**Step 4: Deploy the App Online**

Now, we will **host** the app online so schools can use it.

**🔹 Deploy on Streamlit Cloud**

1. Go to [Streamlit Cloud](https://share.streamlit.io/" \t "_new).
2. Click **"New App"**.
3. Select your **GitHub repository**.
4. In **"Main file path"**, enter:

CopyEdit

writing\_analysis.py

1. Click **"Deploy"**.

**📌 Step 5: Test the Live Platform**

✅ **Go to your deployed Streamlit link.**  
✅ **Log in, select year groups, and upload images.**  
✅ **Check that images are stored in Firebase.**  
✅ **Make comparative judgements and see if data is saved in Firestore.**

**📌 Step 6: Allow Schools to Download Data**

Now, update the code so schools can download **year group data & whole school data**.

**🔹 Add a Download Option**

python

CopyEdit

if st.session\_state.logged\_in:

st.sidebar.subheader("Download Data")

if st.sidebar.button("Download Year Group Data"):

docs = db.collection("writing\_samples").where("school", "==", school\_name).where("year\_group", "==", year\_group).stream()

data = [{"Image": doc.to\_dict()["image\_url"]} for doc in docs]

df = pd.DataFrame(data)

st.sidebar.download\_button("Download", df.to\_csv(index=False).encode("utf-8"), f"{year\_group}\_data.csv", "text/csv")

if st.sidebar.button("Download Whole School Data"):

docs = db.collection("writing\_samples").where("school", "==", school\_name).stream()

data = [{"Year Group": doc.to\_dict()["year\_group"], "Image": doc.to\_dict()["image\_url"]} for doc in docs]

df = pd.DataFrame(data)

st.sidebar.download\_button("Download", df.to\_csv(index=False).encode("utf-8"), "whole\_school\_data.csv", "text/csv")

**📌 Final Checklist**

✅ **Code pushed to GitHub**  
✅ **Firebase Storage & Firestore set up**  
✅ **Firebase credentials added to code**  
✅ **Streamlit app deployed & working**  
✅ **Schools can log in, upload, and rank writing samples**  
✅ **Year group & whole school data can be downloaded**

🎉 **Congratulations! Your writing comparison tool is now live!** 🚀

**✳️ Optional (Allow per-school access)**

If you later want to **allow access only for a specific school**, you can write rules like this:

js

CopyEdit

allow read, write: if request.auth != null && request.auth.token.email.matches(".\*@myschooldomain.com")

This would only allow users from @myschooldomain.com email addresses.

**✅ What You Should Do Now**

**1. Move the file into your project directory securely**

Make sure the file is inside your project folder:

bash

CopyEdit

writing\_comparative\_model/serviceAccountKey.json

**2. Tell Git to ignore this file**

You need to create a .gitignore file in your writing\_comparative\_model folder **if it doesn't already exist**.

Then, open (or create) that .gitignore file and add this line:

bash

CopyEdit

serviceAccountKey.json

This will **prevent Git from uploading your private key** to any public or private repository.

**🧠 What's Happening?**

Git tracks everything in your project unless you explicitly tell it not to. The .gitignore file does this. Adding serviceAccountKey.json to that file means:

* It won’t be committed.
* It won’t be pushed to GitHub or any repo.
* It stays safely on your machine.

**Extra Security Tip (Optional but Smart)**

If you've ever committed that key before (even by accident), **rotate it**:

1. Go to the Firebase console → Project Settings → Service Accounts.
2. Click "Generate new private key."
3. Delete the old one.
4. Update your code to point to the new file.