Assignment 1 – Docker Containers
Cloud Computing for Data Analysis (ITCS 6190/8190, Fall 2025)

Overview
This project demonstrates a **multi-container Docker setup** consisting of:
1. A **PostgreSQL** database service, seeded with a `trips` table and initial data.
2. A **Python application** service that connects to the database, queries data, computes basic statistics, and outputs results to both the terminal and a JSON file.

The goal is to practice container orchestration with **Docker Compose**, learn inter-container networking, and build reproducible workflows.

Prerequisites
Make sure you have the following installed:
- [Docker Desktop](https://docs.docker.com/get-docker/)
- [Docker Compose](https://docs.docker.com/compose/)
- GitHub Desktop or Git for version control
- IDE (VS Code, PyCharm, etc.)

Before starting:
- Verify you can run basic Docker commands (`docker build`, `docker run`, `docker ps`).
- Ensure **port 5432** is available, or update `compose.yml` if needed.

Project Structure
```
■■■ app/
■ ■■■ Dockerfile
■ ■■■ main.py
■■■ db/
■ ■■■ Dockerfile
■ ■■■ init.sql
■■■ compose.yml
■■■ Makefile
■■■ README.md
```

Running the Stack

Using Docker Compose directly
Build and run both containers:
```bash
docker compose up --build
```

Stop and remove containers:
```bash
docker compose down -v
```

```

### Using Makefile (recommended)
Run everything with:
```bash
make all
```

Other available targets:
- `make build` → Builds the images
- `make up` → Starts the services
- `make down` → Stops and removes services
- `make clean` → Cleans containers & recreates `out/` folder

Example Output
When the stack runs, the Python app will:
- Print a summary to the terminal
- Write results to `/out/summary.json`

Example:
```json
{
"total_trips": 26,
"avg_fare_by_city": [
{"city": "Charlotte", "avg_fare": 22.84},
{"city": "New York", "avg_fare": 28.30},
{"city": "San Francisco", "avg_fare": 26.10},
{"city": "Chicago", "avg_fare": 25.40},
{"city": "Boston", "avg_fare": 22.00}
],
"top_by_minutes": [
{"city": "New York", "minutes": 52, "fare": 55.10},
{"city": "San Francisco", "minutes": 55, "fare": 57.25},
{"city": "Boston", "minutes": 50, "fare": 49.75}
]
}
```

Troubleshooting
- **Database not ready** → The Python app retries connection until Postgres is healthy.
- **Permission issues with `/out/`** → Ensure the folder exists and has write access.
- **Port conflicts** → Change the port mapping in `compose.yml` if 5432 is in use.

```