Flask App Containerization and Docker Hub Deployment

🎯 Objective

To create a Flask web application, containerize it using Docker, and deploy it to Docker Hub.

📁 Folder Structure

```
flask-docker-app/
├── app.py
├── requirements.txt
└── Dockerfile
```

📝 Step-by-Step Procedure

✅ Step 1: Create the Flask App

app.py

```
from flask import Flask, render_template_string

app = Flask(__name__)

HTML_TEMPLATE = """
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Flask Docker App</title>
  <style>
    body {
      background: linear-gradient(to right, #00c6ff, #0072ff);
      color: white;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      text-align: center;
      padding-top: 100px;
    }
```

```
      h1 {
        font-size: 3em;
        margin-bottom: 0.2em;
      }
      p {
        font-size: 1.5em;
      }
      .card {
        background: rgba(255, 255, 255, 0.1);
        padding: 2em;
        border-radius: 20px;
        box-shadow: 0 0 20px rgba(0,0,0,0.3);
        width: 50%;
        margin: auto;
      }
    </style>
</head>
<body>
    <div class="card">
      <h1>🚀 Welcome to Flask + Docker!</h1>
      <p>This is a vibrant web page running inside a Docker container.</p>
      <p>Change the code → Rebuild → Restart Docker to see updates!</p>
    </div>
</body>
</html>
"""

@app.route('/')
def home():
    return render_template_string(HTML_TEMPLATE)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

✅ Step 2: Create requirements.txt

```
Flask==2.2.5
```

✅ Step 3: Create Dockerfile

```
FROM python:3.9-slim

WORKDIR /app
```

```
COPY requirements.txt .
RUN pip install -r requirements.txt

COPY . .

EXPOSE 5000
CMD ["python", "app.py"]
```

✅ Step 4: Build Docker Image

docker build -t flask-docker-app .

✅ Step 5: Run the Container

If port 5000 is free:

docker run -p 5000:5000 flask-docker-app

If port 5000 is occupied:

docker run -p 5050:5000 flask-docker-app

Visit: http://localhost:5050

✅ Step 6: Login to Docker Hub

docker login

✅ Step 7: Tag the Docker Image

docker tag flask-docker-app <your-dockerhub-username>/flask-docker-app:latest

✅ Step 8: Push to Docker Hub

docker push <your-dockerhub-username>/flask-docker-app:latest

✅ To Test on Any Machine

docker pull <your-dockerhub-username>/flask-docker-app:latest

docker run -p 5050:5000 <your-dockerhub-username>/flask-docker-app:latest

📌 Notes

• Always rebuild your image after modifying app.py.

• Docker Hub must be public to allow others to pull without authentication.