TRINITY COLLEGE DUBLIN

Week 4 Assignment
Name: Frason Francis

CS7NS4 URBAN COMPUTING APPLICATION
Student-ID: 23333313

CycleCast

Real-Time Bike Updates

Real Time Dataset:
https://data.smartdublin.ie/dataset/dublinbikes-api

TASK - 1 Data fusion and analysis data

Motivation:

My project, Cyclecast, is aimed at providing a real-time, interactive platform for displaying geographical information, with a special focus on bicycle data. It leverages the capabilities of Flask and Folium to offer advanced web-based mapping solutions. The essence of Cyclecast is to fetch and dynamically present data about bicycle locations and availability, enhancing the accessibility and user engagement with spatial data. This project is driven by the goal of improving urban mobility and planning, making it easier for users to access up-to-date information on bicycle use and infrastructure in their city.

The algorithm used for Data processing:

1. Data Retrieval: Fetching real-time bicycle data from an API. This involves making web requests to an external API to gather bicycle location and availability data.

```
import requests

def fetch_bicycle_data(api_url):
    response = requests.get(api_url)
    return response.json()
```

2. Data Processing: Transforming the fetched data into a suitable format for visualization. This may include parsing JSON data and extracting relevant information like coordinates and status of bicycles.

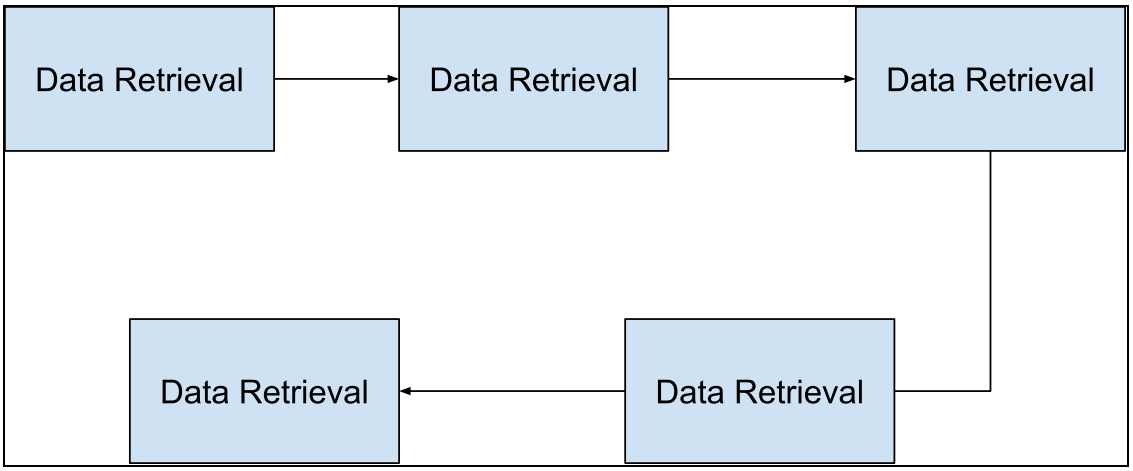
```
def process_data(data):
    processed_data = [{"latitude": item["latitude"], "longitude": item["longitude"],
"status": item["status"]} for item in data]
    return processed_data
```

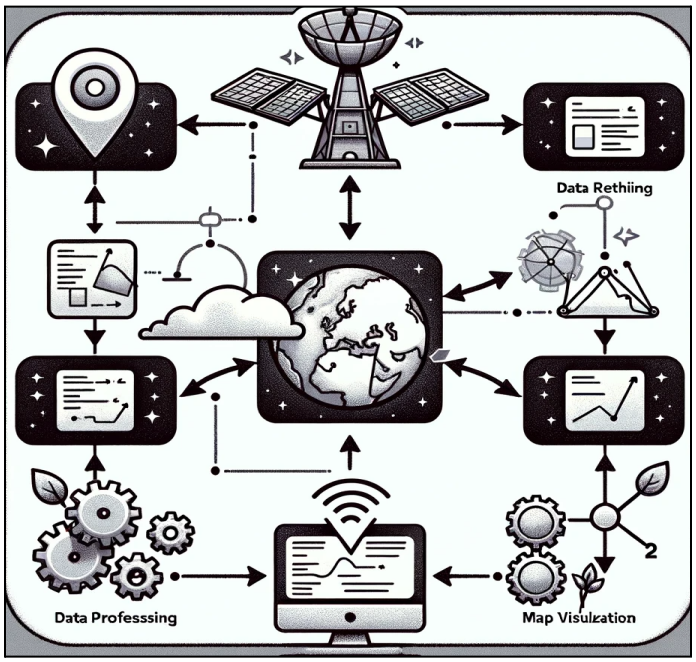
3. Map Visualization: Using Folium to create interactive maps that display the bicycle data. This includes plotting points on the map based on the bicycle locations and adding interactive elements.

```
import folium

def create_map(data):
    map = folium.Map(location=[default_latitude, default_longitude], zoom_start=12)
    for item in data:
        folium.Marker([item['latitude'], item['longitude']], popup=f"Status:
{item['status']}").add_to(map)
    return map
```

CycleCast Project Overview





The overview diagram represents the key components of the Cyclecast project:

1. Data Retrieval: The algorithm for fetching real-time bicycle data from an API.
2. Data Processing: Involves transforming the retrieved data into a format suitable for visualization.
3. Map Visualization: Utilizes Folium to create interactive maps displaying the bicycle data.
4. Flask (Web Framework): The backbone of the web application, managing data flow and user interactions.
5. Folium (Mapping): The tool used for generating interactive maps.

Here's a breakdown of the key components used in Data Fusion and analysis of algorithm prior to making a webapp:

1. Data Collection (collect_data): This function regularly fetches data from the JCDecaux API for a specific city (Lyon) and stores it in a MongoDB database. It uses a timer to repeat this process periodically, ensuring continuous data collection.
2. Data Transformation (convertHeure): Converts timestamp data into a readable date-time format.
3. Data Visualization:
    a. Station Statistics (Velib_stat): Aggregates data from the database for a specific station, calculates bike and stand availability, and visualizes this information as a pie chart.
    b. Station GPS Positions (position_gps_station): Extracts and plots the geographical locations (latitude and longitude) of all bicycle stations.
    c. Histograms (histo_velo_dispo): Creates bar charts showing the availability of bikes and stands at various stations at a specified hour.
4. Distance Calculation (distance): Computes the geographical distance between two points, useful in determining the nearest station.
5. Predictive Analysis (prediction_velo): This function predicts the availability of bikes or stands at various stations based on the time of day. It sorts stations by bike/stand availability and calculates their distance from a given location, aiding users in finding the nearest station with available bikes or stands.

TASK 2 – Data visualization and Actuation

The various methods employed for data visualization and actuation are:

1. Visualization of Station Statistics: The Velib_stat function compiles and visualizes data (available bikes and stands) for specific stations using pie charts, offering an intuitive view of each station's status.
2. Station GPS Mapping: The position_gps_station function plots all bike stations' GPS coordinates on a map, helping users visually locate stations city-wide.
3. Availability Analysis via Histograms: histo_velo_dispo creates bar charts for bike and stand availability at different stations, aiding in identifying usage patterns.
4. Predictive Analysis for User Convenience: prediction_velo predicts bike or stand availability, enhancing user experience by guiding them to likely available stations.

The, actuation in Cyclecast depends on sensors attached to bikes. When a bike station has few bikes available, it affects the total value, impacting the overall station ranking. This dynamic response contributes significantly to the system's actuation, reflecting real-time usage and availability trends.

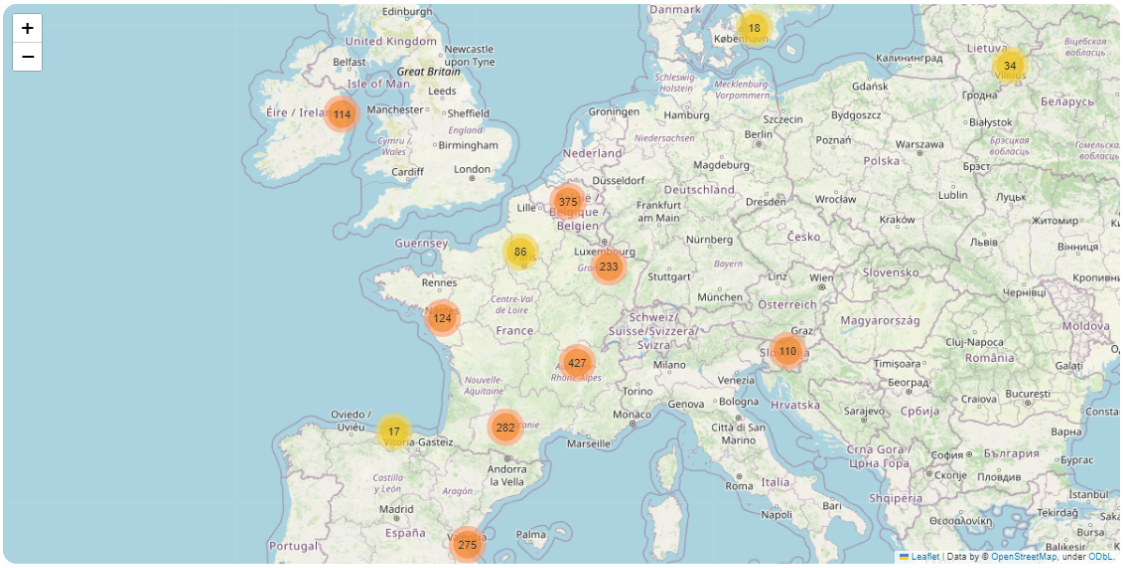
Technical challenges faced while creating the Cyclecast project includes:

1. **Data Integration:** Harmonizing data from various sources, especially in real-time, poses challenges in terms of consistency, format, and reliability.
2. **Scalability**: Ensuring the system efficiently handles increasing amounts of data and user queries as it scales up.
3. **Accuracy in Predictive Models**: Developing predictive algorithms that accurately reflect real-world scenarios can be complex, especially with changing patterns in urban mobility.
4. **Real-Time Data Processing:** Implementing effective real-time data processing and visualization demands high-performance computing and efficient algorithms.
5. **Sensor Data Reliability:** Relying on sensor data introduces challenges in ensuring accuracy and dealing with potential hardware malfunctions or data transmission issues.

CYCLECAST

RANKING

Bike station Map



Amiens | Besancon | Brisbane | Bruxelles | Cergy-pontoise | Creteil | Dublin

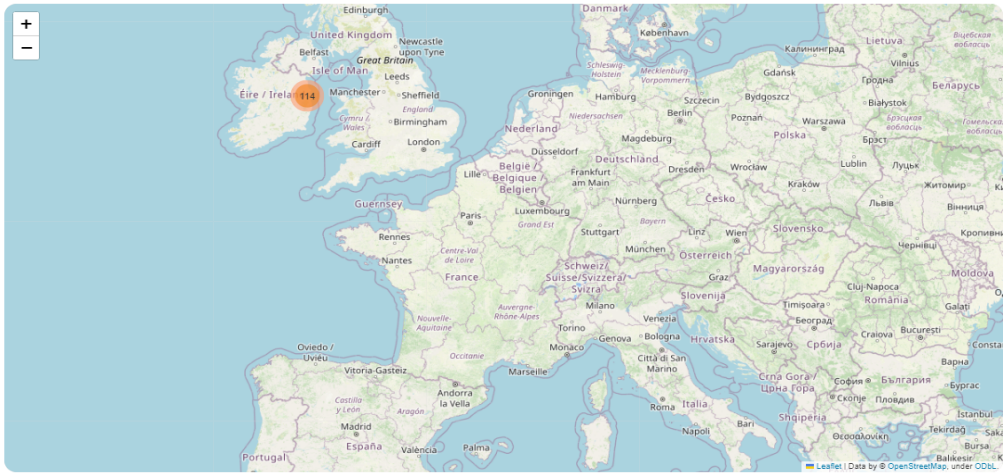
Jcdecauxbike | Lillestrom | Ljubljana | Lund | Luxembourg | Lyon | Maribor

Fig1. Main Page Dashboard

CYCLECAST

RANKING

City of Dublin



CLARENDON ROW | BLESSINGTON STREET | BOLTON STREET | GREEK STREET | CHARLEMONT PLACE | CHRISTCHURCH PLACE

HIGH STREET | CUSTOM HOUSE QUAY | EXCHEQUER STREET | DAME STREET | EARLSFORT TERRACE | ECCLES STREET

FITZWILLIAM SQUARE WEST | FOWNES STREET UPPER | HARDWICKE STREET | GEORGES QUAY | GOLDEN LANE | GRANTHAM STREET

Fig2. Filtering out the relevant city (“Dublin”)

City of Dublin

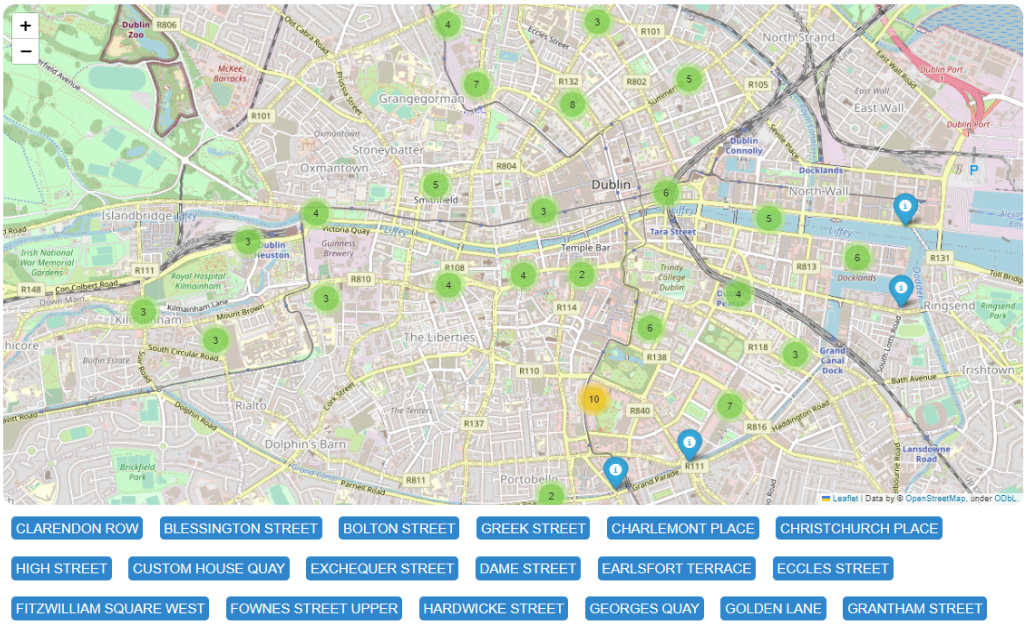


Fig3. After clicking Dublin city you can see all the available bike position on the map in real-time with the total number of of bike available

Station : LEINSTER STREET SOUTH (Dublin)

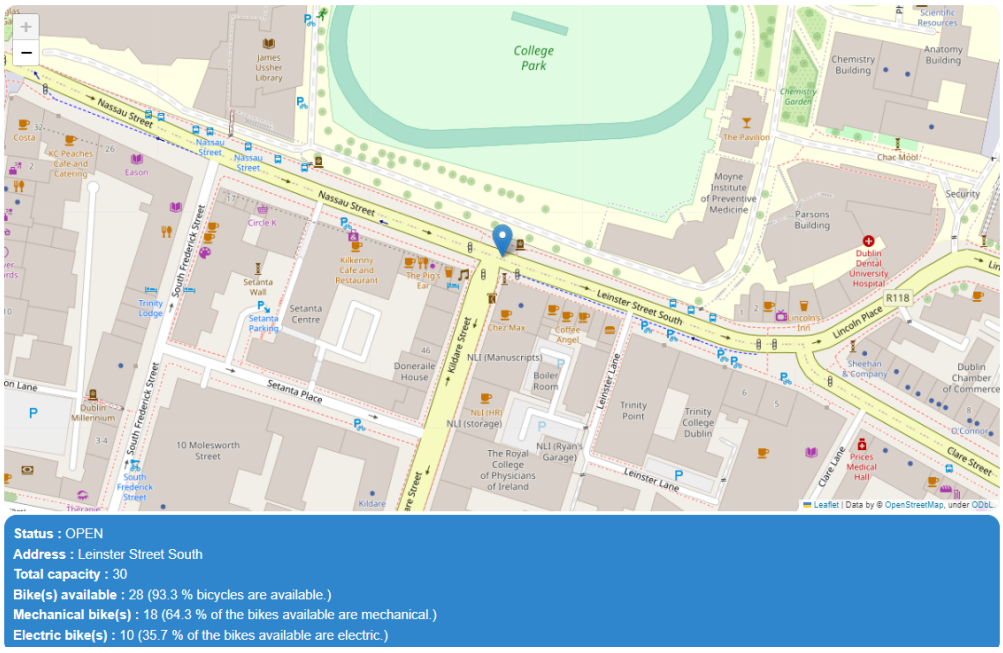


Fig 4. Illustrate a detailed view of the Bike Address location highlighting the Status, Address, Total capacity of the bike, and whether or not the bicycle is mechanical or electrical.

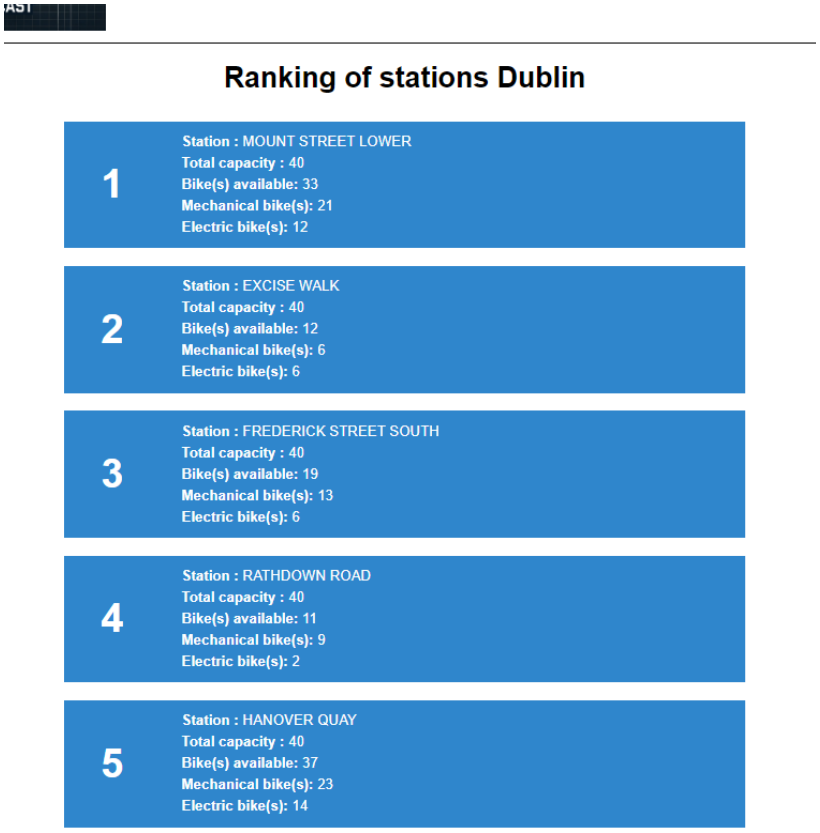


Fig 5. Depicts the Rank by descending order by the total capacity of bike available in the country according to the real time data gathered through Jcdecaux-API

These visualization techniques, combined with data analysis and predictive algorithms, make the Cyclecast project a comprehensive tool for urban mobility, especially for bicycle-sharing systems.

Conclusion & Future work:

The Cyclecast project stands out as a significant advancement in improving urban mobility, particularly within bicycle-sharing systems. Its core strength lies in the effective use of real-time data, coupled with intuitive data visualization and predictive analytics, which collectively enhance the user experience. The integration of bicycle sensors further enriches the system, adding a dynamic and responsive element that reflects actual usage patterns.

Looking ahead, the project could benefit from incorporating additional data sources like traffic and weather conditions to refine its predictive capabilities. The creation of a dedicated mobile application would make the service more accessible. Delving into machine learning algorithms could provide deeper insights and predictions,

while expanding the model to encompass other urban transportation modes could broaden the project's utility and impact.

References:

https://developer.jcdecaux.com/#/opendata/vls?page=dynamic

https://www.jcdecaux.ie/republic-of-ireland/digital

https://www.youtube.com/watch?v=eqL_VPknjes

https://mateuszwiza.medium.com/plotting-api-results-on-a-map-using-flask-and-leafletjs-2cf2d3cc660b