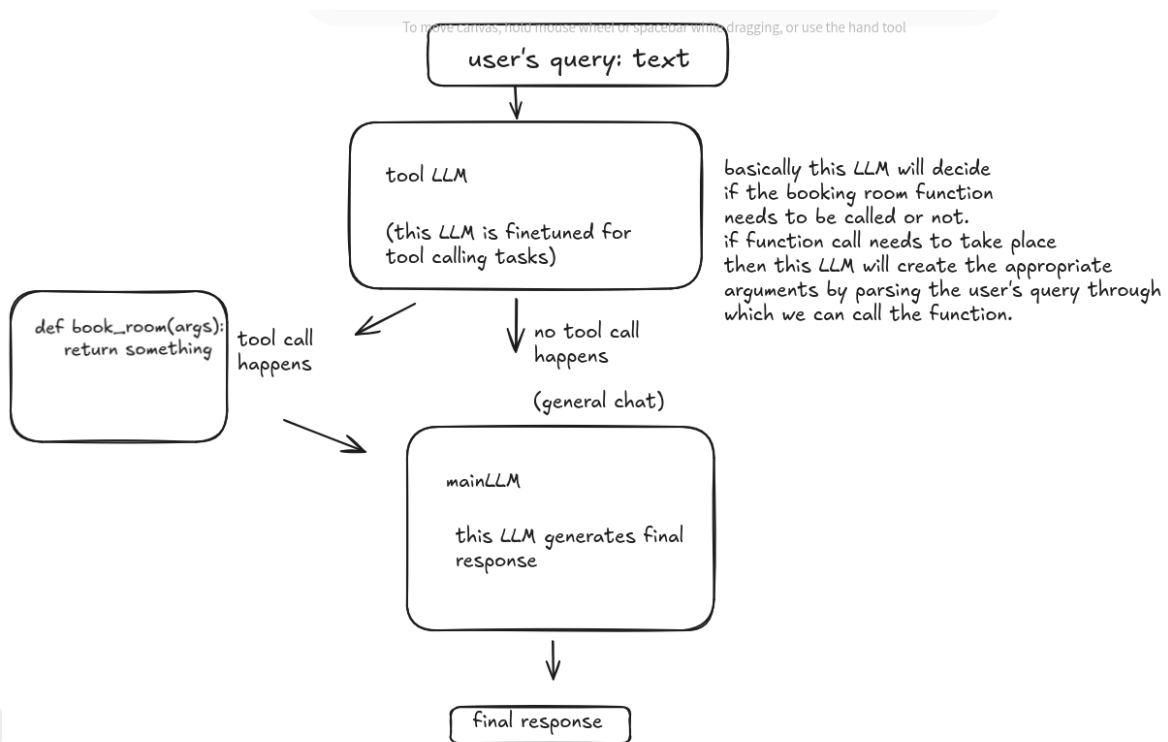# Assignment Report (Booking Agent):

Our objective is to create an LLM integrated with external APIs/functions which can fetch real time data/insights and based on that information and then our LLM can generate responses based on the information. LLMs with this type of flow are known as AI Agents.
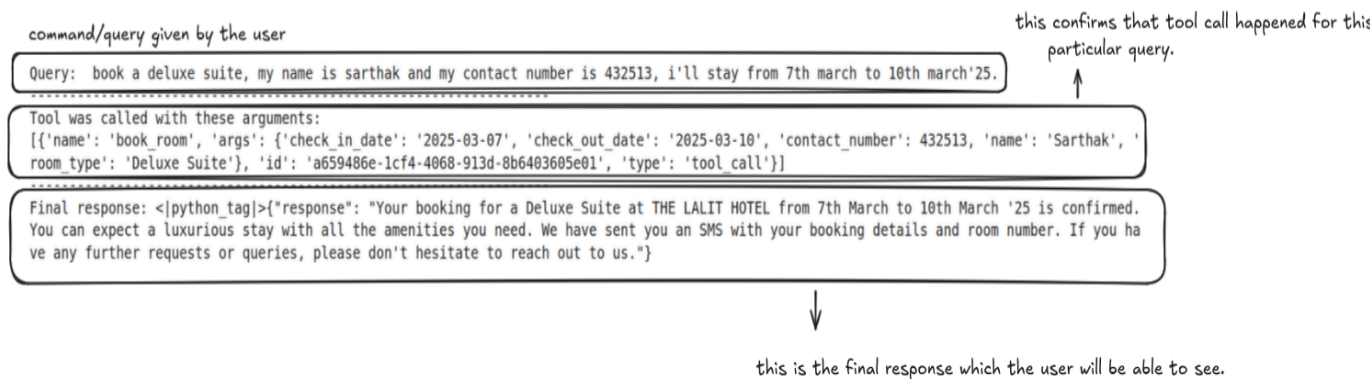
I've used ollama to get the model weights which can be used for inference. Apart from that I've used langchain to integrate our LLM (which is downloaded from ollama), with API/function call, which is commonly known as tool calling.

Also the system prompt given to these LLMs also play a key role in how the final response will look like. These models are prompt sensitive.
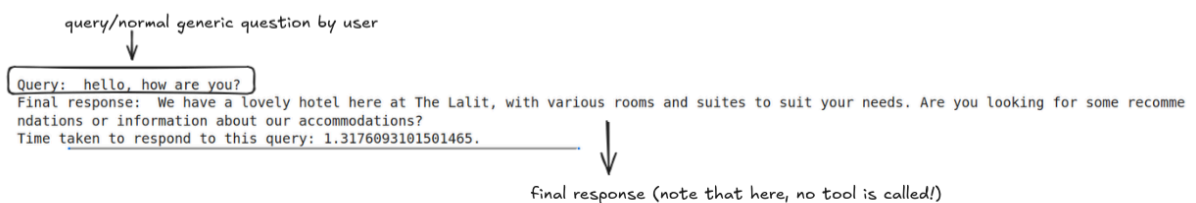
The whole structure of the code is like this:

Output if tool calling happens:

command/query given by the user

this confirms that tool call happened for this particular query.

```
Query:  book a deluxe suite, my name is sarthak and my contact number is 432513, i'll stay from 7th march to 10th march'25.
..............................................................
Tool was called with these arguments:
[{'name': 'book_room', 'args': {'check_in_date': '2025-03-07', 'check_out_date': '2025-03-10', 'contact_number': 432513, 'name': 'Sarthak', '
room_type': 'Deluxe Suite'}, 'id': 'a659486e-1cf4-4068-913d-8b6403605e01', 'type': 'tool_call'}]
..............................................................
Final response: <|python_tag|>{"response": "Your booking for a Deluxe Suite at THE LALIT HOTEL from 7th March to 10th March '25 is confirmed.
You can expect a luxurious stay with all the amenities you need. We have sent you an SMS with your booking details and room number. If you ha
ve any further requests or queries, please don't hesitate to reach out to us."}
```

this is the final response which the user will be able to see.

Output for general chat:

query/normal generic question by user

```
Query:  hello, how are you?
Final response:  We have a lovely hotel here at The Lalit, with various rooms and suites to suit your needs. Are you looking for some recomme
ndations or information about our accommodations?
Time taken to respond to this query: 1.3176093101501465.
```

final response (note that here, no tool is called!)

```
Query:  hello, how you can help me?
Final response:  Additionally, what type of accommodation are you looking for (e.g., single occupancy, couple, family)? This will help me bet
ter understand your needs and suggest the most suitable rooms for you.

We have various types of rooms available, including Deluxe Rooms, Premier Rooms, Executive Club Rooms, Executive Suites, Luxury Suites, Presi
dential Suite, and The Lalit Legacy Suite. Each room offers unique amenities and services, so please let me know if there's anything specific
you're looking for (e.g., extra bed, balcony view, etc.).
Time taken to respond to this query: 3.539876699447632.
```

———

Future scope of improvement:
- Making the LLM more robust by adding the chat history functionality i.e. LLMs now remembers all the previous conversations which have happened with a particular user.
- Currently the tool/function is a dummy function (code snippet attached below for reference), we can connect it with DB and can share the latest information with the user.

```python
# Function which will be called if tool calling happens.
@tool
def book_room(name, contact_number, check_in_date, check_out_date,
room_type)-> str:
        """Book a hotel's room on behalf of the user."""
        if(payment_done()):
        # update the DB that a room is booked.
        return f"Thanks {name}. Your {room_type} room has been booked
and details will be shared to your contact number:
{contact_number}."
        return "Payment was not successful. Room wasn't booked."
```

- Integration with website or hotel documents via RAG so that even staff can use it.