### **Step 1: Create a GitHub Repository**

1. **Go to GitHub**:
   1. Open [GitHub](https://github.com/).
2. **Create a new repository**:
   1. Click on the **"New"** button in the upper-right corner.
   2. Name your repository, e.g., upload-app.
   3. Select **Public** for the repository visibility (you can change it later).
   4. Click on **Create Repository**.

### **Push Your Code to GitHub**

Now you need to push your local Flask app code to your newly created GitHub repository.

1. **Open Terminal/Command Prompt** and navigate to your project folder:

bash

CopyEdit

cd path/to/your/my\_upload\_app

1. **Initialize Git**:
   1. Run the following commands in the terminal:

bash

CopyEdit

git init  
git add .  
git commit -m "First commit - Upload app"

1. **Push your code to GitHub**:
   1. Replace your-username with your actual GitHub username and run these commands:

bash

CopyEdit

git branch -M main  
git remote add origin <https://github.com/your-username/upload-app.git>git push -u origin main

### **Step 3: Create requirements.txt**

Render needs to know which Python packages to install. Here's how to create the requirements.txt:

1. Inside your **my\_upload\_app** folder, create a file called **requirements.txt**.
2. Add the following packages to requirements.txt:

txt

CopyEdit

flask  
gunicorn

1. Install gunicorn (if you haven't already):

bash

CopyEdit

pip install gunicorn

1. To make sure all installed packages are in requirements.txt, run:

bash

CopyEdit

pip freeze > requirements.txt

### **Step 4: Create a Procfile**

Render uses a Procfile to determine how to run your app. This file doesn't have an extension.

1. In your **my\_upload\_app** folder, create a **Procfile** (without any extension).
2. Add the following line to the Procfile:

txt

CopyEdit

web: gunicorn app:app

This tells Render to run your Flask app using Gunicorn.

### **Step 5: Create a Render Account and Deploy**

1. **Sign Up on Render**:
   1. Go to [Render.com](https://render.com/).
   2. Sign up with your **GitHub** account.
2. **Create a New Web Service**:
   1. After logging in, click on the **New +** button and select **Web Service**.
   2. Choose your **GitHub repository** (upload-app).
   3. Fill out the details as follows:

|  |  |
| --- | --- |
| **Field** | **Value** |
| **Name** | upload-app |
| **Branch** | main |
| **Build Command** | (Leave Blank) |
| **Start Command** | gunicorn app:app |
| **Environment** | Python |
| **Instance Type** | Free |

1. **Click "Create Web Service"**.

### **Step 6: Wait for Deployment**

Render will automatically build your app, install the dependencies, and start the web service.

* This can take **1-2 minutes**.

### **Step 7: Access Your Live App**

* After the build process is complete, Render will provide a public URL like this:

txt

CopyEdit

<https://upload-app.onrender.com/>

* Click on the URL to see your live Flask app running.

### **Additional Tips**

* **Testing Locally**: You can always run your Flask app locally by using:

bash

CopyEdit

python app.py

* **Debugging**: If you encounter issues, check the **Render logs** for error messages.
* **Security**: You might want to use environment variables (e.g., for database credentials) instead of hard-coding them in your app.

**To run in streamlit**

Opening the env. In streamlit

Open the project folder  
venv\Scripts\activate

python -m streamlit run app.py

**To reupload to git hub**

**git branch**

If it shows \* main, you’re good.

If it doesn’t show main, create and switch to it:

**git checkout -b main**

You need to **merge the GitHub repo’s initial commit (like a README.md)** with your local files:

bash

CopyEdit

**git pull origin main --allow-unrelated-histories**

This will merge the remote repo's files with your local files.  
If it asks you to write a merge commit message, just save and close the editor.

**Then push the code**

git add .

git commit -m "Initial commit"

git push -u origin main