Objective:

The goal of this project is to share real-time information about citi bikes in NYC and also get historical trip data for analyzing trips and improving user experience for customers.

Motivation:

For someone who enjoys all kinds of sports and outdoor activities and is a travel enthusiast, I am always on the lookout for activities that are easily accessible. Having this problem statement in mind I wanted to capture and publish data on bike-sharing apps across NYC as it is a city always bustling with locals and tourists alike.

Scope:

The analysis has 2 data sources

- **Part I: Real-time data**
    - **Data:** I am using GBFS real-time feed from Citi Bikes NYC. GBFS stands for Global Bikeshare Feed Specification. It is a global standard for sharing system information about various bike stations across a country, city, and region along with details like the number of docks, availability of bikes at a given timestamp, station id, region id, plans and pricing
    - **Objective:** Collect real-time data for designing the backend of the app for getting the dimensions like
        - Regions:
        - Station_status:
        - Free_bike_status:
        - Ebikes_at_station:

The GBFS standard has the following feeds for real-time system information

- The over-arching feed has an array of key: value pairs of the name of the feed and the JSON file for the feed
- **name: ebikes_at_stations**,
  Url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/ebikes_at_stations.json

- **name: system_information**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/system_information.json

- **name: station_information**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/station_information.json

- **name: station_status**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/station_status.json

- **name: free_bike_status**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/free_bike_status.json

- **name: system_regions**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/system_regions.json

- **name: system_pricing_plans**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/system_pricing_plans.json

- **name: system_alerts**,
  url: https://gbfs.lyft.com/gbfs/1.1/bkn/en/system_alerts.json

- **Part II: Historical trip data**
  - **Data:** Citi bikes also shares downloadable CSV files of historical trips which goes back to 2013 when they started the bike-share service and started collecting data.
    Historical Citibike trip data : https://citibikenyc.com/system-data

  - **Objective:** Analyze user activity across all stations to find out interesting insights like
    - What is the highest trip distance taken?
    - Avg trip per day/week/month? Depending on who is the audience for the visualization
    - What time of the day were the trips taken
    - What are the most popular stations?

Tech Stack

Real-time data analysis

- Apache Flink for capturing real-time feed from bike stations like station id, number of docks, bikes available, bikes booked and bikes disabled and partition by region id
- Store into relational database to maintain ACID properties as we don't want to show a booked bike as available
- Move to Apache Iceberg for analytics

Historical Data Analysis and Visualization

- Convert csv to Parquet files in Apache Iceberg as
  - Parquet compresses the data and Iceberg offers pointers to last updated file which saves full file scan and time travel
- Aggregate data for weekly and monthly analysis of rides, popular stations, average distance traveled
- Create Dash plots for interactive visualization

References:
Historical Citibike trip data : https://citibikenyc.com/system-data
Realtime citi bike system data published as GBFS format. GBFS feed linked here: https://gbfs.citibikenyc.com/gbfs/2.3/gbfs.json