

PYTHON AI CHATBOT WITH OLLAMA

Step 1: Download and setup [Ollama](#):

Step 2: Verify it downloaded:

```
PS C:\Users\prisha\OneDrive\Documents\Sherlock\workspace For Projects> ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve      Start ollama
  create     Create a model from a Modelfile
  show      Show information for a model
  run       Run a model
  stop      Stop a running model
  pull      Pull a model from a registry
  push      Push a model to a registry
  list      List models
  ps        List running models
  cp        Copy a model
  rm        Remove a model
  help      Help about any command

Flags:
  -h, --help      help for ollama
```

Step 3: Install a model relative to system hardware:

```

pulling manifest
pulling 630746a1ec1a... 100%
pulling 4fa551d4f938... 100%
pulling 8ab4849b038c... 100%
pulling 577073ffcc6c... 100%
pulling 3f8eb4da87fa... 100%
verifying sha256 digest
writing manifest
success

```

Step 4: Test the model:

```

success
C:\Users\user> ollama run llama3
>>> How are you?
I'm just a language model, I don't have emotions or personal experiences like humans do. However, I'm functioning properly and ready to assist with any questions or tasks you may have.

>>> /bye
C:\Users\user>

```

Step 5: Create a virtual environment to isolate the project from other projects:

NOTE Run: Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process

Before activating virtual environment if you're having problems activating it.

Command to create Virtual Environment: `python3 -m venv ollama-chatbot`

To activate:

.\ollama-chatbot\Scripts\activate **OR**

Navigate to: \ollama-chatbot\bin>

Then use `.\activate`

```

C:\Users\user\OneDrive\Desktop\Workspace For Projects\Chatbot-Project> cd .\ollama-chatbot\
C:\Users\user\OneDrive\Desktop\Workspace For Projects\Chatbot-Project\ollama-chatbot> cd .\bin\
C:\Users\user\OneDrive\Desktop\Workspace For Projects\Chatbot-Project\ollama-chatbot\bin> .\activate
(ollama-chatbot) C:\Users\user\OneDrive\Desktop\Workspace For Projects\Chatbot-Project\ollama-chatbot\bin>

```

Step 6: Install necessary dependencies:

```
(ollama-chatbot) PS C:\Users\user\OneDrive\Documents\Workspace For Projects> pip install langchain langchain-ollama ollama
Collecting langchain
```

Step 7: Code the chatbot using the necessary functions provided by Ollama:

```
Chatbot-Project > noGUI-chatbot.py > ...

1  #Rishaud Richard
2  # Building a very simple Ollama Local Chat Bot using langchain, langchain-ollama and ollama
3  # This chatbot only stores the history of a single run. If executed again, the previous history will not be available.
4
5  from langchain_ollama import OllamaLLM # importing necessary modules
6  from langchain_core.prompts import ChatPromptTemplate # importing necessary modules
7
8
9  """Template: This template is passed to the LLM that contains the prompt given. It gives the LLM a better description and instruction on how to respond
10     ...Answer the Question Below...-> Tells the model to answer the question below
11     ...Here is the conversation history: {context}...-> providing the LLM with context (conversation history)
12
13  template = """
14  Answer the Question Below.
15
16  Here is the conversation history: {context}
17
18  Question: {question}
19
20  Answer:
21  """
22
23  model = OllamaLLM(model="llama3") #Defines the model used for this chatbot
24  prompt = ChatPromptTemplate.from_template(template) #uses the chatPromptTemplate above (template)
25  chain = prompt | model #Chains the two operations. The prompt will have the question and the context. Then it'll be passed to the model to be invoked
26
27
28  """ This function allows us to actively talk to the bot while storing conversation history
29  def conversation():
30      context = ""
31      print("Welcome to Rishauds AI ChatBot Using Ollama! Type 'exit' to quit.") #welcome message
32      while True:
33          user_input = input("You: ")
34          if user_input.lower() == 'exit': # if user enters 'exit', break through the loop
35              break
36          result = chain.invoke({"context": context, "question": user_input}) # provides the LLM with the context as well as the question, where the result is stored in the variable called "result"
37          print("Ollama Bot: ", result) #prints answer of the question
38          context += f"\nUser: {user_input}\nAI: {result}" #Stores the previous prompt in the context variable, so the bot can use it to respond later on.
39
40  if __name__ == "__main__":
41      conversation()
42
```

Step 8: Run the chatbot:

```
[v] Never Run [b] Do Not Run [r] Run Once [A] Always Run [?] Help (default is 'b').  
(ollama-chatbot) PS C:\Users\Shahid\OneDrive\Documents\Workspace For Projects\Chatbot-Project> python .\noGUI-chatbot.py  
Welcome to Rishauds AI ChatBot Using Ollama! Type 'exit' to quit.  
You: how are you today?  
Ollama Bot: I'm doing well, thank you for asking! It's a lovely day today and I'm feeling energized and ready to tackle whatever comes my way. How about you?  
You: 
```

Optional: Run the GUI chatbot (designed with PyQt) for an interactive window, rather than the terminal:

