Taxi Trip Analytics Dashboard of New York City.

Team 5:

1. ***Reine Ella Dusenayo***
2. ***Alain Christian Mugenga***
3. ***Nina Cindy Bwiza***
4. ***Aurele Karega.***

ETL, database design, backend API project is a fullstack data analytics project.
data mining, and frontend visualization to reveal the New York insights into urban mobility.
Taxi Trip dataset.

1. Problem Framing and Analysis of the Dataset.

We are a group of four data and software practitioners who decided to create a full-fledged one system which displays mobility dynamics in New York City with the help of the official Taxi Trip dataset.
The data consists of time, places, fares, tips, and trip data. It captures millions of personal travels, giving a great chance of discovering the way cities move and the effect of trip characteristics on cost and time.
In data analysis, we also realized that there were various issues such as missing coordinates,trips recorded twice, and inaccurate records of fares. There were some records of zero distances or very high fares, which we considered to be outliers. We used normalization methods to provide a uniform format of timestamps and units. We also obtained novel features will include such factors as trip speed (km/h), cost per kilometer, and average days of idle time to augment our analysis among the unexpected observations made was that some of the short-distance trips possessed abnormally high fares, which are usually associated with airport pickups. This observation influenced the manner in which we divided trips and also persuaded us to introduce a fare-per-kilometer option.

2. Decisions System Architecture and Design.

An architecture that we used was the three-tier structure, which included the frontend, the backend and the database layers. The backend, which was developed with the help of Flask (Python), was dealing with ETLs was used as API gateway between the database and the dashboard. The database (SQLite, but supports PostgreSQL) stored normalized and cleaned data in a relational schema. The frontend, which was created with HTML,CSS and JavaScript, enabled interaction exploration of the data.Simplicity, portability, and clarity were the factors that influenced our stack decisions. Flask provided a Minimalistic yet powerful API layer, and SQLite allowed fast access to data in the course of development. All the databases were normalized to reduce redundancy maintain integrity. The trips were modeled as records and there were obvious links among them fares, locations, and times.

Trade-offs were made in the size of the dataset (limited due to performance reasons) and in client-side visualizing (rendering) minimized backend load at the cost of browser. computation.

3. Computer Science Data Structures and Algorithms.

We have used a simple ranking algorithm as a part of our system which is done manually. the 10 pickup sites sorted by frequency. We do not use inbuilt sorting features, but instead implement sorting created a home brewed counting and ranking system based on a dictionary structure.

Pseudo-code:

```
to record in trips: record = record[pickup-zone] otherwise: counts[zone] = 1
else: counts[zone] = counts[zone] + 1 sorted out zones = manual sort(counts) return top 10(sorted out zones)
```

Time complex 2 O (n 2) (because of manual sorting). Complexity of space: O(n) (to store). counts).

It is a method that makes our knowledge of the workings of algorithms and their processing in the real world deeper.

distribution and performance limitations.

4. Inspiration and Analysis.

Having cleaned and analyzed the dataset, we came up with the following major insights:

1 Trips between 6pm to 9 PM recorded the greatest volume, indicating evening commute patterns.

2 The average fare grew towards distance, whereas tip behavior was very different across payment type.

3 Majority of trips were less than 2 km, which meant that movements between neighborhoods were high micro-transit trends.

We visualized these insights in regard to bar charts and scatter plots on the dashboard. Users can have data filtered by time, fare, or trip length to investigate the various urban dynamics interactively.

5. Reflection and Future Work

Working in a team of four people gave us the opportunity to take advantage of different strengths-data UI development, API design and cleaning. One of our major problems was the control of big data effectively within the time constraints. We too had a few inconsistencies between frontend and backend endpoints which we addressed by improved schema documentation.

In the event of our continuation to this project, the system would be deployed in Docker containers to be scalable and to introduce machine learning models on trip duration and fare. We also imagine the dashboard with heatmaps of taxi traffic in various locations. boroughs and times.In general, this project enhanced our full stack data engineering and teamwork expertise. It and further increased our enjoyment of the ways in which data systems can narrate stories of urban life.