Strucuture

project\_folder/

│

├── app.py

├── Dockerfile

├── requirements.txt <-- Add this file

├── templates/

│ └── login.html

App.py

from flask import Flask, render\_template, request, redirect, url\_for

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('login.html')

@app.route('/login', methods=['POST'])

def login():

username = request.form['username']

password = request.form['password']

# You can add your login logic here

# For simplicity, let's just check if the username and password are not empty

if username and password:

return f'Logged in as {username}'

else:

return 'Invalid login credentials'

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=7000, debug=True)

Dockerfile

# Use an official Python runtime as a parent image

FROM python:3.8-slim

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install Flask

RUN pip install --trusted-host pypi.python.org Flask

# Make port 7000 available to the world outside this container

EXPOSE 7000

# Run app.py when the container launches

CMD ["python", "app.py"]

Login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login Form</title>

</head>

<body>

<h2>Login Form</h2>

<form action="{{ url\_for('login') }}" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required><br>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required><br>

<input type="submit" value="Login">

</form>

</body>

</html>

To run and build image

docker build -t flask-login-app .

docker run -p 7000:7000 flask-login-app

Then go to http://localhost:7000