

Case: Sensor data dashboard for city officials

Role Introduction

You work as a back-end developer at Coding the Curbs. Together with the CTC team, you are responsible for developing robust, scalable, and efficient back-end systems that power our Smart Zone solutions. As a back-end developer, your role involves handling and processing large datasets, building secure APIs, and ensuring the smooth operation of our software infrastructure. Your work directly contributes to enabling smarter city logistics and seamless user experiences.

Project Introduction

Recently, Coding the Curbs has been collecting valuable data from Smart Zones through a network of sensors. These sensors record parking activity in real time, providing insights into how loading and unloading zones are utilized. This data is crucial for city officials to evaluate the performance of Smart Zones, make informed decisions about infrastructure improvements, and ensure efficient urban logistics.

For this assessment, you will work with real data collected from **a single Smart Zone location equipped with four sensors**, each placed **5.5 meters apart** along the zone. The dataset is from 1 month. The sensors monitor vehicle presence. An **attached map illustration** shows the location of the sensors within the Smart Zone and their corresponding IDs.





The dataset includes the following fields:

- **Status:** Indicates whether a vehicle is parked (1) or not (0).
- **Timestamp:** The exact date and time of the status change.
- **Sensor ID:** Identifies the sensor's location as shown on the map.

Note: If the sensor's status remains unchanged for approximately **4 hours**, it will repeat the signal to confirm its current state.

Assignment

Your task is to process the provided dataset (attached) and create a simple dashboard for city officials that visualizes this data. The dashboard should provide actionable insights and help stakeholders understand how their Smart Zones are being used.

Please meet the following requirements:

1. Data Processing

1. Import the dataset into a relational database (SQL).
2. Write SQL queries / Python to calculate meaningful insights, such as:
 - Total number of parking activities.
 - Average parking duration.
 - Occupancy rate over a defined time period.

2. Dashboard Creation

Create a simple dashboard, designed for city officials, that:

- Includes **at least one graph** (e.g., count of parking activities per hour, occupancy rate over time, or sensor-specific activity).
- Displays **at least three calculated metrics** (e.g., average duration, total parking activities, peak parking times).
- Is built using **HTML, CSS, and JavaScript**.

3. Documentation

Provide a brief document or README explaining:

- The functionality of your dashboard (what it does and how it works).
- The decisions you made during the development process (e.g., why specific calculations, metrics, or visualizations were chosen).
- Any limitations or assumptions you identified while working with the dataset.

Assessment Focus

You will **not** be assessed on the look & feel or design quality of the dashboard. Instead, you will be evaluated on the following:

1. **Data Processing and Analysis**
 - Accuracy and efficiency of your SQL queries or Python scripts.
 - Logical database design to effectively handle the dataset.
 - Creativity your calculations, demonstrating an understanding of the dataset's potential for Smart Zone performance evaluation.
2. **Code Quality**
 - Clarity, maintainability, and organization of your codebase.
 - Proper use of functions, comments, and adherence to best practices in SQL, Python, or scripting languages.
3. **Relevance for City Officials**
 - Ability to transform raw data into actionable insights.
 - Appropriateness of metrics and visualizations for evaluating Smart Zone performance.

Additional Information

- Handle the dataset with care and do not share it externally.
- You may use open-source libraries and tools to enhance functionality, but ensure the core logic is your own.
- This is a technical assessment, and your problem-solving skills will be the primary focus.
- Try to spend no longer than 5 hours on this assignment. It should be brief & efficient.

Good luck with the assignment!