# Python Flask and MySQL with Docker Compose - Project Documentation

## Table of Contents

* 1. Project Overview
* 2. Prerequisites
* 3. Project Structure
* 4. Code Files
* - app.py
* - requirements.txt
* - Dockerfile
* - docker-compose.yml
* 5. Step-by-Step Guide
* - Step 1: Set up the Flask Application
* - Step 2: Set up the MySQL Database
* - Step 3: Create the Dockerfile
* - Step 4: Create the Docker Compose File
* - Step 5: Run the Project
* 6. Testing and Verification
* 7. Stopping and Cleaning Up
* 8. Summary

## 1. Project Overview

This project demonstrates how to build a basic Python Flask application that connects to a MySQL database, with both components running in separate Docker containers. Docker Compose is used to orchestrate and manage the multi-container application setup, making it easy to run and scale the application. This setup allows seamless database interaction within a containerized environment.

**Key Features:**- Flask Web Application  
- MySQL Database  
- Docker Containers  
- Docker Compose

## 2. Prerequisites

To run this project, you need:  
- Docker  
- Docker Compose  
Additional knowledge of Python, Flask, MySQL, and Docker is helpful.

## 3. Project Structure

Directory structure:  
  
flask\_app/  
├── app.py  
├── requirements.txt  
├── Dockerfile  
└── docker-compose.yml

## 4. Code Files

### app.py  
Defines basic routes and database connections for the Flask app.  
  
Code Explanation:  
- Imports  
- App Configuration  
- Routes  
  
```python  
from flask import Flask, jsonify  
from flask\_mysqldb import MySQL  
app = Flask(\_\_name\_\_)  
  
# Config  
app.config['MYSQL\_HOST'] = 'mysql-db' ...

## 5. Step-by-Step Guide

Step 1: Set up Flask Application  
1. Create `app.py`  
  
Step 2: Set up MySQL Database in `docker-compose.yml`  
  
Step 3: Create Dockerfile  
  
Step 4: Create docker-compose.yml  
  
Step 5: Run the Project  
```bash  
docker-compose up -d  
```

## 6. Testing and Verification

After running containers, test the application by visiting:  
- `http://localhost:5000/`  
- `http://localhost:5000/users`  
Check container status with:  
```bash  
docker-compose ps  
```

## 7. Stopping and Cleaning Up

To stop and remove containers, run:  
```bash  
docker-compose down  
```

## 8. Summary

This project demonstrates a basic setup using Flask, MySQL, and Docker Compose for efficient development or production environments.