Assignment 1 Submission: Docker Containers

Sumanth Dasari
801420544

1. Exact Commands to Build and Run the Stack

The entire stack was built and run using the following command from the project's root directory:

docker compose up --build

2. Pasted Summary from stdout

The following JSON summary was printed to stdout by the Python application after successfully connecting to the database and running the queries.


```
{
  "total_trips": 6,
  "avg_fare_by_city": [
    {
      "city": "Charlotte",
      "avg_fare": 16.25
    },
    {
      "city": "New York",
      "avg_fare": 19.0
    },
    {
      "city": "San Francisco",
      "avg_fare": 20.25
    }
  ],
  "top_5_longest_trips": [
    {
      "city": "San Francisco",
      "minutes": 28,
      "fare": 29.3
    },
```

```
    {
      "city": "New York",
      "minutes": 26,
      "fare": 27.1
    },
    {
      "city": "Charlotte",
      "minutes": 21,
      "fare": 20.0
    },
    {
      "city": "Charlotte",
      "minutes": 12,
      "fare": 12.5
    },
    {
      "city": "San Francisco",
      "minutes": 11,
      "fare": 11.2
    }
  ]
}
```


3. Contents of out/summary.json

The summary.json file was created in the out/ directory and contains the exact same content as the stdout summary.


```
{
  "total_trips": 6,
  "avg_fare_by_city": [
    {
      "city": "Charlotte",
      "avg_fare": 16.25
    },
    {
      "city": "New York",
      "avg_fare": 19.0
    },
    {
      "city": "San Francisco",
      "avg_fare": 20.25
```

```
    }
  ],
  "top_5_longest_trips": [
    {
      "city": "San Francisco",
      "minutes": 28,
      "fare": 29.3
    },
    {
      "city": "New York",
      "minutes": 26,
      "fare": 27.1
    },
    {
      "city": "Charlotte",
      "minutes": 21,
      "fare": 20.0
    },
    {
      "city": "Charlotte",
      "minutes": 12,
      "fare": 12.5
    },
    {
      "city": "San Francisco",
      "minutes": 11,
      "fare": 11.2
    }
  ]
}
```


4. Short Reflection

Through this assignment, I learned the practical workflow of creating a reproducible, multi-container application with Docker Compose. A key takeaway was understanding how Docker's service networking allows containers to communicate using simple hostnames, like db. Implementing the healthcheck and depends_on conditions was crucial for ensuring the application only started after the database was fully ready, which prevents common startup race conditions. In the future, I would improve this stack by using a .env file to manage environment variables, which would keep credentials and configuration separate from the compose.yml file.