**Python-MYsql Docker Assignment**

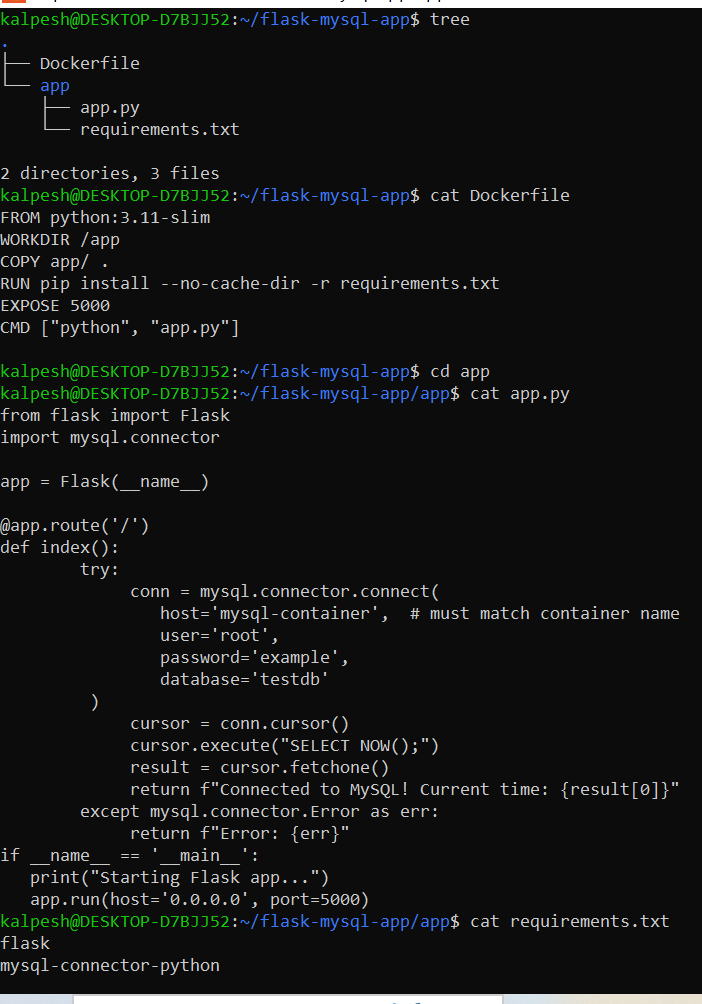
**📁 Project Overview**

The project runs:

* A **Flask web application** in a Docker container.
* A **MySQL database** in a separate Docker container.
* Both containers communicate over a custom Docker **bridge network**.

**🔧 1. Project Files**

**1.created files app/app.py (Flask App) & app/requirements.txt and Dockerfile**



**🚀 Step-by-Step Instructions to Run the Project**

**Step 1️⃣: Extract and navigate into the project**

cd flask-mysql-app

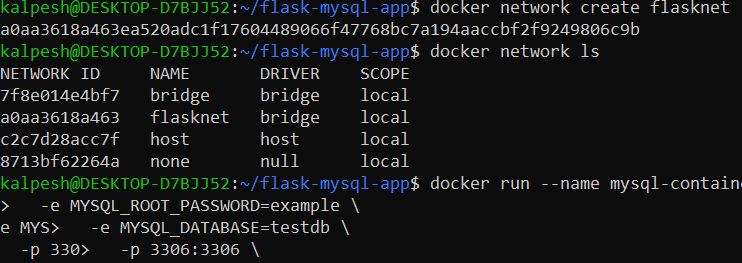
**Step 2️⃣: Create a custom Docker network**

docker network create flasknet

So both containers can communicate using container names as hostnames.

Docker network ls

To view all docker network list



**Step 3️⃣: Run MySQL container**

docker run --name mysql-container --network flasknet \

-e MYSQL\_ROOT\_PASSWORD=example \

-e MYSQL\_DATABASE=testdb \

-p 3306:3306 \

-d mysql:8.0

* --name mysql-container: Container name (used by Flask to connect).
* --network flasknet: Connects it to the custom network.
* -e ...: Sets root password and database.
* -p 3306:3306: Exposes MySQL port for outside access (optional).
* -d: Detached mode.

**Step 4️⃣: Build the Flask Docker image**

docker build -t flask-mysql-app .

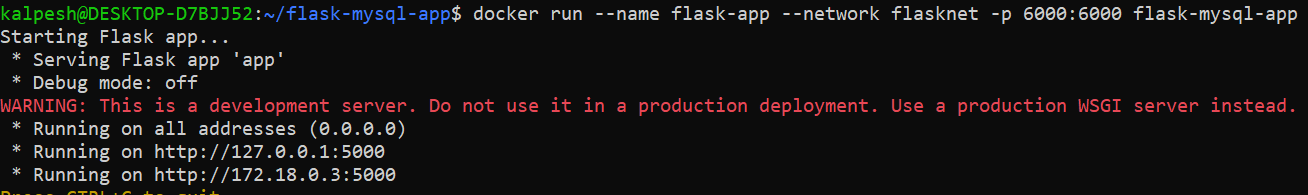
* -t: Tag the image.
* .: Build from current directory (Dockerfile present here).



**Step 5️⃣: Run Flask app container**

docker run --name flask-app --network flasknet -p 5000:5000 flask-mysql-app

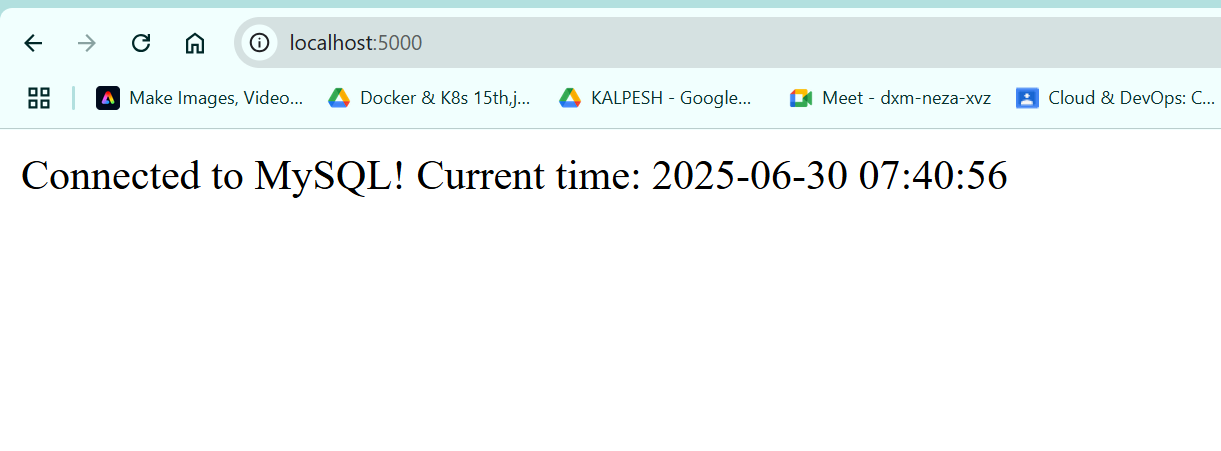
* --name flask-app: Name of the Flask container.
* --network flasknet: Must match MySQL's network.
* -p 5000:5000: Maps Flask port for browser access.



**Step 6️⃣: Access the App in Browser**

http://localhost:5000

✅ You should see something like:

  
Connected to MySQL! Current time: 2025-06-30 07:40:56

**🧼 Optional: Cleanup**

docker stop flask-app mysql-container

docker rm flask-app mysql-container

docker network rm flasknet