IoT Log Parser Software Assessment

# Introduction

At Smart City Living Lab, we work across hardware, software, and their intersection. A core requirement for a smart city is a platform that collects data from diverse hardware devices and sensors, making it accessible for analysis and visualization. This is what makes a Smart City truly "smart."

# Problem Statement

Your task is to build a robust IoT log parser. You will be provided with a log file containing data from various sensors and devices, potentially including Base64 encoded images and web server logs. Your parser must extract, structure, and potentially visualize this data efficiently and accurately.

## Specification 1: Data Extraction and Structuring

A core part of this assessment involves building a parser that can:

* **Extract Data:** Accurately extract all key-value pairs from each log entry, handling various data types (string, integer, float, Boolean) and managing missing values ("null") gracefully. Example: Temperature=28.92 should extract the key "Temperature" and the float value 28.92.
* **Structure Data:** Organize the extracted data into a well-defined, easily accessible structure, such as a list of dictionaries or a Pandas DataFrame. Ensure correct data type assignment for each extracted value.
* **Base64 Image Decoding:** If the log file contains Base64 encoded images, decode them. Points will be awarded for displaying or saving the images.
* **Data Visualization:** Present the parsed data using clear and informative charts, graphs, or tables.
* **Web Server Log Parsing:** If provided with Apache/Nginx web server log files, extract key metrics such as request counts, response codes, and access times.
* All the above should be incorporated into a comprehensive and well-designed dashboard.

## Specification 2: Error Handling

Real-world logs often contain errors or unexpected formats. Your parser must:

* **Handle Errors Robustly:** Gracefully handle invalid log entries or unexpected formats without crashing. Skip or report these errors without interrupting the overall parsing process.
* **Provide Informative Error Messages:** Output helpful error messages that pinpoint the type and location of parsing errors, aiding in debugging.

## Report (README.md)

Document your project thoroughly in a README.md file within your GitHub repository. Include the following for each specification:

* **Installation Instructions/Setup:** Detailed steps to set up and run your parser, including dependencies.
* **How to Run:** Clear instructions on executing the parser.
* **Screenshots:** Illustrative screenshots of your parser in action and any visualizations.
* **Assumptions:** Document any assumptions you made during development.

# Bonus: Performance Analysis

Analyze the performance of your parser. Include metrics like processing time for various log file sizes and, if applicable, the time taken from data ingestion to visualization.

## Additional Points

* While Python or JavaScript are suggested, you are free to use any language or framework. However, choosing Python or JavaScript will keep you in line with the structure followed at Smart City Living Lab. Your choice of Language will have no effect on the evaluations.
* Document any use of AI tools in your report.

## Submission Guidelines

1. **GitHub Repository:** Create a **public** GitHub repository and push your complete code to the main branch.
2. **README.md:** Include a comprehensive README.md file as described above.
3. **Submit the Link:** Share the URL of your public GitHub repository. No zipped code or email submissions will be accepted.