- # UPLOAD PROJECT ON GIT AND DOCKER AT SAME TIME

1. Create a virtual environment

code type on terminal → python –m venv env_name

2. Create file_name.py file

Write your code
If want to run code type on terminal → python file_name.py

3. Create requirements.txt file

Write libraries that needed to run file
To install libraries type on terminal → pip install –r requirements.txt

4. Create docker file

```
# Use an official Python runtime as a parent image
FROM python:3.9-slim
# Set the working directory in the container
WORKDIR /app
# Copy the current directory contents into the container at /app
COPY . /app
# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
# Make port 8000 available to the world outside this container
EXPOSE 8000
# Define environment variable
ENV FLASK_APP=app.py
# Run app.py when the container launches
CMD ["flask", "run", "--host=0.0.0.0", "--port=8000"]
```

Write according to yourself

5. Create .GitHub folder in this workflows folder in this a filename.yml file

.github/workflows/filename.yml

Write yml file file code

Or give this commend to any ai/gpt this will give you docker file and .yml file contant
this is my app.py in flask applicatio . mY task is i have to create a ci/cd piepeline so that i can update in application in locall and when i push it to github then ci/cd will automatically push it to dockerhub . I have main file is app.py) .
also upload .py file

6. Create new repository on Github

Settings > **Secrets** > **New repository secret**.

Add the following secrets:

- DOCKER_HUB_USERNAME: Your Docker Hub username.
- DOCKER_HUB_TOKEN: Your Docker Hub access token (you can generate this in Docker Hub under Account Settings > Security).

7. Push your code to github

The GitHub Actions workflow will automatically build the Docker image and push it to Docker Hub.

8. Run the Docker Image Locally

After the Docker image is pushed to Docker Hub, you can pull and run it locally:

docker pull <your-docker-hub-username>/your-repo-name:latest

docker run -p 8000:8000 <your-docker-hub-username>/your-repo-name:latest

9. To confirm that the app.py is running

Open your browser and go to http://localhost:8000

10. Update the Tag

docker tag your-docker-hub-username/aadi:latest your-docker-hub-username/aadi:v1.1

11. **Push the Tagged Image to Docker Hub**
    docker push your-docker-hub-username/aadi:v1.1