

Name: **Dhruv Maheshbhai Patel**
Student ID: **9062297**
Subject: **Database Automation**

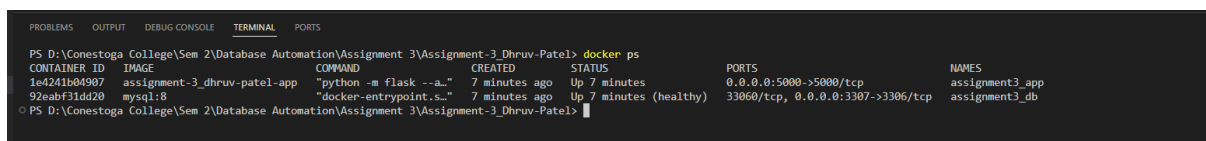
Assignment Report 2

In this project, a portion of NYC 311 service requests is extracted, processed as data, loaded into a MySQL database, and a web application is provided to visualize the count of complaints by each borough. Selenium is automated and used to test Docker and is available and reproducible.

Dataset

Dataset: NYC 311 Service Requests
- Slice used for testing: `data/fixtures.csv` (50 rows)

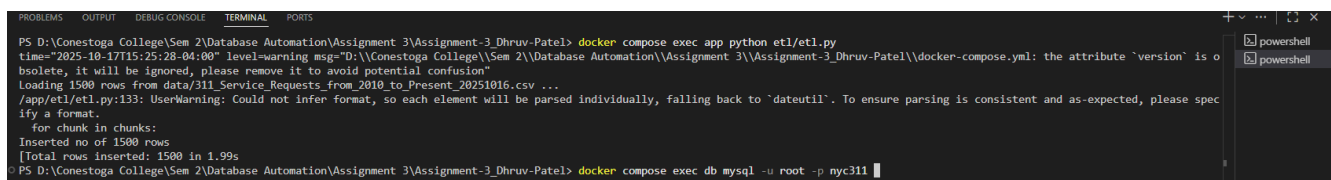
Step 1 – Setup:



```
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                NAMES
1e4241604907   assignment-3_dhruv-patel-app         "python -m flask --a..." 7 minutes ago  Up 7 minutes  0.0.0.0:5000->5000/tcp              assignment3_app
92eab731d420   mysql:8                              "docker-entrypoint.s..." 7 minutes ago  Up 7 minutes  33060/tcp, 0.0.0.0:3307->3306/tcp    assignment3_db
```

Step 2 – ETL (Extract, Transform, Load)

The screenshot depicts the ETL script in operation and loading a slice of the dataset into the service_requests table. The resulting output of the script validates the quantity of rows that have been added, and it also illustrates that the script can clean, convert and import CSV data to MySQL.



```
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel> docker compose exec app python etl/etl.py
time="2025-10-17T15:25:28-04:00" level=warning msg="D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
Loading 1500 rows from data/311_Service_Requests_from_2010_to_Present_20251016.csv ...
/app/etl/etl.py:133: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify a format.
for chunk in chunks:
  Inserted no of 1500 rows
[Total rows inserted: 1500 in 1.99s
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel> docker compose exec db mysql -u root -p nyc311
```

Step 3 – Database Schema & Indexes

This is a screenshot of the schema of the service requests table with column names, data type, and indexes. It confirms that the database format matches the ETL process cleaned and transformed data guaranteeing the efficiency of storage and retrieval.

```
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel> docker compose exec db mysql -u root -p nyc311
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.

mysql> SHOW TABLES;
+-----+
| Tables_in_nyc311 |
+-----+
| ingestion_log     |
| service_requests  |
+-----+
2 rows in set (0.01 sec)

mysql> SELECT COUNT(*) FROM service_requests;SELECT COUNT(*) FROM service_requests;
+-----+
| COUNT(*) |
+-----+
| 1500     |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COUNT(*) FROM service_requests;
+-----+
| COUNT(*) |
+-----+
| 1500     |
+-----+
1 row in set (0.00 sec)

mysql> SELECT request_id, complaint_type, borough, status, created_datetime
-> FROM service_requests
-> LIMIT 10;
+-----+-----+-----+-----+-----+
| request_id | complaint_type | borough | status | created_datetime |
+-----+-----+-----+-----+-----+
| 63941259   | Consumer Complaint | QUEENS | Closed | 2025-01-31 15:53:12 |
| 63941260   | Illegal Parking   | BROOKLYN | Closed | 2025-01-31 17:47:12 |
| 63941279   | Abandoned Vehicle | BROOKLYN | Closed | 2025-01-31 16:57:16 |
| 63941283   | Residential Disposal Complaint | BROOKLYN | Closed | 2025-01-31 15:25:51 |
| 63941300   | Abandoned Vehicle | BROOKLYN | Closed | 2025-01-31 15:38:40 |
| 63941302   | Abandoned Vehicle | QUEENS | Closed | 2025-01-31 16:45:29 |
| 63941303   | Abandoned Vehicle | QUEENS | Closed | 2025-01-31 16:11:58 |
| 63941318   | Abandoned Vehicle | BROOKLYN | Closed | 2025-01-31 15:04:23 |
| 63941323   | Damaged Tree      | BROOKLYN | Closed | 2025-01-31 16:52:24 |
| 63941329   | Abandoned Vehicle | QUEENS | Closed | 2025-01-31 16:03:14 |
+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql>
```

Explain indexes for BROOKLYN, Noise, Date:

```
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel> docker compose exec db mysql -u root -p nyc311
| service_requests | 1 | idx_month_key | 1 | month_key | A | 1 | NULL | NULL | YES | BTREE | | | YES |
| NULL |
+-----+
7 rows in set (0.05 sec)

mysql> EXPLAIN SELECT * FROM service_requests WHERE borough = 'BROOKLYN';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | service_requests | NULL | ref | idx_borough | idx_borough | 203 | const | 472 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

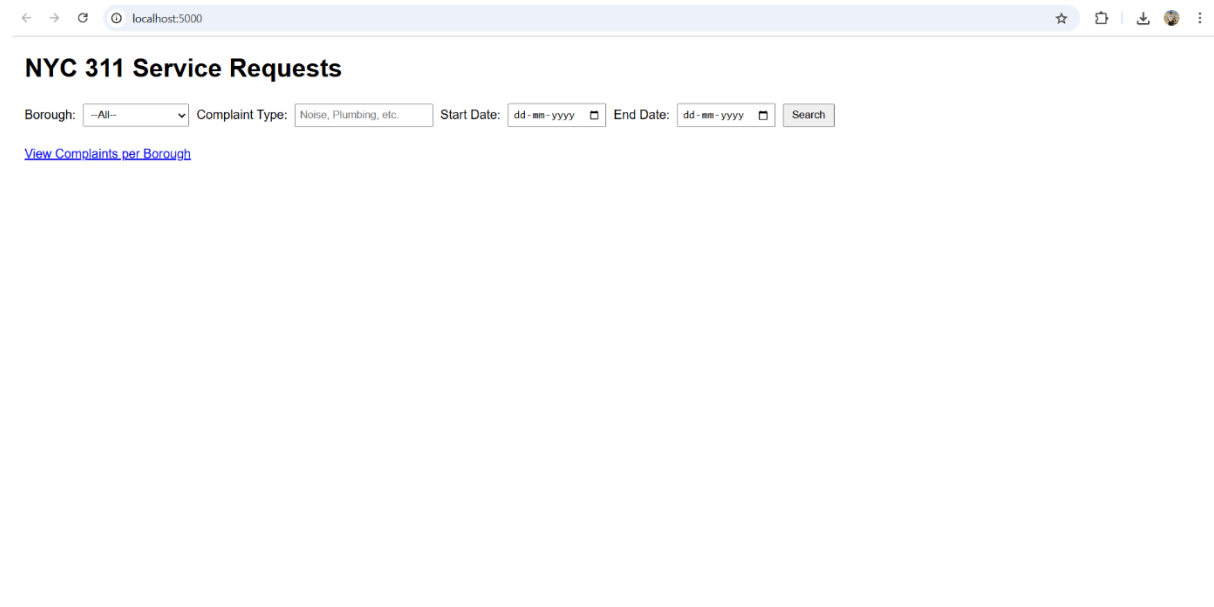
mysql> EXPLAIN SELECT * FROM service_requests WHERE complaint_type = 'Noise';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | service_requests | NULL | ref | idx_complaint_type | idx_complaint_type | 1023 | const | 8 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> EXPLAIN SELECT * FROM service_requests
-> WHERE created_datetime BETWEEN '2025-01-01' AND '2025-01-31';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | service_requests | NULL | range | idx_created_datetime | idx_created_datetime | 6 | NULL | 1 | 100.00 | Using index condition |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql>
```

Step 4 – Web Application:

This is a screenshot of the aggregate page of the web application, which contains the number of complaints in each borough. It confirms that the data that was loaded using ETL is available in the web interface and the aggregation query itself functions as expected.



A search form of BROOKLYN and Complaint of Plumbing from 01-01-2025 to 31-01-2025:



Results from MySQL:

Search Results - NYC 311

localhost:5000/search

Search Results

Request ID	Created Date	Closed Date	Agency	Agency Name	Complaint Type	Descriptor	Borough	City	Latitude	Longitude	Status	Resolution Description
63949423	2025-01-31 17:51:38	2025-03-13 06:23:24	HPD	Department of Housing Preservation and Development	PLUMBING	BATHTUB/SHOWER	BROOKLYN	BROOKLYN	40.6240207212	-73.95605857507	Closed	An HPD Inspector was not able to gain access to inspect this complaint. The Inspector left a card at the time of the inspection and a letter was sent from HPD after the inspection attempt to encourage you to call to schedule an inspection. HPD did not receive a timely response from you and the complaint has been closed. Please submit a new service request with 311 if the condition still exists.
63947266	2025-01-31 17:51:38	2025-03-13 06:23:24	HPD	Department of Housing Preservation and Development	PLUMBING	BASIN/SINK	BROOKLYN	BROOKLYN	40.6240207212	-73.95605857507	Closed	An HPD Inspector was not able to gain access to inspect this complaint. The Inspector left a card at the time of the inspection and a letter was sent from HPD after the inspection attempt to encourage you to call to schedule an inspection. HPD did not receive a timely response from you and the complaint has been closed. Please submit a new service request with 311 if the condition still exists.
												An HPD Inspector was not

An aggregate view (e.g., complaints per borough):

Complaints per Borough - NYC

localhost:5000/aggregate

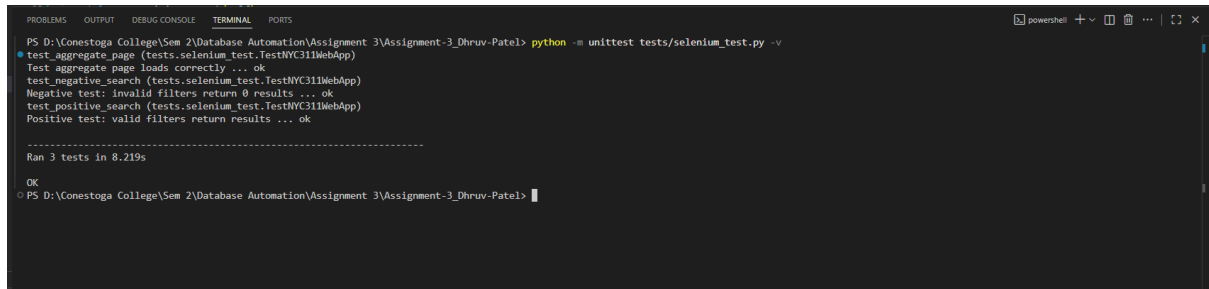
Complaints per Borough

Borough	Number of Complaints
BRONX	290
BROOKLYN	472
MANHATTAN	313
QUEENS	380
STATEN ISLAND	44
UNKNOWN	1

[Back to Search](#)

Step 5 – Automated Testing

This screenshot shows the automated testing of Selenium and pytest. It verifies the web application pages have loaded properly and that the expected items, e.g. tables, actually exist. This makes the functionality of the web app right.

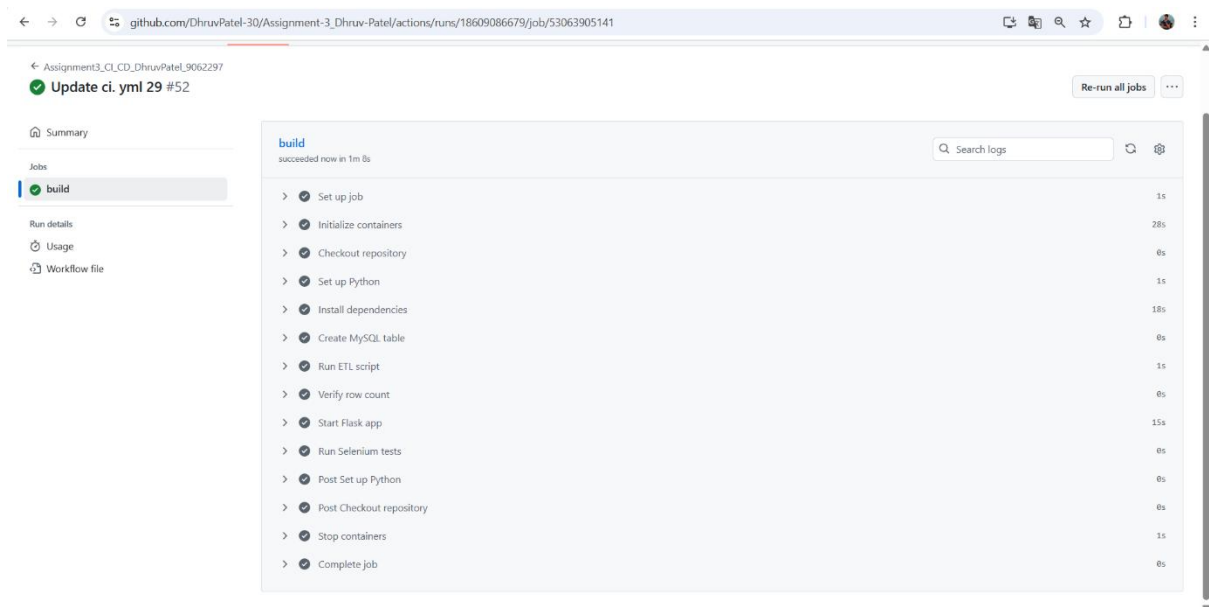


```
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel> python -m unittest tests\selenium_test.py -v
test_aggregate_page (tests.selenium_test.TestNYC311WebApp)
Test aggregate page loads correctly ... ok
test_negative_search (tests.selenium_test.TestNYC311WebApp)
Negative test: invalid filters return 0 results ... ok
test_positive_search (tests.selenium_test.TestNYC311WebApp)
Positive test: valid filters return results ... ok

-----
Ran 3 tests in 8.219s

OK
PS D:\Conestoga College\Sem 2\Database Automation\Assignment 3\Assignment-3_Dhruv-Patel>
```

Step 6 – CI/CD (GitHub Actions)



Reflections:

- Docker Compose made setting up of the environment easier. The restriction of dependencies and versions served to ensure that the ETL, database, and web app were compatible and did not fail in various machines. One of the problems was to learn the right service connection between app and db containers.
- A chunked ETL pipeline was written to deal with large amounts of data. Ensuring correct conversion of the type and correcting missing or bad data was also a challenge especially date and numeric columns.
- The choice of appropriate data types and a primary key of the service requests table allowed quick inserts and simple access to the results. One of the difficulties was to deal with duplicate rows safely with on duplicate KEY UPDATE.

- The flow of the application was confirmed by doing automated testing with Selenium and pytest. One obstacle was ensuring that the tests were able to locate elements in a reliable way and also running the tests in the headless mode in CI/CD pipelines.

Repo link: https://github.com/DhruvPatel-30/Assignment-3_Dhruv-Patel/