PROJECT REPORT

Groq-Powered Chatbot Using Streamlit and LLaMA 3.1

Introduction

Artificial Intelligence–based chatbots are widely used for automating conversations, answering queries, and providing intelligent assistance. This project demonstrates the development of a chatbot using:

- Streamlit – for frontend interface
- Groq API – for fast AI inference
- LLaMA 3.1 8B Instant – Large Language Model for generating responses

The chatbot allows users to interact in real-time and receive AI-generated responses through a simple web interface.

Objective of the Project

The main objectives of this project are:

- To build a real-time conversational AI chatbot
- To integrate Groq's LLM API with a web application
- To understand session state management in Streamlit
- To demonstrate AI-powered text generation

Technologies Used

Technology	Purpose
Python	Programming Language
Streamlit	Frontend Web Framework
Groq API	AI Model Inference
LLaMA 3.1 8B Instant	Language Model
VS Code	Development Environment

System Architecture

User → Streamlit UI → Groq API → LLaMA Model → Response → Display in UI

Working Flow:

1. **User enters a message.**
2. **Message is stored in session state.**
3. **Request is sent to Groq API.**
4. **LLaMA model processes input.**
5. **Response is returned.**
6. **Response is displayed in chat format.**

Code Explanation

```python
import streamlit as st
from groq import Groq

#initialize Groq client
client=Groq(api_key="YOUR_GROO_API_KEY")

st.set_page_config(page_title="Groq Chatbot")
st.title("Groq-Powered Chatbot")

#chat history

if "messages" not in st.session_state:
    st.session_state.messages=[]

#display messages
for msg in st.session_state.messages:
    with st.chat_message(msg["role"]):
        st.markdown(msg["content"])

#user input

user_input=st.chat_input("Ask me anything...")

if user_input:
    st.session_state.messages.append(
        {"role":"user","content":user_input}
    )
    with st.chat_message("user"):
        st.markdown(user_input)
```

```
response=client.chat.completions.create(
    model="llama-3.1-8b-instant",
    messages=st.session_state.messages
)
bot_reply=response.choices[0].message.content

st.session_state.messages.append(
    {"role":"assistant","content":bot_reply}
)
with st.chat_message("assistant"):
    st.markdown(bot_reply)
```

Step 1: Import Libraries

```
import streamlit as st
from groq import Groq
```

- **streamlit creates the web interface.**
- **Groq connects to Groq API.**

Step 2: Initialize Groq Client

```
client=Groq(api_key="YOUR_GROQ_API_KEY")
```

This connects your application to Groq's servers using your API key.

Step 3: Page Configuration

```
st.set_page_config(page_title="Groq Chatbot")
st.title("Groq-Powered Chatbot")
```

Sets page title and heading.

Step 4: Session State Initialization

```
if "messages" not in st.session_state:
    st.session_state.messages=[]
```

Stores chat history so conversation continues without resetting.

Step 5: Display Previous Messages

```
for msg in st.session_state.messages:
    with st.chat_message(msg["role"]):
        st.markdown(msg["content"])
```

Displays chat messages in conversation format.

Step 6: User Input

```
user_input=st.chat_input("Ask me anything...")
```

Creates input box for user.

Step 7: API Call to Groq

```
response=client.chat.completions.create(
    model="llama-3.1-8b-instant",
    messages=st.session_state.messages
)
```

- **Sends conversation to LLaMA 3.1 model.**
- **Receives AI-generated response.**

Step 8: Display Bot Response

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

Extracts and displays response.

Groq-Powered Chatbot

Ask me anything...

Groq-Powered Chatbot

what is deeplearning

Deep Learning: An Overview

Deep learning is a subset of machine learning that is inspired by the structure and function of the human brain. It is a type of artificial intelligence (AI) that enables computers to learn and improve on their own by analyzing large amounts of data, especially unstructured data such as images, videos, and audio files.

Key Characteristics of Deep Learning:

1. **Hierarchical Architecture**: Deep learning models are composed of multiple layers, each performing a different transformation on the input data. This hierarchical structure allows

Ask me anything...

Features of the Chatbot

- ✅ Real-time conversation
- ✅ Maintains chat history
- ✅ Uses powerful LLaMA model
- ✅ Simple and interactive UI
- ✅ Easy to deploy

Advantages

- Fast inference using Groq
- Easy integration
- Lightweight application
- Beginner-friendly implementation

Limitations

- Requires API key
- Needs internet connection
- Dependent on Groq service availability
- Limited customization without fine-tuning

Applications

- AI Virtual Assistant
- Educational Tutor Bot
- Customer Support Chatbot
- FAQ Automation
- Mini AI Project for Academics

Future Enhancements

- **Add voice input/output**
- **Add authentication system**
- **Store chat history in database**
- **Deploy on cloud (AWS / Render / Streamlit Cloud)**
- **Add model selection option**

Conclusion

This project successfully demonstrates how to build a real-time AI chatbot using Streamlit and Groq API. By integrating a powerful large language model like LLaMA 3.1, the system provides intelligent and context-aware responses. The project is suitable for beginners learning Generative AI, API integration, and web application development.