Project Documentation Report

Project Title: Dockerized MySQL - Airplane Passengers Database

Author: Abdulharis Shaikh

Tech Stack: Docker, MySQL 8.0

Date: June 28, 2025

# 1. Introduction

This project demonstrates how to containerize a MySQL database using Docker. The container is pre-configured to create a `FlightEn` database and load it with a passenger bookings table and initial records directly from a SQL dump file (`FlightEn.sql`).

# 2. Objectives

✔️ Create a Docker image that includes MySQL 8.0 and a preloaded database

✔️ Learn Dockerfile usage, image building, and container configuration

✔️ Automate SQL data import during container startup

✔️ Practice secure credential handling using environment variables

# 3. Tools & Technologies Used

|  |  |
| --- | --- |
| Technology/Tool | Purpose |
| Docker | Containerization platform |
| MySQL 8.0 | Database server |
| Dockerfile | Image automation script |
| SQL | Data definition and insertion |
| CMD / Terminal | Container and DB management |

# 4. Project Structure

DATABASE/

* ├── Dockerfile
* └── FlightEn.sql

# 5. Environment Configuration

MYSQL\_ROOT\_PASSWORD

MYSQL\_DATABASE

MYSQL\_USER

MYSQL\_PASSWORD

# 6. Commands Used

* Build Docker image:

docker build -t flighten-mysql .

* Run Docker container:

docker run --name flighten-db -p 3307:3306 -d flighten-mysql

* Access MySQL CLI:

docker exec -it flighten-db mysql -u root -p

* Inside MySQL:

USE FlightEn;  
SELECT \* FROM flight\_bookings;

# 7. Output Verification

After successful execution, the `flight\_bookings` table is created and preloaded with 3 records. A sample query (`SELECT \* FROM flight\_bookings;`) was executed and displayed correct results, confirming the containerized database works as expected.

# 8. Learning Outcomes

✔️ Learned how to automate MySQL database setup in Docker

✔️ Built confidence working with SQL in containerized environments

✔️ Improved understanding of environment variable usage for secure DB config

✔️ Experienced full Docker image lifecycle: build, run, exec, and query

# 9. Conclusion

This project provides a complete hands-on experience in Dockerized database deployment. It lays the groundwork for future integration with web services, container orchestration, and CI/CD pipelines.

