**Create Flask App**

**app/app.py**:

from flask import Flask, render\_template

import mysql.connector

app = Flask(\_\_name\_\_, template\_folder='templates')

@app.route('/')

def index()

    try:

        conn = mysql.connector.connect(

            host='mysql',

            user='root',

            password='example',

            database='testdb',

            port=3306

        )

        cursor = conn.cursor()

        cursor.execute("SELECT NOW();")

        result = cursor.fetchone()

        cursor.close()

        conn.close()

        return f"Connected to MySQL! Current time: {result[0]}"

    except mysql.connector.Error as err:

        return f"Error: {err}"

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True, host="0.0.0.0", port=5000)  # Flask will listen on port 5000

### Create Requirements File

**app/requirements.txt**

flask

mysql-connector-python

### Create Dockerfile

**Dockerfile**:

FROM python:3.9-slim

WORKDIR /app

COPY app/ .

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 5000

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

### Create Dockerfile-compose.yml

**Dockerfile-compose.yml**:

services:

  pythonapp:

    container\_name: pythonapp

    build:                         # Bulid the image

      context: .

      dockerfile: Dockerfile       # Dockerfile for the python application

    ports:

      - 8000:5000                  # expose port 5000 of the container to port 8000 on the host

    networks:

      - flasknet

    depends\_on:

      - mysql

  # mysql: MySQL database service

  mysql:

    image: mysql:latest

    container\_name: mysql

    environment:

      MYSQL\_ROOT\_PASSWORD: example

      MYSQL\_DATABASE: testdb

    ports:

      - 3306:3306

    networks:

      - flasknet

networks:

  flasknet:

    driver: bridge

### Run Flask App Container

* Docker-compose up -d

### Access the App

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