Documentation Learning Method

In documentation, the most basic version of the implementation will be given. You need to find out what are the different ways to implement that functionality. What are the possible parameters for the functionality? And what are the possible formats in which we can pass the values? What would be the format of the output?

Important: Any pre-built function in the LangGraph, allows us to generate the specific information in an output, and it is totally upon us to decide which part of that information is useful for us & which part we want to extract and use further. For that, It is important to understand the output format and what information exactly is being pushed in the output. So that we can decide about the suitable method to extract the relevant information from the output.
Important: Understand how the function is implemented in the documentation. Then try to check how it is possible to use it for your own case in your own way.

Pre-built functions in documentations: All pre-built functions are just the “functions stored in the backend” that takes input from us and produces output in a particular format for us.

Output of pre-built functions: We can fetch the information with some conditions of availability and put it in the requirement format which can be useful for us further.

Logic Building post-output: Most of the logic outside of the pre-built functions implementation is about fetching required information with conditions of availability and else case, storing it for further use, then breaking it down into multiple sections to deep dive inside it in order to find and show desired components as final output from overall information in the end.
How do we want it to give output?

Logic Building processing-output: Most of the logic within the processing of the pre-built functions is about “info fact check conditions e.g. after 1 run add value 1 in abc else 0, then if node wants to decide what to do it needs to check the value of abc and decide”. Mostly, it is about guiding the execution based on our conditions by using the inputs for the functions. We can pass inputs & design its execution based on conditions made upon those inputs.
How do we want it to run?

Usage of Async Functions:

For the functions where there can be a need to wait for the response from the external tool or LLM (I/O-bound tasks), if we are using the Async function for it. It means we are labeling it as a “coroutine” which means while it is calling its response, the program does not need to wait for the response to get generated in order to run the next independent task. We will “await” API/LLM call which means while response is awaited, our program can immediately start the next independent task - Meaning both running concurrently.

Like waiting for the response of LLM for user query (awaited) – AND next task of logging user query in DB is executed as it is independent of the awaited task.

1. It is necessary for the streaming of the output.

   Why: Because Without blocking the whole program while generating the response for I/O-bound query, we want the streaming function to work concurrently for sending the continuously generated chunks when available. So, to ensure while the response is getting generated the next task of streaming response works at the same time requires the functions to be Async.