

■ Groq-Bot: An AI ChatBot - Project Documentation

This document provides full documentation for the **Groq-Bot AI ChatBot** project. It is an open-source Streamlit application that integrates with Groq's Python SDK to provide a conversational chatbot interface with streaming responses.

■ Key Features

- Built with Streamlit for an interactive and responsive UI.
- Integration with Groq's Python SDK for LLM chat completion.
- Streaming responses for real-time chat experience.
- Session state management to maintain conversation history.
- Sidebar controls for selecting model, temperature, and clearing chat.
- Customizable via ``.streamlit/secrets.toml`` for secure API key management.

■ Project Structure

- ``app.py`` → Main Streamlit application script. - ``.streamlit/secrets.toml`` → Stores API key securely (not committed to GitHub). - ``.gitignore`` → Ensures secrets and cache files are not pushed to repo. - ``requirements.txt`` → Python dependencies for deployment.

■■ Code Walkthrough

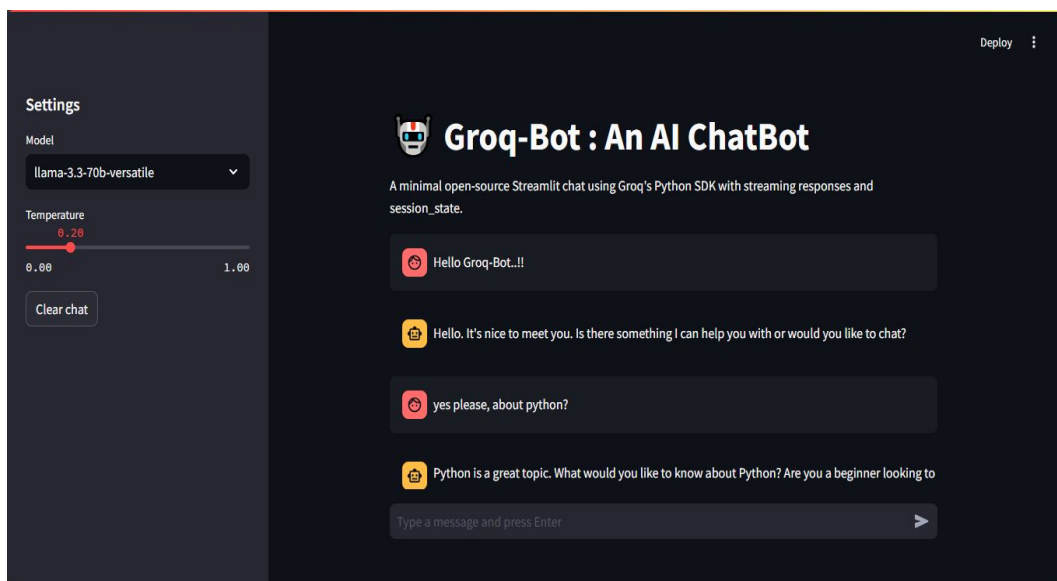
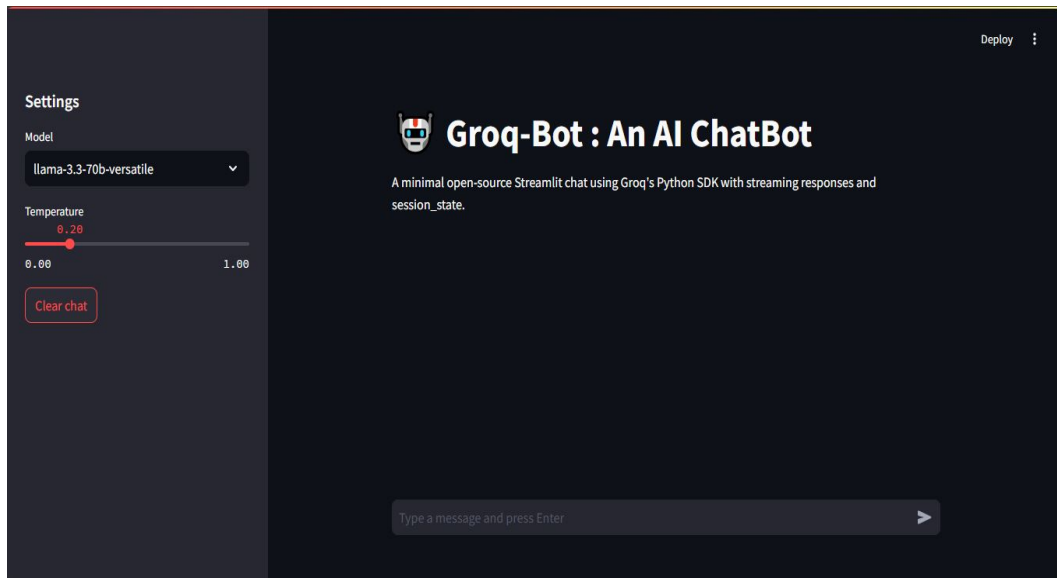
1. **get_client()** - Retrieves the `GROQ_API_KEY` from Streamlit secrets or environment variables. - Initializes and returns a `Groq` client instance. - Displays an error and stops the app if no API key is found. 2. **stream_response()** - Handles streaming responses from Groq API. - Iterates over incoming deltas from the model and yields incremental assistant text. - Supports robust parsing of streaming chunks, appending partial responses to final output. 3. **UI and Session State** - Title and description are displayed with `st.title()` and `st.write()`. - Sidebar provides settings: - Select Groq model (`llama-3.3-70b-versatile`, `mistral-saba-24b`, `gemma-7b`). - Adjust response temperature. - Clear chat history button. - Session state (`st.session_state`) tracks: - `messages` → list of all system, user, and assistant messages sent to the API. - `history` → stores chat history for UI display. - User input is captured with `st.chat_input()` and rendered in chat format using `st.chat_message()`. - Responses are streamed and updated live using a placeholder that refreshes with partial results.

■ Deployment Instructions

1. **Local Setup** - Clone the repository and install dependencies: `pip install -r requirements.txt`. - Create ``.streamlit/secrets.toml`` and add your Groq API key: `toml GROQ_API_KEY = "your_api_key_here"```. - Run app locally: `streamlit run app.py`. 2. **Deployment on Streamlit Cloud** - Push code to GitHub (excluding secrets). - On [\[share.streamlit.io\]\(https://share.streamlit.io\)](https://share.streamlit.io), create an app linked to the repo. - Add API key in the app's **Secrets Manager** under `GROQ_API_KEY`. - Deploy and share your live chatbot app.

■ ■ UI Screenshots (Reference)

The following screenshots illustrate the chatbot interface and its functionality.



Settings

Model

llama-3.3-70b-versatile

Temperature

0.28

0.001.00

Clear chat

Deploy

Python is a great topic. What would you like to know about Python? Are you a beginner looking to learn the basics, or do you have a specific question about a particular library or concept?

Some topics we could discuss include:

- Basics of Python syntax and data types
- Control structures (if/else, for loops, etc.)
- Functions and modules
- Object-Oriented Programming (OOP) concepts
- Popular libraries like NumPy, Pandas, and Matplotlib
- Web development with Django or Flask
- Data science and machine learning with scikit-learn and TensorFlow

Let me know what's on your mind, and I'll do my best to help!

Data science and machine learning with scikit-learn and TensorFlow

Type a message and press Enter

Settings

Model

llama-3.3-70b-versatile

Temperature

0.28

0.001.00

Clear chat

Deploy

Data science and machine learning are fascinating fields, and scikit-learn and TensorFlow are two of the most popular libraries used in these areas.

Scikit-learn

Scikit-learn is a Python library that provides a wide range of algorithms for machine learning tasks, including:

- Supervised learning: regression, classification, and support vector machines
- Unsupervised learning: clustering, dimensionality reduction, and anomaly detection
- Model selection: techniques for selecting the best model and hyperparameters

Some popular algorithms and tools in scikit-learn include:

- Linear Regression: `LinearRegression` and `Ridge` for simple and regularized regression
- Decision Trees: `DecisionTreeClassifier` and `DecisionTreeRegressor` for classification and regression
- Random Forests: `RandomForestClassifier` and `RandomForestRegressor` for ensemble learning

Type a message and press Enter

Settings

Model

llama-3.3-70b-versatile

Temperature

0.28

0.001.00

Clear chat

Deploy

- Support Vector Machines: `SVC` and `SVM` for classification and regression
- K-Means Clustering: `KMeans` for unsupervised clustering

TensorFlow

TensorFlow is an open-source machine learning library developed by Google. It's particularly well-suited for:

- Deep learning: neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs)
- Large-scale computations: distributed training and deployment of models

Some popular applications of TensorFlow include:

- Image classification: using CNNs to classify images
- Natural Language Processing: using RNNs and CNNs for text classification, sentiment analysis, and language modeling
- Speech recognition: using RNNs and CNNs for speech-to-text systems

Key differences between scikit-learn and TensorFlow

Type a message and press Enter

