**Llama 2 Chatbot Streamlit App Documentation**

**Introduction**

Welcome to the Llama 2 Chatbot Streamlit App documentation. This app allows users to interact with the Llama 2 Chatbot using the Replicate API within a Streamlit interface. Users can engage in conversations by providing prompts and receiving responses from the chatbot. This documentation provides a detailed guide on how to set up, run, and utilize the app effectively.

**Table of Contents**

Introduction

Getting Started

Prerequisites

Installation

Obtaining Replicate API Token

Usage

Running the App

Chat Interface

Clearing Chat History

Code Overview

App Setup

Generating Llama 2 Responses

Additional Resources

Llama 2 Models

**1. Introduction**

The Llama 2 Chatbot Streamlit App is an interactive interface that enables users to have conversations with the Llama 2 Chatbot using the Replicate API. By providing prompts, users can receive responses from the chatbot, facilitating engaging and dynamic interactions. This documentation will guide you through the setup, usage, and code structure of the app.

**2. Getting Started**

Prerequisites

Before using the Llama 2 Chatbot Streamlit App, ensure that you have the following prerequisites:

Python

streamlit library

replicate library

Installation

To install the required libraries, open your terminal and execute the following commands:

bash

pip install streamlit replicate

Obtaining Replicate API Token

To authenticate with the Replicate API, you need an API token. Here's how you can obtain it:

Sign up for an account on Replicate.

Access your API token from your account settings.

**3. Usage**

Running the App

To run the Llama 2 Chatbot Streamlit App, navigate to the directory containing your app.py file and execute the following command:

bash

streamlit run app.py

The app will launch in your default web browser.

Chat Interface

The app provides a user-friendly chat interface that allows you to have conversations with the Llama 2 Chatbot. Here's how the chat interface works:

Input prompts in the text box provided to initiate conversations.

Messages are displayed in a chat format, alternating between user and assistant roles.

The assistant's responses are generated using the Llama 2 Chatbot through the Replicate API.

Clearing Chat History

If you want to start a new conversation, you can clear the chat history by clicking the "Clear Chat History" button located in the sidebar.

**4. Code Overview**

In this section, we'll provide an overview of the key components of the code used to build the Llama 2 Chatbot Streamlit App.

App Setup

The app's setup includes the following functionalities:

User authentication using the Replicate API token.

Display of chat messages using Streamlit's chat component.

Ability to clear chat history.

Here's the code responsible for the app setup:

python

import streamlit as st

import replicate

import os

# App title

st.set\_page\_config(page\_title="🦙💬 Llama 2 Chatbot")

This code snippet imports necessary libraries, sets the app's title, and prepares for user authentication.

The next step is to authenticate the user using the Replicate API token. This is accomplished using Streamlit's sidebar and secrets feature:

python

# Replicate Credentials

with st.sidebar:

st.title('🦙💬 Llama 2 Chatbot')

if 'REPLICATE\_API\_TOKEN' in st.secrets:

st.success('API key already provided!', icon='✅')

replicate\_api = st.secrets['REPLICATE\_API\_TOKEN']

else:

replicate\_api = st.text\_input('Enter Replicate API token:', type='password')

if not (replicate\_api.startswith('r8\_') and len(replicate\_api) == 40):

st.warning('Please enter your credentials!', icon='⚠️')

else:

st.success('Proceed to entering your prompt message!', icon='👉')

st.markdown('📖 Learn how to build this app in this [blog](https://blog.streamlit.io/how-to-build-a-llama-2-chatbot/)!')

os.environ['REPLICATE\_API\_TOKEN'] = replicate\_api

Generating Llama 2 Responses

The generate\_llama2\_response function is responsible for generating responses from the Llama 2 Chatbot using the Replicate API. It leverages the chat history to create a dialogue context for generating coherent responses:

python

def generate\_llama2\_response(prompt\_input):

string\_dialogue = "You are a helpful assistant. You do not respond as 'User' or pretend to be 'User'. You only respond once as 'Assistant'."

for dict\_message in st.session\_state.messages:

if dict\_message["role"] == "user":

string\_dialogue += "User: " + dict\_message["content"] + "\n\n"

else:

string\_dialogue += "Assistant: " + dict\_message["content"] + "\n\n"

output = replicate.run('a16z-infra/llama13b-v2-chat:df7690f1994d94e96ad9d568eac121aecf50684a0b0963b25a41cc40061269e5',

input={"prompt": f"{string\_dialogue} {prompt\_input} Assistant: ",

"temperature":0.1, "top\_p":0.9, "max\_length":512, "repetition\_penalty":1})

return output

This function constructs the dialogue context, interacts with the Llama 2 Chatbot using the Replicate API, and returns the generated response.

**5. Additional Resources**

Llama 2 Models

Explore different Llama 2 models for chatbot interactions:

Llama2-7B

Llama2-13B

Llama2-70B

