**User Guide: Deploying Personal AI Assistant with Firestore and Gemini AI on Streamlit Community Cloud**

### 🚀 Project Overview

This project integrates Google Firestore and Gemini AI to create a Personal AI Assistant that allows users to:

* Chat with Gemini
* Store and retrieve tasks in Firestore
* View reminders by date

### 📁 Step-by-Step Deployment Instructions

#### ✅ 1. **Prepare Your App**

* Finalize your app.py Streamlit script.
* Ensure Firestore and Gemini API integrations are complete.
* Test locally to confirm functionality.

#### 🔍 2. **Set Up GCP (Google Cloud Platform)**

* Create a new GCP project or use an existing one.
* Enable **Firestore API** and **Vertex AI API**.

#### 🎓 3. **Create a Service Account**

* Go to **IAM & Admin > Service Accounts**.
* Create a new service account (e.g., streamlit@your-project.iam.gserviceaccount.com).
* Generate a JSON key for the service account.
* Rename the file as gcp\_service\_account.json (optional for clarity).

#### 📄 4. **Grant Required Roles to Service Account**

* IAM Roles required:
  + Firestore Viewer
  + Firestore User
  + Cloud Datastore User
  + Vertex AI User

Note: You may need to search for roles using **Manage Roles** > enable **All roles visibility**.

#### 🎯 5. **Create Your GitHub Repository**

* Upload app.py, requirements.txt, gcp\_service\_account.json, and README.md.
* Commit changes using the following script:

# Push your code to GitHub  
git init  
  
# Add GitHub remote repo  
git remote add origin https://github.com/yourusername/your-repo.git  
  
# Push code  
git add .  
git commit -m "Initial commit for Streamlit AI Assistant"  
git push -u origin main

#### 🚀 6. **Deploy on Streamlit Community Cloud**

* Visit <https://streamlit.io/cloud>
* Click **New app** > Connect your GitHub repo
* Set the main file as app.py
* Set the secrets securely:
* [gcp]  
  type = "service\_account"  
  project\_id = "your-project-id"  
  private\_key\_id = "xxx"  
  private\_key = "-----BEGIN PRIVATE KEY-----\n...\n-----END PRIVATE KEY-----\n"  
  client\_email = "..."  
  token\_uri = "https://oauth2.googleapis.com/token"
* (Optional) Add runtime.txt to specify Python version: python-3.11.9

### 📈 Troubleshooting

**Issue:** PermissionDenied: 403 Missing or insufficient permission

* **Solution:** Make sure Firestore User and Cloud Datastore User roles are granted.

**Issue:** pyaudio or other package build errors

* **Solution:** Remove unused heavy packages from requirements.txt. Streamlit Community Cloud has limited support for audio/voice libraries.

**Issue:** Firestore run\_query failure

* **Solution:** Double-check the GCP service account roles. Make sure you are reading from the correct collection.

### 🌎 Embedding in a Website

* You can embed your Streamlit app via iframe:

<iframe src="https://your-app-name.streamlit.app" width="100%" height="800"></iframe>

### 📖 Future Enhancements (Optional)

* Add user authentication via Firebase
* Use Google Calendar API for meeting/event reminders
* Integrate text-to-speech
* Convert to mobile app using Anvil, Flutter, or React Native wrappers

If you need the guide exported to PDF or formatted version for sharing, let me know!