

Design bots using Flowise

Last amended: 25th July, 2024

My folder: C:\Users\ashok\OneDrive\Documents\flowise

Table of Contents

A. Simple demo	2
B. Translation bot:	2
C. Chat with llama:	3
D. Simple Conversational Chain	4
E. Using Conversational Agents	4
F. Export import chat flows:	5
i) Export chatflow	5
ii) Load chat flow	5
G. Simple RAG with single text file	6
H. RAG with chroma store and single text file	7
I. Combining Multiple Chains (Prompt Chaining)	8
First LLM chain.....	8
Second LLM chain added to first	9
J. Flowise Using Hugging Face Models.....	10

A. Simple demo

See [this link](#) on YouTube.

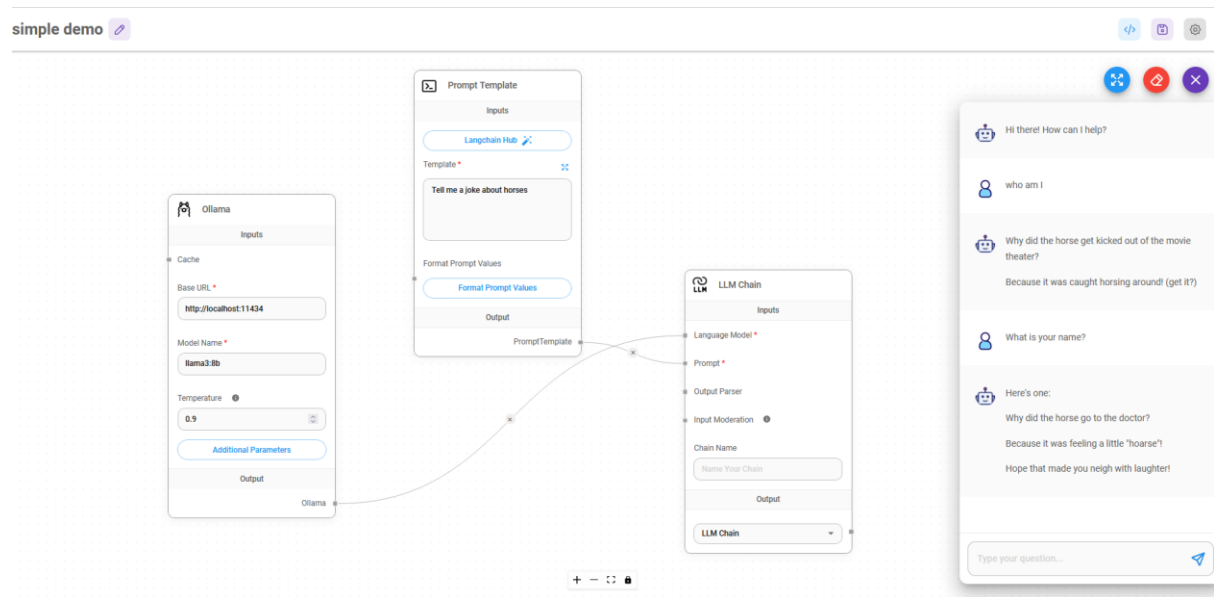


Figure 1: This bot will always answer your questions as a horse's joke. The only prompt is: **Tell me a joke about horses.**

B. Translation bot:

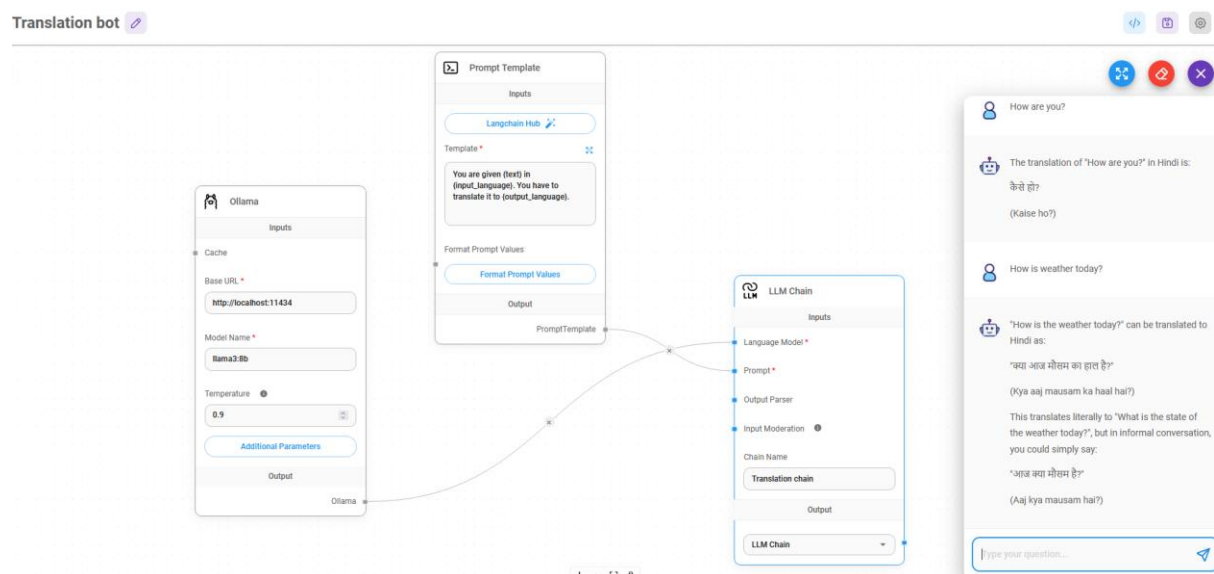


Figure 2: This bot translates all your questions into the desired language.

Prompt Template is:

You are given {text} in {input_language}. You have to translate it to {output_language}.

And formatted template is as follows. Note the **text** pertains to user's question asked in the chat-bot.

Format Prompt Values

```
{ 3 items
  text : "{{question}}"
  input_language : "English"
  output_language : "Hindi"
}
```

Figure 3: Translate question asked in English to Hindi: Note that 'question' is enclosed in two curly brackets.

C. Chat with llama:

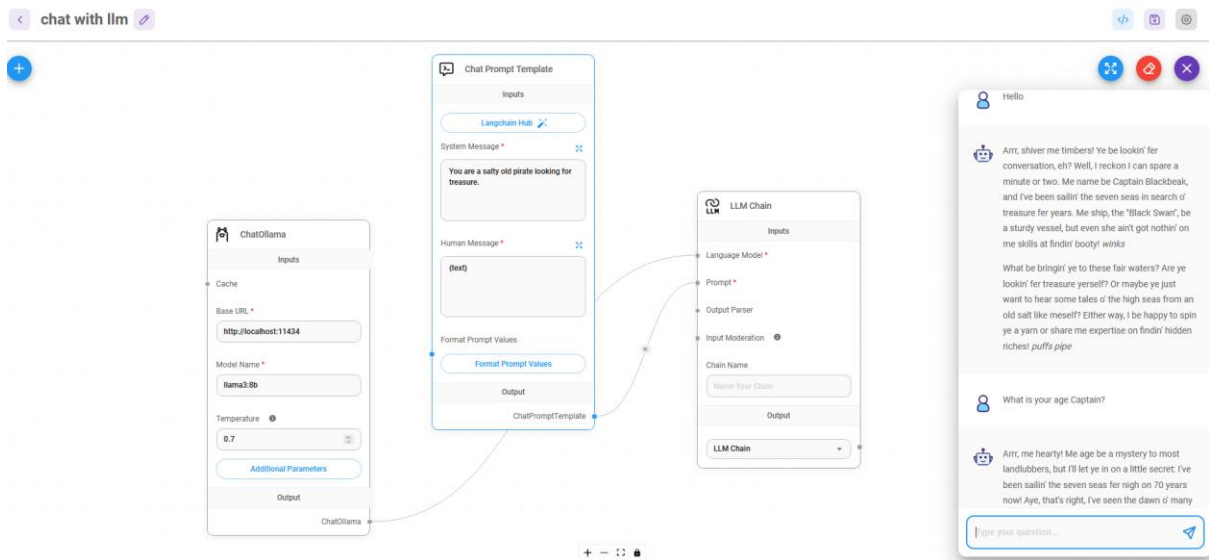
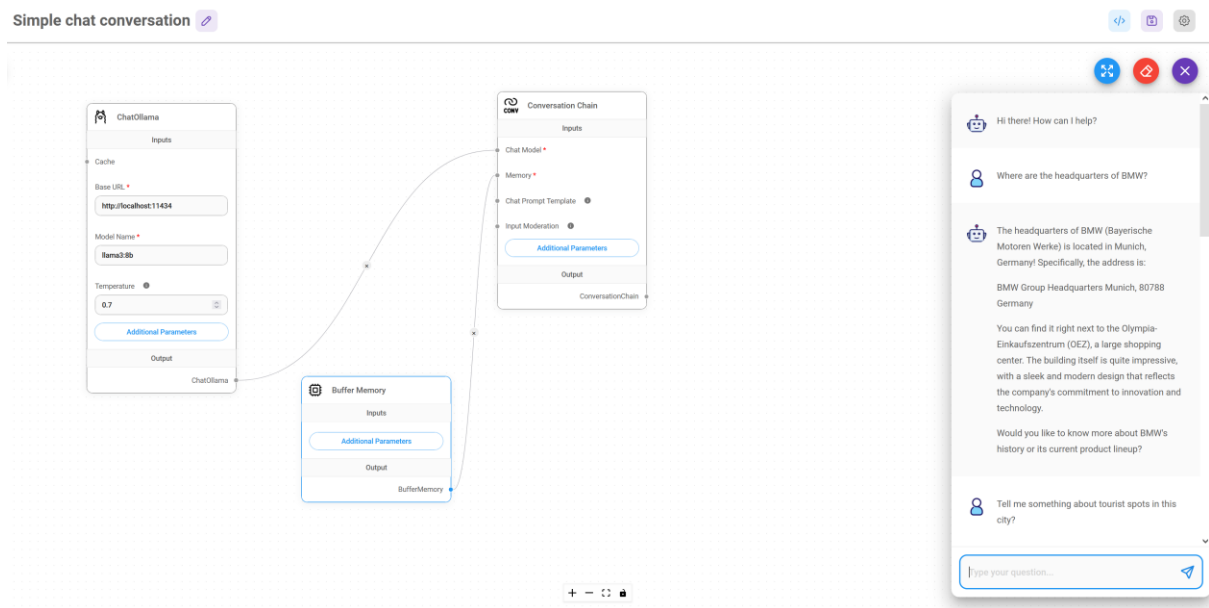


Figure 4: The 1st question is just **Hello** but the 11nd question asks more details about **Captain** referred to into the answer to **Hello**.

D. Simple Conversational Chain

Refer [YouTube video](#)



E. Using Conversational Agents

Refer [YouTube video](#)

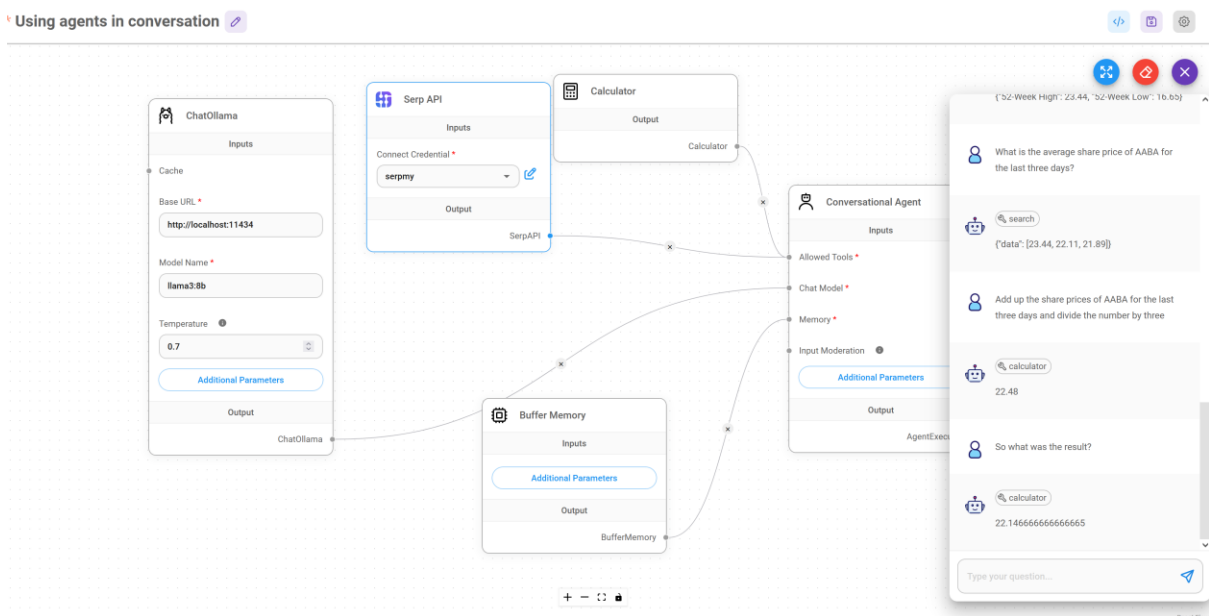


Figure 5: Agents can work even with ollama. SERP API key is a must. To calculate average, we have to tell the bot how to do it.

F. Export import chat flows:

i) Export chatflow

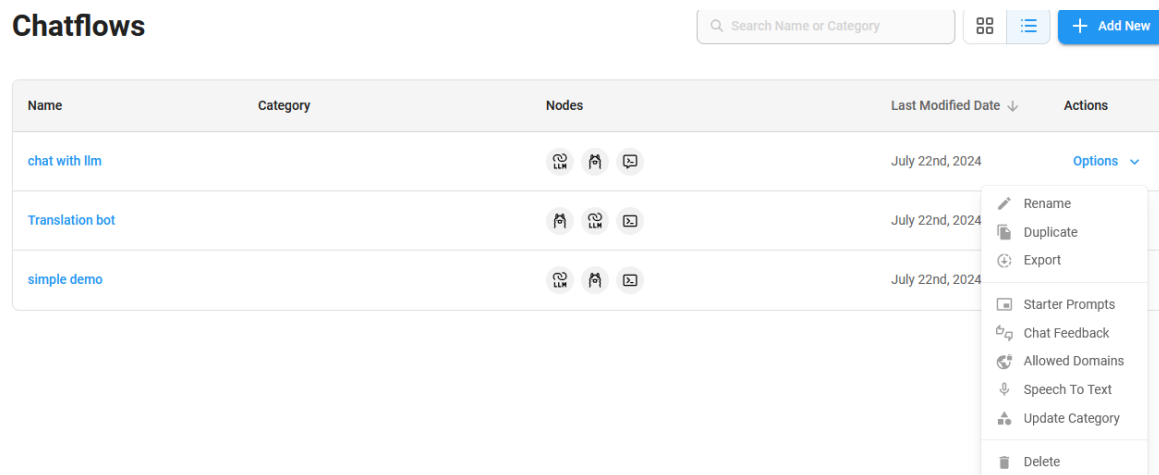


Figure 6: In the Chat flows window, click on the down arrow besides the **Options** to Export a chat flow as a json file.

ii) Load chat flow

To load a json file, first create a new (blank) chatflow by any name, say 'abc'. Save the blank chatflow. Click on Settings icon on top-right. And then click on Load chatflow to open and load the json file.

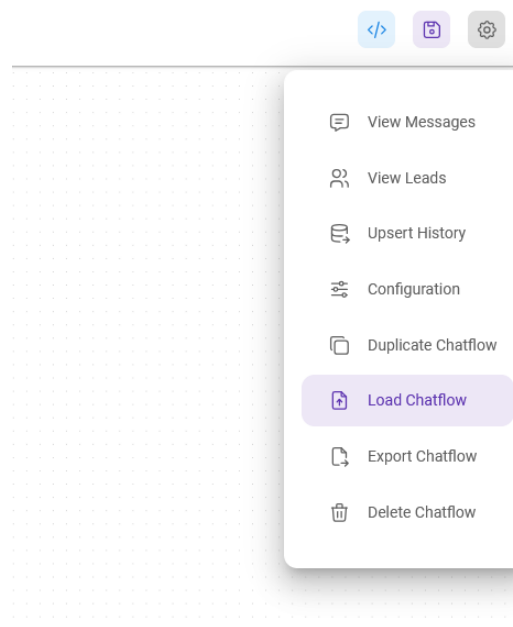


Figure 7: Click on Settings icon to import an exported chat flow (i.e. json file).

The following figure shows a chatflow loaded in Flowise:

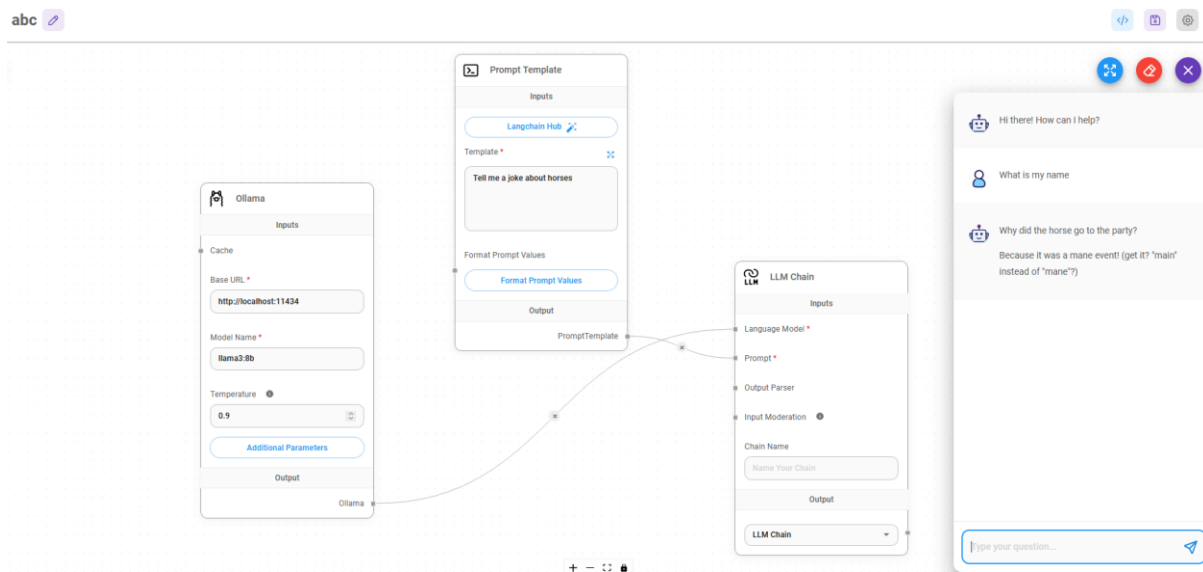


Figure 8: A chatflow loaded in a blank 'abc' chatflow canvas.

G. Simple RAG with single text file

Refer [Flowise tutorial #3](#)

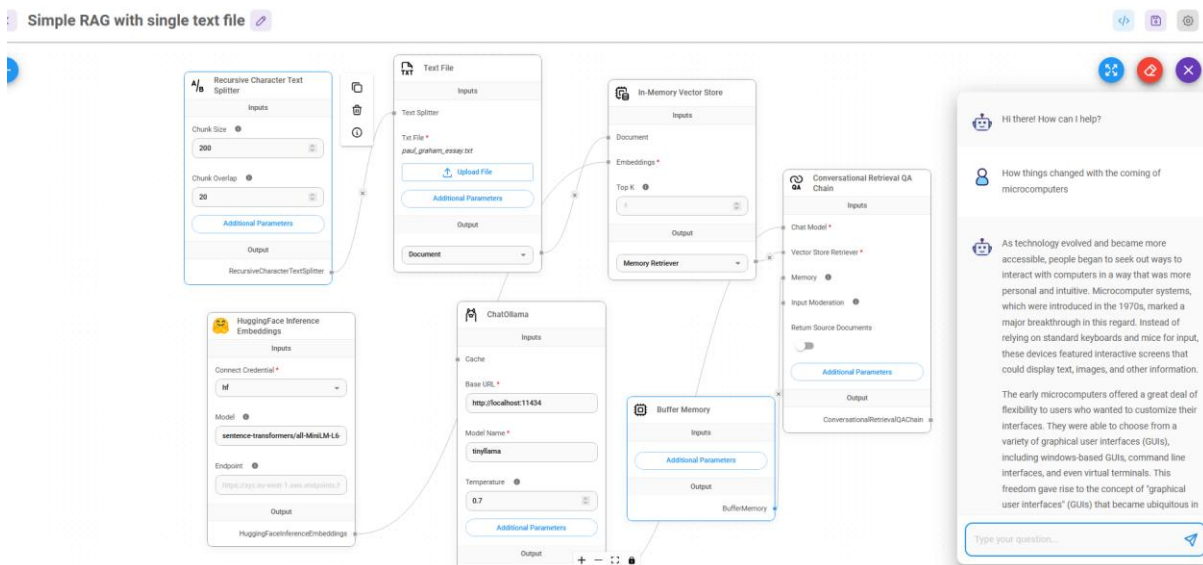


Figure 9: After connecting all flow-widgets and uploading of text file, first click on Upsert Vectors button and then start chatting.

Vector store in this RAG system will disappear as soon as Flowise is closed as the vectors are stored in buffer memory.

H. RAG with chroma store and single text file

RAG with chroma and single text file

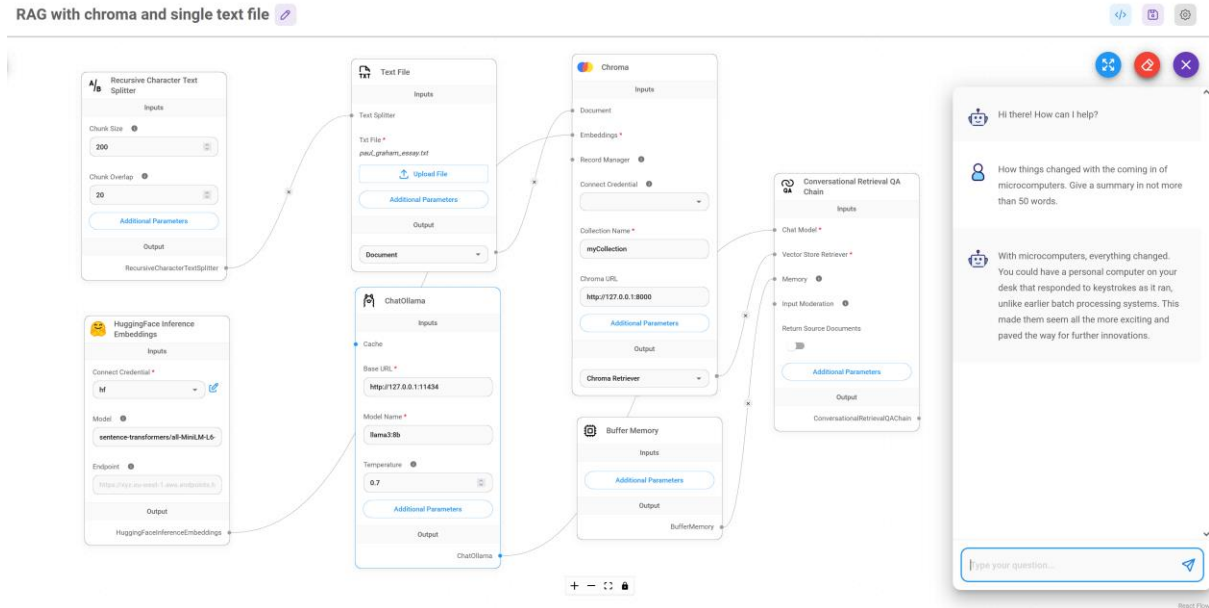


Figure 10: Vector store is replaced by a more durable chroma store. Chroma store retains its vectors even after Flowise is hut down.

RAG with chroma and single text file

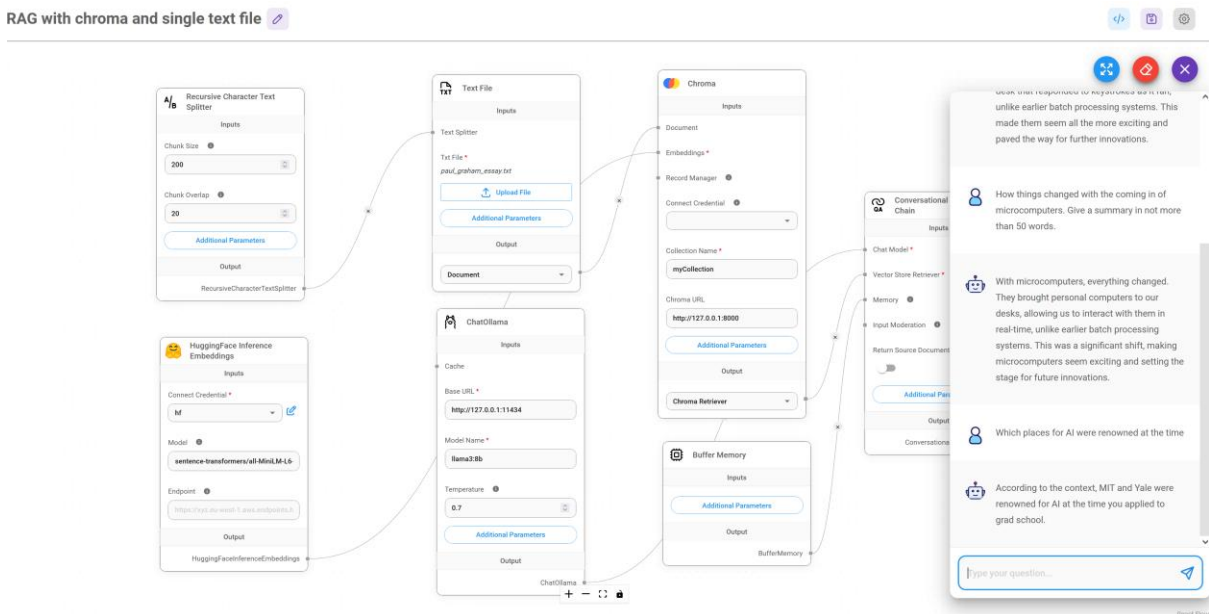


Figure 11: Flowise restarted. More questions asked and replies are given based upon the earlier storage.

I. Combining Multiple Chains (Prompt Chaining)

A. Refer [Flowise tutorial](#)

First LLM chain

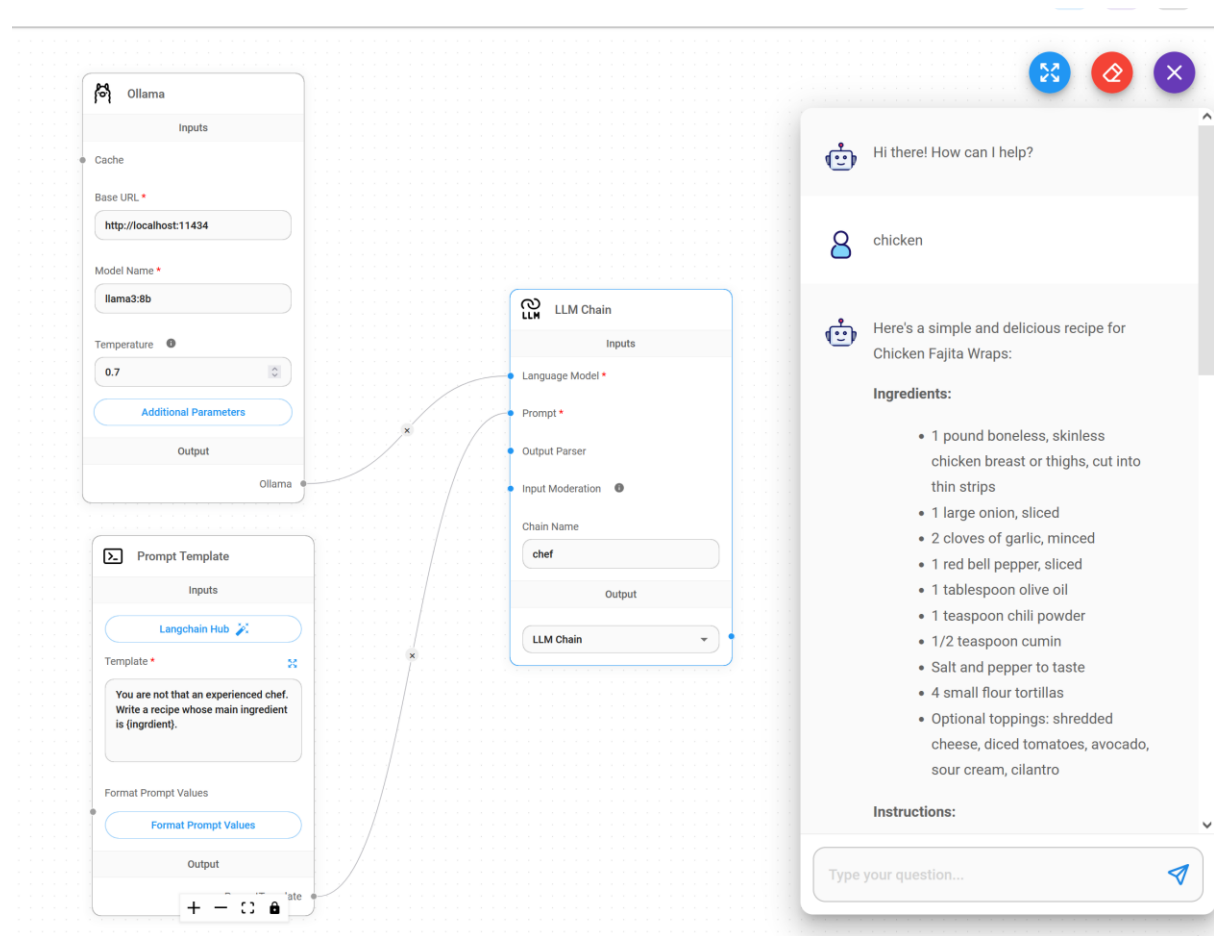


Figure 12: First LLM chain named as chef.

Second LLM chain added to first

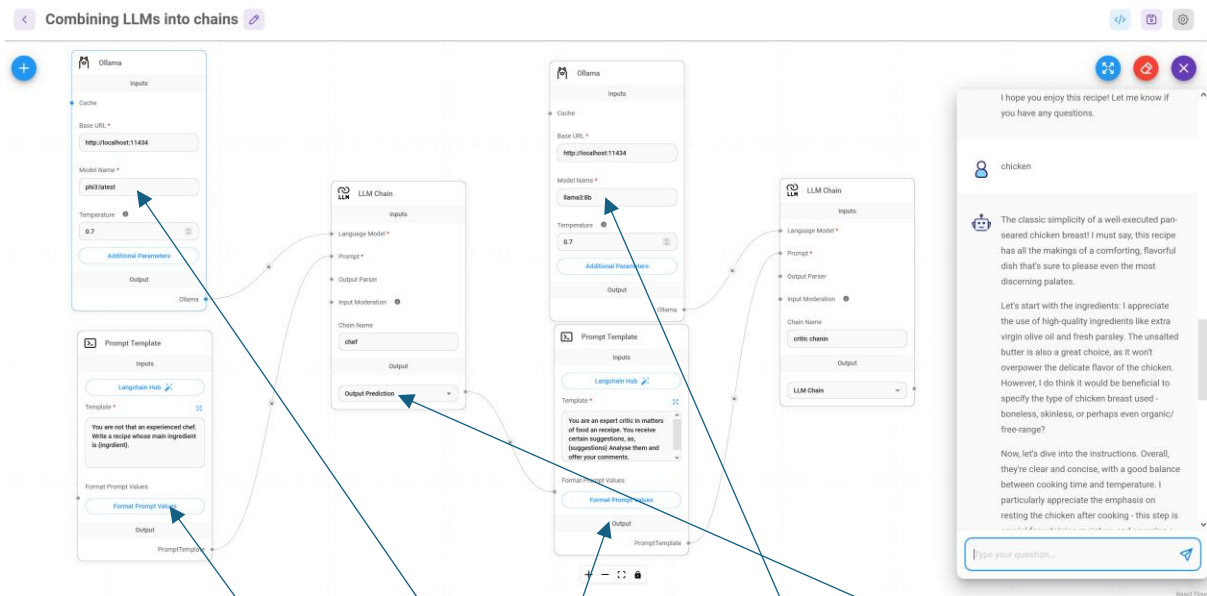


Figure 13: Note that the first LLM is using **phi3** and the critic language model is **llama3**. In the chatbox, the **llm** LLM chain does make minor suggestions to improve the quality. Note that the output of 1st LLM chain is now **Output Prediction**.

Here are the prompts used:

chef chain prompt: *You are not that an experienced chef. Write a recipe whose main ingredient is {ingredient}. Formatting of prompt values is as:*

```
Format Prompt Values
{ 1 item
  ingredient: "{{question}}"
}
```

Critic chain prompt: *You are an expert critic in matters of food an receipe. You receive certain suggestions, as, {suggestions} Analyse them and offer your comments. The formatting of prompt values is as:*

```
Format Prompt Values
{ 1 item
  suggestions : "{{llmChain_0.data.instance}}"
}
```

J. Flowise Using Hugging Face Models

Ref: [This link](#).

