What will be the names of file. Lets start with,

1. Initialize LangChain with Local LLM
2. Build a Basic Chatbot with LangChain
3. Deploy via FastAPI

Give me a directory structure as well

So my project folder structure looks like this. Under each folder i have created files as you suggested. Like following,

1. D:\GenAI2025\chatbot\config\config.py
2. D:\GenAI2025\chatbot\src\api\server.py
3. D:\GenAI2025\chatbot\src\chatbot\chatbot.py
4. D:\GenAI2025\chatbot\src\models\local\_llm.py

Ive also created a virtual environment called .venv as per the screenshot that I pasted above.

Help me with,

1. One warning,

(.venv) PS D:\GenAI2025\chatbot> python D:\GenAI2025\chatbot\src\models\local\_llm.py

D:\GenAI2025\chatbot\.venv\Lib\site-packages\langchain\llms\\_\_init\_\_.py:549: LangChainDeprecationWarning: Importing LLMs from langchain is deprecated. Importing from langchain will no longer be supported as of langchain==0.2.0. Please import from langchain-community instead:

`from langchain\_community.llms import OpenAI`.

To install langchain-community run `pip install -U langchain-community`.

warnings.warn(

D:\GenAI2025\chatbot\.venv\Lib\site-packages\langchain\llms\\_\_init\_\_.py:549: LangChainDeprecationWarning: Importing LLMs from langchain is deprecated. Importing from langchain will no longer be supported as of langchain==0.2.0. Please import from langchain-community instead:

`from langchain\_community.llms import OpenAI`.

To install langchain-community run `pip install -U langchain-community`.

warnings.warn(

D:\GenAI2025\chatbot\src\models\local\_llm.py:5: LangChainDeprecationWarning: The class `OpenAI` was deprecated in LangChain 0.0.10 and will be removed in 1.0. An updated version of the class exists in the :class:`~langchain-openai package and should be used instead. To use it run `pip install -U :class:`~langchain-openai` and import as `from :class:`~langchain\_openai import OpenAI``.

return OpenAI(openai\_api\_base="http://localhost:5001/v1", openai\_api\_key="sk-fake-key")

Local LLM initialized successfully!

And one error

(.venv) PS D:\GenAI2025\chatbot> python D:\GenAI2025\chatbot\src\api\server.py

Traceback (most recent call last):

File "D:\GenAI2025\chatbot\src\api\server.py", line 3, in <module>

from src.chatbot.chatbot import chatbot

ModuleNotFoundError: No module named 'src'

=🡺>>>

Warning is gone.

Give me final code file for server.py

After running code in the virtual environment .venv, getting this error

(.venv) PS D:\GenAI2025\chatbot> python src/api/server.py

Traceback (most recent call last):

File "D:\GenAI2025\chatbot\src\api\server.py", line 5, in <module>

from src.chatbot.chatbot import chatbot # Import chatbot function

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

ModuleNotFoundError: No module named 'src'

Solution

i tried solution 1, and updated the code file for service.py and it worked.

(.venv) PS D:\GenAI2025\chatbot> python src/api/server.py

D:\GenAI2025\chatbot\src\chatbot\chatbot.py:6: LangChainDeprecationWarning: Please see the migration guide at: https://python.langchain.com/docs/versions/migrating\_memory/

memory = ConversationBufferMemory()

D:\GenAI2025\chatbot\src\chatbot\chatbot.py:8: LangChainDeprecationWarning: The class `ConversationChain` was deprecated in LangChain 0.2.7 and will be removed in 1.0. Use :meth:`~RunnableWithMessageHistory: https://python.langchain.com/v0.2/api\_reference/core/runnables/langchain\_core.runnables.history.RunnableWithMessageHistory.html` instead.

chatbot = ConversationChain(llm=llm, memory=memory)

INFO: Started server process [31576]

INFO: Waiting for application startup.

INFO: Application startup complete.

INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

What is the next step?

Facing this error in the API, using http://127.0.0.1:8000/docs

**Response body**

**Error: Internal Server Error**

**Response headers**

**content-length: 21 content-type: text/plain; charset=utf-8 date: Sun,16 Feb 2025 13:10:44 GMT server: uvicorn**

When checked through **test\_chat.py** inside D:\GenAI2025\chatbot

Got following error,

PS D:\GenAI2025\chatbot> & D:/GenAI2025/chatbot/.venv/Scripts/python.exe d:/GenAI2025/chatbot/test\_chat.py

Traceback (most recent call last):

File "D:\GenAI2025\chatbot\.venv\Lib\site-packages\requests\models.py", line 974, in json

return complexjson.loads(self.text, \*\*kwargs)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\Chittaranjan\AppData\Local\Programs\Python\Python312\Lib\json\\_\_init\_\_.py", line 346, in loads

return \_default\_decoder.decode(s)

^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\Chittaranjan\AppData\Local\Programs\Python\Python312\Lib\json\decoder.py", line 337, in decode

obj, end = self.raw\_decode(s, idx=\_w(s, 0).end())

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

File "C:\Users\Chittaranjan\AppData\Local\Programs\Python\Python312\Lib\json\decoder.py", line 355, in raw\_decode

raise JSONDecodeError("Expecting value", s, err.value) from None

json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

File "d:\GenAI2025\chatbot\test\_chat.py", line 7, in <module>

print(response.json()) # Should print chatbot's response

^^^^^^^^^^^^^^^

File "D:\GenAI2025\chatbot\.venv\Lib\site-packages\requests\models.py", line 978, in json

raise RequestsJSONDecodeError(e.msg, e.doc, e.pos)

requests.exceptions.JSONDecodeError: Expecting value: line 1 column 1 (char 0)

Here are the steps that are complete,

1. **Create a new folder:** my\_chatbot
2. Open my\_chatbot folder in VS Code
3. Created a new file named chatbot.py inside my\_chatbot project folder.
4. virtual environment (.venv) is activated
5. installed langchain requests
6. creates chatbot.py with your code
7. LM Studio reachable at <http://127.0.0.1:1234>

Then this error,

(.venv) PS D:\GenAI2025\my\_chatbot> python chatbot.py

Traceback (most recent call last):

File "D:\GenAI2025\my\_chatbot\chatbot.py", line 31, in <module>

llm = LMStudioLLM(model\_url=lm\_studio\_url)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

TypeError: Can't instantiate abstract class LMStudioLLM without an implementation for abstract method '\_llm\_type'

✅ Local LLM (Llama 3.2 3B) initialized successfully! message received, but

When I run test\_lm\_studio.py I get following error,

PS D:\GenAI2025\chatbot> & D:/GenAI2025/chatbot/.venv/Scripts/python.exe d:/GenAI2025/chatbot/test\_lm\_studio.py

Status Code: 200

Response: {'error': 'Unexpected endpoint or method. (POST /completions)'}

Old server.py code

from fastapi import FastAPI, Request

from pydantic import BaseModel

import requests  # Required to make API calls

# LM Studio API Endpoint

LM\_STUDIO\_URL = "http://127.0.0.1:1234/v1/chat/completions"

app = FastAPI()

class ChatRequest(BaseModel):

    message: str

@app.post("/chat")

async def chat\_endpoint(request: ChatRequest):

    try:

        # Prepare request payload

        payload = {

            "model": "llama-3.2-3b-instruct-uncensored",  # Adjust model name if needed

            "messages": [{"role": "system", "content": "You are a helpful assistant."},

                         {"role": "user", "content": request.message}],

            "temperature": 0.7,

            "max\_tokens": 256

        }

        # Send request to LM Studio

        response = requests.post(LM\_STUDIO\_URL, json=payload)

        response\_data = response.json()

        # Extract the chatbot response

        if "choices" in response\_data and response\_data["choices"]:

            chat\_response = response\_data["choices"][0]["message"]["content"]

        else:

            chat\_response = "Error: No valid response from LLM."

        return {"response": chat\_response}

    except Exception as e:

        return {"error": str(e)}

# ✅ Handle unexpected favicon requests to avoid warnings

@app.get("/favicon.ico")

async def favicon():

    return {}

# Root endpoint

@app.get("/")

async def root():

    return {"message": "Welcome to Local LLM Chatbot API!"}

if \_\_name\_\_ == "\_\_main\_\_":

    import uvicorn

    uvicorn.run(app, host="0.0.0.0", port=8000)

Feedback, when i run the powershell command i get following,

curl -X POST "http://127.0.0.1:8000/chat" -H "Content-Type: application/json" -d "{\"message\": \"Hello!\"}"

and under which folder shall I create test\_api.py file?

I want all of the following,

1. Improve the chatbot response handling?
2. Integrate with LM Studio properly?
3. Deploy the API somewhere?

And also to create a UI where user can interact and easily communicated with LLM.

Old chat.py code – working

import streamlit as st

import requests

# ✅ API Endpoint

API\_URL = "http://127.0.0.1:8000/chat"

# ✅ Streamlit UI

st.set\_page\_config(page\_title="Chatbot UI", layout="wide")

st.title("🗨️ Local LLM Chatbot")

st.write("Chat with your locally hosted Llama model.")

# ✅ Chat history storage

if "messages" not in st.session\_state:

    st.session\_state.messages = []

# ✅ Display chat history

for message in st.session\_state.messages:

    with st.chat\_message(message["role"]):

        st.markdown(message["content"])

# ✅ User Input

user\_input = st.chat\_input("Type your message...")

if user\_input:

    # ✅ Add user message to chat history

    st.session\_state.messages.append({"role": "user", "content": user\_input})

    # ✅ Send request to FastAPI backend

    response = requests.post(API\_URL, json={"message": user\_input})

    if response.status\_code == 200:

        reply = response.json()["response"]

    else:

        reply = "⚠️ Error: Could not reach API."

    # ✅ Add assistant message to chat history

    st.session\_state.messages.append({"role": "assistant", "content": reply})

    # ✅ Display chatbot response

    with st.chat\_message("assistant"):

        st.markdown(reply)

another working chat.py

import streamlit as st

import requests

import time  # For simulating AI typing

# ✅ API Endpoint

API\_URL = "http://127.0.0.1:8000/chat"

# ✅ Streamlit Page Config

st.set\_page\_config(page\_title="Chatbot UI", layout="wide")

# 🎨 \*\*Header with Logo\*\*

st.markdown(

    """

    <div style="text-align: center;">

        <h1>🤖 Local AI Chatbot</h1>

        <p style="color: gray;">Powered by Llama 3.2 3B on LM Studio</p>

    </div>

    """,

    unsafe\_allow\_html=True,

)

# ✅ Sidebar settings

with st.sidebar:

    st.title("⚙️ Settings")

    temperature = st.slider("Creativity Level", 0.0, 1.0, 0.7, 0.1)

    max\_tokens = st.slider("Max Response Length", 50, 500, 200, 50)

    if st.button("🗑️ Clear Chat"):

        st.session\_state.messages = []  # Clear chat history

        st.rerun()  # ✅ Use st.rerun() instead of experimental\_rerun()

# ✅ Chat history storage

if "messages" not in st.session\_state:

    st.session\_state.messages = []

# ✅ Chat layout using columns

col1, col2 = st.columns([3, 1])  # Left: Chat, Right: Sidebar

with col1:

    # ✅ Display chat history with avatars

    for message in st.session\_state.messages:

        with st.chat\_message(message["role"], avatar="🧑‍💻" if message["role"] == "user" else "🤖"):

            st.markdown(message["content"])

    # ✅ User Input

    user\_input = st.chat\_input("Type your message...")

    if user\_input:

        # ✅ Add user message to chat history

        st.session\_state.messages.append({"role": "user", "content": user\_input})

        # ✅ Display user message immediately

        with st.chat\_message("user", avatar="🧑‍💻"):

            st.markdown(user\_input)

        # ✅ Send request to FastAPI backend

        response = requests.post(API\_URL, json={"message": user\_input})

        if response.status\_code == 200:

            reply = response.json()["response"]

        else:

            reply = "⚠️ Error: Could not reach API."

        # ✅ Typing Effect

        with st.chat\_message("assistant", avatar="🤖"):

            with st.spinner("Typing..."):

                time.sleep(1.5)  # Simulate AI thinking

                st.markdown(f"📝 \*\*AI:\*\* {reply}")

        # ✅ Add assistant message to chat history

        st.session\_state.messages.append({"role": "assistant", "content": reply})

Can you summarize all the steps that we done so I can copy and paste this for my future use? Give details in following areas to start,

1. Business case
2. Our solution approach
3. Errors that we faced
4. More improvement areas in future

Go in details of each step. Explain why that step was done. Don’t give me code for this summary as I already have it in the files. I want detailed summary to store it and possibly teach others on how to do this, but I need this summary with commentary on each step with,

1. What is the significance of this step? And why it was important to do this?

2. How this will impact other steps? Put all the steps in sequential manner, put all the error that we encountered and how we overcame those errors.

More requests,

1. I want to put this project on GitHub so tell me step by step on how shall I do this so my friends also can see this.
2. I want to add text to readme.txt on the github. Give me text which i can copy and paste to this file. This text must describe the project in apt detail and anyone who is reading this should know what they are reading. Also tell me how can i format this text in right way so the headings, paragraphs and text is aligned properly. i want you to provide me right text content, image content on the workflow and how this ai resume screener work with proper flowchart