

Zero to Chatbot

Build & Deploy a GenAI App in 2 Hours

Live Coding Workshop

The "Zero-Boilerplate" Stack

The Ingredients

- **Python 3.9+:** The logic.
- **Streamlit:** The UI (No HTML/CSS!).
- **Groq:** The Brain (Llama 3.1).

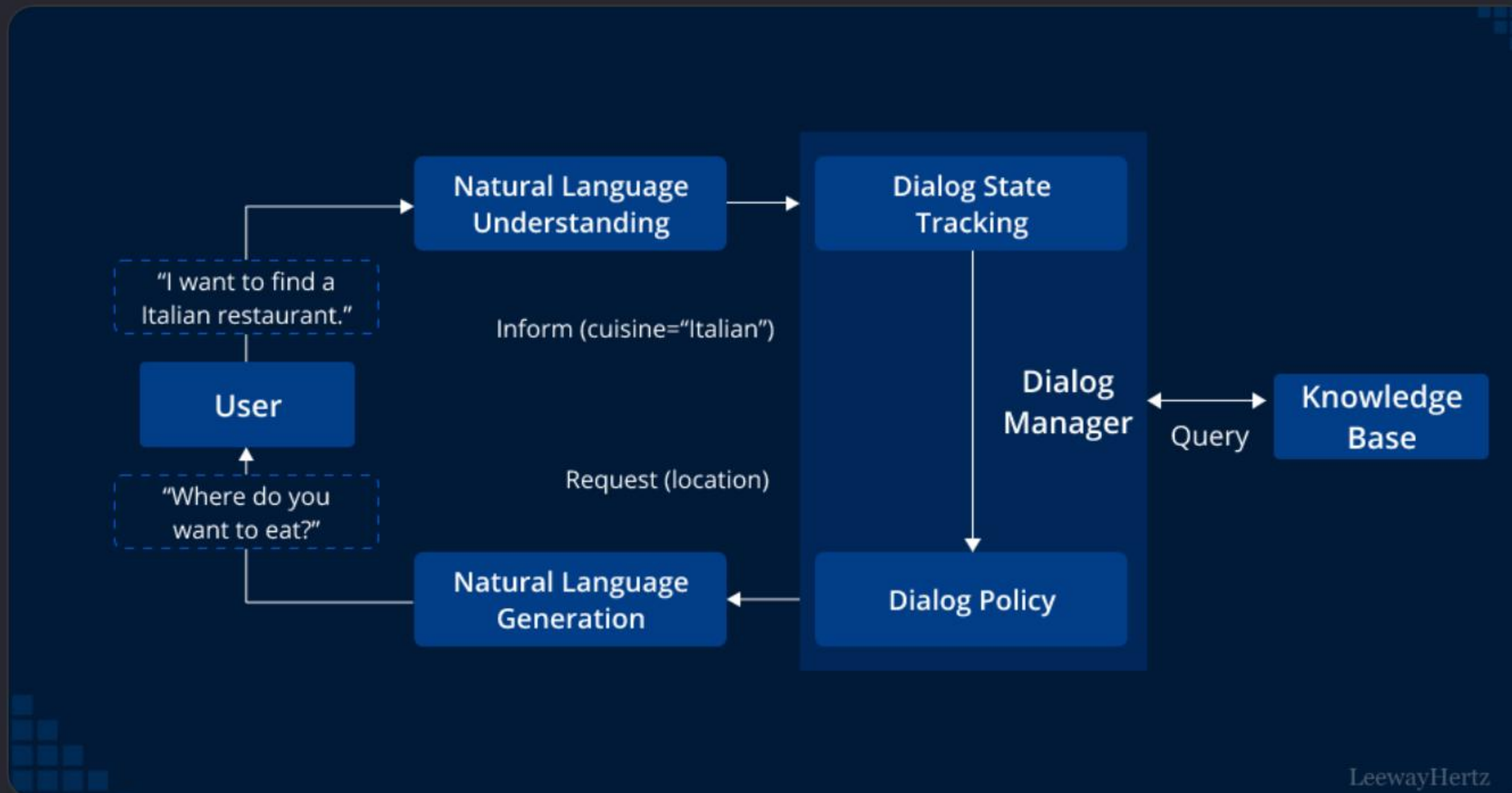
Critical Step:

Create a file named `requirements.txt` now.
Deployment fails without it.

```
# requirements.txt

streamlit
groq
python-dotenv
```


How It Works: The "Rerun" Loop



The Streamlit Flow

1. User types message & hits Enter.
2. **Script reruns from top to bottom.**
3. We pull history from `session_state`.
4. We append new message to history.
5. We send history to AI & render reply.

Block 1: Setup & Secrets

```
import os
import streamlit as st
from groq import Groq
from dotenv import load_dotenv

# 1. Load Keys (Local Only)
load_dotenv()

# 2. Init Client
# Finds GROQ_API_KEY automatically
client = Groq()

# 3. Page Config
st.set_page_config(
    page_title="My Bot",
    page_icon="🤖"
)
```

Why this code?

- `load_dotenv()` allows us to use a `.env` file on our laptop.
- **Cloud Note:** On Streamlit Cloud, we won't have a `.env` file; we will use the "Secrets" dashboard instead.
- `client = Groq()` initializes our connection to the AI.

Block 2: The Brain (Memory)

```
# 4. Initialize History
if "messages" not in st.session_state:
    st.session_state.messages = []
```

Session State



Because Streamlit reruns the script constantly, normal variables get wiped. `st.session_state` is a persistent dictionary that survives the reruns.

Block 3: The Display Loop

```
# 5. Display History
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])
```

Re-drawing the UI

Every time the app updates, we must loop through our saved messages list and draw them on the screen again.

- `st.chat_message()` creates the bubble.
- `role` is either "user" or "assistant".

Block 4: Handling User Input

```
# 6. Capture Input
if prompt := st.chat_input("Hi?"):

    # Save to memory
    st.session_state.messages.append(
        {"role": "user", "content": prompt}
    )

    # Show immediately
    with st.chat_message("user"):
        st.write(prompt)
```

The Trigger

`st.chat_input` pauses the script at the bottom.

Only when the user hits Enter does the code inside the `if` block run. This is where we capture their text and save it.

Block 5: Calling Intelligence

```
# 7. Generate Response
try:
    completion = client.chat.completions.create(
        model="llama-3.1-8b-instant",
        messages=st.session_state.messages,
        temperature=1,
        max_tokens=1024,
    )

    response = completion.choices[0].message.content
```

Connecting to Llama 3

- We send the **entire** messages list so the AI has context.
- **Important:** Use a valid model ID like `llama-3.1-8b-instant`. Old IDs cause crashes.
- We wrap it in `try/except` for safety.

Block 5: Closing the Loop

```
# 8. Display & Save
with st.chat_message("assistant"):
    st.markdown(response)

st.session_state.messages.append(
    {"role": "assistant", "content": response}
)

except Exception as e:
    st.error(f"Error: {e}")
```

Finalizing

Once we get the text back from Groq:

1. We draw the assistant's bubble.
2. We append the answer to `session_state` so it remembers it next time.

Deployment: Going Live

1. GitHub

Upload `app.py` and `requirements.txt` to a **Public** repository.

2. Streamlit Cloud

Log in to share.streamlit.io and select your repo.

3. The Secret Sauce

Before deploying, click **Advanced Settings** → **Secrets**.



Format for Secrets:

```
GROQ_API_KEY = "gsk_..."
```


Emergency Troubleshooting

- **400 Error?** Check your Model ID. Is it valid?
- **ModuleNotFound?** You forgot `requirements.txt`.
- **API Key Error?** You forgot to set Secrets in the cloud.
- **App runs but no reply?** Check indentation under `if` prompt:.



Image Sources



<https://static.vecteezy.com/system/resources/thumbnails/013/134/366/small/aesthetic-black-abstract-background-vector.jpg>

Source: www.vecteezy.com



<https://d3lkc3n5th01x7.cloudfront.net/wp-content/uploads/2023/07/20012845/Transactional-chatbot.png>

Source: www.leewayhertz.com



https://img.freepik.com/premium-photo/science-technologies-artificial-intelligence-concept-with-side-view-digital-human-brain-with-polygonal-glowing-lines-around-dark-background-3d-rendering_670147-11616.jpg

Source: www.freepik.com



https://img.lovepik.com/photo/48050/2340.jpg_wh860.jpg

Source: lovepik.com



https://static.vecteezy.com/system/resources/previews/001/986/251/non_2x/404-error-neon-sign-bright-red-warning-sign-on-dark-brick-wall-background-error-page-not-found-concept-neon-signboard-vector.jpg

Source: www.vecteezy.com