1. Commands used to build and run the stack

The commands used to build and run the stack are taken directly from the "Exact commands to run/stop" section of the README.

To clean the project and start from scratch, the following command was used:

make all

To run the stack without cleaning:

docker compose up --build
Or
make up

2. The contents of out/summary.json


```
{
  "total_trips": 6,
  "avg_fare_by_city": [
    {
      "city": "San Francisco",
      "avg_fare": 20.25
    },
    {
      "city": "New York",
      "avg_fare": 19.0
    },
    {
      "city": "Charlotte",
      "avg_fare": 16.25
    }
  ],
  "top_by_minutes": [
    {
      "id": 6,
      "city": "San Francisco",
      "minutes": 28.0,
      "fare": 29.3
    },
    {
      "id": 4,
      "city": "New York",
      "minutes": 26.0,
      "fare": 27.1
```

```
    },
    {
      "id": 2,
      "city": "Charlotte",
      "minutes": 21.0,
      "fare": 20.0
    },
    {
      "id": 1,
      "city": "Charlotte",
      "minutes": 12.0,
      "fare": 12.5
    },
    {
      "id": 5,
      "city": "San Francisco",
      "minutes": 11.0,
      "fare": 11.2
    },
    {
      "id": 3,
      "city": "New York",
      "minutes": 9.0,
      "fare": 10.9
    }
  ]
}
```


3. Output

```
PS C:\Users\Mitsy\5190-assignment1> make up
docker compose up --build
[+] Building 4.0s (21/21) FINISHED
 => [internal] load local bake definitions
0.0s
 => => reading from stdin 983B
0.0s
 => [db internal] load build definition from Dockerfile
0.1s
 => => transferring dockerfile: 107B
0.0s
 => [app internal] load build definition from Dockerfile
0.1s
 => => transferring dockerfile: 178B
0.0s
```

```
 => [db internal] load metadata for docker.io/library/postgres:16
1.0s
 => [app internal] load metadata for docker.io/library/python:3.11-slim
0.9s
 => [auth] library/postgres:pull token for registry-1.docker.io
0.0s
 => [auth] library/python:pull token for registry-1.docker.io
0.0s
 => [app internal] load .dockerignore
0.1s
 => => transferring context: 2B                                                0.0s
 => [db internal] load .dockerignore
0.1s
 => => transferring context: 2B                                                0.0s
 => [app 1/4] FROM
docker.io/library/python:3.11-slim@sha256:91e9d01cf4bd56be7128c603506b6fe367ef7506f9f2
ad8f3a908aeec8941bb9   0.1s
 => => resolve
docker.io/library/python:3.11-slim@sha256:91e9d01cf4bd56be7128c603506b6fe367ef7506f9f2
ad8f3a908aeec8941bb9       0.1s
 => [app internal] load build context
0.0s
 => => transferring context: 2.27kB
0.0s
 => [db internal] load build context                                           0.1s
 => => transferring context: 30B
0.0s
 => [db 1/2] FROM
docker.io/library/postgres:16@sha256:29a9393ca3b02d1f5ec46ea6e73bf6af9cb4400e8d2a127
bbb1e1f32c12b2c6f         0.1s
 => => resolve
docker.io/library/postgres:16@sha256:29a9393ca3b02d1f5ec46ea6e73bf6af9cb4400e8d2a127
bbb1e1f32c12b2c6f           0.1s
 => CACHED [app 2/4] RUN pip install --no-cache-dir psycopg[binary]==3.2.1
0.0s
 => CACHED [app 3/4] WORKDIR /app
0.0s
 => [app 4/4] COPY main.py /app/
0.1s
 => CACHED [db 2/2] COPY ./init.sql /docker-entrypoint-initdb.d/init.sql
0.0s
 => [db] exporting to image                                                    0.5s
 => => exporting layers                                                       0.0s
```

```
 => => exporting manifest
sha256:cb97840ed7e0297ff43c2bf4a4d5a477bbad3373cfc473cb3e576533715e389a
0.0s
 => => exporting config
sha256:a43f589592f83787b084d766ad8e8838f51532431eee0a024b7577171be97945
0.0s
 => => exporting attestation manifest
sha256:ea36540b80e7d02fbbb237db1c6376df79baa39a984ed3ee6da887ec9ba407fd
0.1s
 => => exporting manifest list
sha256:5f875e2ca81e2d97f5246f669c398cc9d6f618256f1f2194158e78b7143e53de
0.1s
 => => naming to docker.io/library/5190-assignment1-db:latest
0.0s
 => => unpacking to docker.io/library/5190-assignment1-db:latest
0.1s
 => [app] exporting to image                                          1.1s
 => => exporting layers                                               0.4s
 => => exporting manifest
sha256:65a84dcf8c21abe128bf17e5f242fc0057a312f6eef5a643c1065ebc8783accb
0.1s
 => => exporting config
sha256:1c52acf6dad72715281ef96cdca8d7a148a245f2163a081c2c72b13cf1e96ff7
0.1s
 => => exporting attestation manifest
sha256:b2469542a4fd3407670db55775b05c2d5700d62c67388156060c61793d4c9fc8
0.1s
 => => exporting manifest list
sha256:b089a131cc89eaee838897b9e051b01e6c627d9095d59fd2a5c7f4ca6284bfad
0.1s
 => => naming to docker.io/library/5190-assignment1-app:latest
0.0s
 => => unpacking to docker.io/library/5190-assignment1-app:latest
0.2s
 => [db] resolving provenance for metadata file
0.0s
 => [app] resolving provenance for metadata file
0.0s
[+] Running 4/4
 ✔ 5190-assignment1-db              Built
0.0s
 ✔ 5190-assignment1-app             Built
0.0s
```

```
 ✔ Container 5190-assignment1-db-1   Recreated
0.5s
 ✔ Container 5190-assignment1-app-1  Recreated
0.5s
Attaching to app-1, db-1
db-1  |
db-1  | PostgreSQL Database directory appears to contain a database; Skipping initialization
db-1  |
db-1  | 2025-09-09 23:02:08.369 UTC [1] LOG:  starting PostgreSQL 16.10 (Debian
16.10-1.pgdg13+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 14.2.0-19) 14.2.0, 64-bit
db-1  | 2025-09-09 23:02:08.370 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db-1  | 2025-09-09 23:02:08.370 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db-1  | 2025-09-09 23:02:08.387 UTC [1] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
db-1  | 2025-09-09 23:02:08.412 UTC [29] LOG:  database system was shut down at
2025-09-09 23:01:33 UTC
db-1  | 2025-09-09 23:02:08.442 UTC [1] LOG:  database system is ready to accept connections
app-1  | === Summary ===
app-1  | {
app-1  |   "total_trips": 6,
app-1  |   "avg_fare_by_city": [
app-1  |     {
app-1  |       "city": "San Francisco",
app-1  |       "avg_fare": 20.25
app-1  |     },
app-1  |     {
app-1  |       "city": "New York",
app-1  |       "avg_fare": 19.0
app-1  |     },
app-1  |     {
app-1  |       "city": "Charlotte",
app-1  |       "avg_fare": 16.25
app-1  |     }
app-1  |   ],
app-1  |   "top_by_minutes": [
app-1  |     {
app-1  |       "id": 6,
app-1  |       "city": "San Francisco",
app-1  |       "minutes": 28.0,
app-1  |       "fare": 29.3
app-1  |     },
app-1  |     {
app-1  |       "id": 4,
app-1  |       "city": "New York",
```

```
app-1  |      "minutes": 26.0,
app-1  |      "fare": 27.1
app-1  |    },
app-1  |    {
app-1  |      "id": 2,
app-1  |      "city": "Charlotte",
app-1  |      "minutes": 21.0,
app-1  |      "fare": 20.0
app-1  |    },
app-1  |    {
app-1  |      "id": 1,
app-1  |      "city": "Charlotte",
app-1  |      "minutes": 12.0,
app-1  |      "fare": 12.5
app-1  |    },
app-1  |    {
app-1  |      "id": 5,
app-1  |      "city": "San Francisco",
app-1  |      "minutes": 11.0,
app-1  |      "fare": 11.2
app-1  |    },
app-1  |    {
app-1  |      "id": 3,
app-1  |      "city": "New York",
app-1  |      "minutes": 9.0,
app-1  |      "fare": 10.9
app-1  |    }
app-1  |  ]
app-1  | }
app-1 exited with code 0
```


4. Reflection

I learned that using docker compose with a make file can greatly simplify the process of running a multi-container application. The make all command is particularly useful as it handles the entire clean-and-run cycle in a single, repeatable step. This stack also highlights the value of using mounted volumes to persist data, such as the out/ directory, after containers have been stopped. To improve this setup, I would consider adding more specific error handling within the application itself to provide more informative messages beyond just "Waiting for database..." and perhaps a mechanism to automatically create the out/ directory if it doesn't exist.