Streamlit App Deployment Guide

This guide helps beginners create a Streamlit app in Google Colab, link it to GitHub, and deploy it on Streamlit Cloud. Each step is explained simply.

1. Preparing Your Project in Google Colab

Google Colab is a free online environment for Python coding.

Step 1: Generate Your Gemini API Key from Google AI Studio

- **Action:**
  1. Go to https://aistudio.google.com/ and sign in.
  2. Find and click Get API key (usually under "Develop" -> "API key").
  3. Create a new API key.
  4. Copy the generated key.
- **Reason:** This key lets your app use Google's Gemini models.

Step 2: Create credentials.json in Google Colab

- **Action:** In a new Colab code cell, run this. **Replace "YOUR_GEMINI_API_KEY" with your actual key.**

```
import json

api_key_data = {"GOOGLE_API_KEY": "YOUR_GEMINI_API_KEY"}

with open("credentials.json", "w") as f:
    json.dump(api_key_data, f)

print("credentials.json created!")
```

- **Reason:** This file stores your API key separately from your main code. Your Streamlit app will read it from here.

Step 3: Install Streamlit and Google Generative AI Library

- **Action:** In a new Colab code cell, run:

```
!pip install streamlit google-generativeai
```

- **Reason:** These are the Python libraries needed for your web app and to talk to Gemini.

Step 4: Ask Gemini to Generate Your Streamlit App Code

- **Action:** Go to https://aistudio.google.com/. In the chat, use this prompt:
  Give me streamlit code in python for a multi-turn application that answers questions using gemini related to healthcare only. The API key should be read from a file named 'credentials.json' where the key is named 'GOOGLE_API_KEY'.

- **Reason:** You're using Gemini to generate the core code for your app.

Step 5: Copy Gemini-Generated Code to Google Colab

- **Action:**
  1. Copy the Python code provided by Gemini.
  2. Paste it into a new code cell in your Colab notebook.
  3. **Important Checks:**
     - **API Key Reading:** Ensure the code reads the API key from credentials.json using GOOGLE_API_KEY and the path /content/credentials.json.
     - **Gemini Model:** Verify that the code specifies gemini-2.0-flash when initializing the Gemini model (e.g., genai.GenerativeModel('gemini-2.0-flash')).
- **Reason:** This is your app's main code. Verifying these details ensures it works as expected with your API key and the correct Gemini model.

Step 6: Create requirements.txt

- **Action:** In a new Colab code cell, run:
  %%writefile requirements.txt
  streamlit
  google-generativeai

- **Reason:** This file tells Streamlit Cloud which libraries to install for your app.

Step 7: Create Dockerfile (Optional)

- **Action:** In a new Colab code cell, run:
  %%writefile Dockerfile
  FROM python:3.9-slim-buster

  WORKDIR /app

  COPY requirements.txt .
  RUN pip install -r requirements.txt

  COPY . .

```
EXPOSE 8501

ENTRYPOINT ["streamlit", "run", "app.py", "--server.port=8501",
"--server.address=0.0.0.0"]
```

- **Reason:** A Dockerfile provides precise instructions for setting up your app's environment, ensuring it runs consistently.

Step 8: Save Your Notebook

- **Action:** Click File > Save a copy in Drive. You can rename it.
- **Reason:** This saves all your project files in Google Drive.

2. Linking Google Colab to GitHub

GitHub is a platform for storing and managing code. Streamlit Cloud deploys apps directly from GitHub.

Step 1: Save a Copy to GitHub

- **Action:** In your Colab notebook, click File > Save a copy in GitHub.
- **Reason:** This pushes your Colab notebook and associated files to GitHub.

Step 2: Authenticate GitHub

- **Action:** Authorize Google Colab to access your GitHub account. Create a GitHub account if you don't have one.
- **Reason:** Colab needs permission to interact with your GitHub repositories.

Step 3: Choose Repository and Commit

- **Action:**
  1. Select or create a GitHub repository.
  2. Enter a short commit message (e.g., "Initial healthcare Gemini app").
  3. Click OK.
- **Reason:** This organizes your project code on GitHub. **Note:** Saving credentials.json to a *public* GitHub repository makes your API key visible. For production, use Streamlit Secrets (next section).

3. Deploying on Streamlit Cloud

Streamlit Cloud hosts your apps online for free.

Step 1: Go to Streamlit Cloud

- **Action:** Open https://share.streamlit.io/.
- **Reason:** This is where you'll deploy your app.

Step 2: Sign Up / Log In

- **Action:** Sign up or log in using your GitHub account.
- **Reason:** Streamlit Cloud connects to GitHub for deployment.

Step 3: Deploy a New App

- **Action:** On the Streamlit Cloud dashboard, click New app.
- **Reason:** This starts the deployment process.

Step 4: Select Repository and File

- **Action:**
  1. **Repository:** Select your GitHub repository.
  2. **Branch:** Choose main or master.
  3. **Main file path:** Create an app.py file in your GitHub repository's root directory. Copy the Streamlit code (from Section 1, Step 5) into this app.py file. Then, enter app.py here.
  4. **Ensure all files are in the root:** Make sure app.py, requirements.txt, credentials.json, and Dockerfile are in your GitHub repository's main folder.
- **Reason:** Streamlit Cloud needs to know where your main app code is located.

Step 5: Configure Streamlit Secrets for Your API Key

- **Action:** Before clicking Deploy!, find "Advanced settings" or "Secrets." Add a new secret:
  - **Key:** GOOGLE_API_KEY (must match the key name in credentials.json).
  - **Value:** Your actual Gemini API Key.
- **Reason:** Even if you use credentials.json in your repository, **you must add your API key as a Streamlit Secret for deployment.** This is the secure way to provide sensitive information to your app in the cloud, as Streamlit Cloud loads these secrets as environment variables. Your app.py will then read this environment variable instead of the credentials.json file on the deployed server.

Step 6: Deploy the App

- **Action:** Click the Deploy! button.
- **Reason:** Streamlit Cloud will now set up your app's environment and launch it.

Step 7: View Your Deployed App

- **Action:** Once deployed, click the provided URL.
- **Reason:** Your app is now live and accessible online.

Congratulations! You've successfully created, linked, and deployed your Streamlit app.