# Thesis Handover

## GIT Files

Files you will need:  
1. App.py

2. Templates folder and file for HTML page.  
3. A database to store the data

4. Set up OpenAI secret key into a venv on your PC.

## Connection to AWS

**Step 1: Connect to Your EC2 Instance**

1. Open your terminal (PowerShell, Command Prompt, or Linux/macOS terminal).

Use SSH to connect to your instance (replace path/to/your-key.pem with the actual key file location):  
bash  
  
ssh -i "C:\Users\ashwa\Downloads\thesis.pem" ubuntu@3.133.115.61

* + If you didn't create a key pair during setup, you will need to create a new instance with a key pair.
  + If SSH still fails, check that **port 22 is open** in your AWS Security Group.

**Step 2: Update and Install Dependencies**

Once inside the EC2 instance, run the following commands:

**Update the package list:**bash  
  
sudo apt update && sudo apt upgrade -y

**Install necessary dependencies:**bash  
  
sudo apt install -y python3 python3-pip python3-venv git nginx

**Step 3: Clone Your Flask App from GitHub**

If your Flask app is on GitHub:

**Navigate to your home directory:**bash  
  
cd ~

**Clone the repository (replace with your actual GitHub repo URL):**bash  
  
git clone https://github.com/ashwathchadha/thesis.git

**Enter the project folder:**bash  
  
cd thesis

**Step 4: Set Up a Virtual Environment**

**Create the virtual environment:**bash  
  
python3 -m venv venv

**Activate the virtual environment:**bash  
  
source venv/bin/activate

**Install required Python packages:**bash  
  
pip install flask requests openai requests-oauthlib gunicorn

**Step 5: Run Your Flask App**

**Start your app manually for testing:**bash  
  
python3 app.py

* + If it runs without errors, **your Flask app is working!**
  + Stop the app using CTRL + C.

**Run the app in the background using Gunicorn (a production-ready server):**bash  
  
gunicorn --bind 0.0.0.0:5000 app:app

Your app should now be accessible at:  
cpp  
  
http://3.133.115.61:5000

**Step 6: Set Up Nginx for Public Access**

To serve your Flask app on port **80 (HTTP)** instead of 5000:

**Edit the Nginx config file:**bash  
  
sudo nano /etc/nginx/sites-available/flask\_app

**Add the following configuration:**nginx  
  
server {

    listen 80;

    server\_name 3.133.115.61;

    location / {

        proxy\_pass http://127.0.0.1:5000;

        proxy\_set\_header Host $host;

        proxy\_set\_header X-Real-IP $remote\_addr;

        proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

    }

}

* + Save and exit (CTRL + X, then Y, then Enter).

**Enable the config and restart Nginx:**bash  
  
sudo ln -s /etc/nginx/sites-available/flask\_app /etc/nginx/sites-enabled

sudo systemctl restart nginx

**Step 7: Open Port 80 in AWS Security Group**

1. Go to AWS **EC2 Dashboard** → **Instances**.
2. Click on **Security Groups** (on the left).
3. Find your instance's security group and **edit inbound rules**.
4. **Allow HTTP (port 80) from anywhere (0.0.0.0/0)**.

**Step 8: Access Your Flask App**

Now, go to:

cpp

http://3.133.115.61

**Next Steps**

If you **update your app on GitHub**, run:  
bash  
  
Cd the

cd ~/thesis

git pull origin main

sudo systemctl restart flask\_app

**Force delete a directory (Use with caution):**

bash

rm -rf folder\_name

Rm

## Data collection

1. Currently, the data storage can get messy if multiple users are using at the same time. To prevent this, modify code to make separate AWS directories for each user.

2. Image submitted into the interface are saved in these directories.

3. Data in the log file contains input from user and output from AI agent. New ways to store this data could be beneficial.