

## Tech-Lead - Software engineer (mid) assignment

**Time** – 5 Days.

### Deliverables

1. You should submit efficient, clean, and sufficiently commented code. Your submission should be hosted on a **private git repository** and invite the reviewers (they will be reported through an e-mail) to your personal repository. Please include a Read.me file with any necessary documentation
2. You should submit a presentation in PowerPoint where you present at a high-level your solution's design. This presentation will be used to present your solution during the interview following the assignment. You should also submit a **report** (in pdf - 500 words) describing the process followed and key results.  
*Note: The presentation will be used to present your solution during the interview following the assignment.*

### Requirements

#### Part A

In this assignment, you will need to design and implement a backend solution for a meteorological web application. In this application there are several data sources:

- Every 10 seconds, IoT devices installed in meteorological stations in each city across the county measure properties such as temperature, humidity, and wind, and stream them to the backend.
- Every morning, each meteorological station in each city collects all the data from the last 24 hours and sends them to the backend as a batch
- A user uses daily the UI and submits a form with the weather forecast for tomorrow for each city regarding temperature, humidity, and wind

The backend should be able to receive all data and store them. It should also be able to provide the data to the UI upon request.

- a) *Please design (**NOT implement**) the backend solution of this application and describe the functionality of the individual components that you have used justifying your choices.*

#### Part B

In this part, you will need to implement part of the Backend solution of part A, given that:

1. The meteorological stations have the following properties:
  - Code
  - City
  - Latitude
  - Longitude
  - Date of installation
2. Sensors have the following properties:
  - a. Sensor Id
  - b. Code of the Meteorological Station they belong to
  - c. Property of measurement (temperature, humidity or wind)
3. The form submitted by the user through the UI contains the following fields:
  - a. Forecast date
  - b. City
  - c. Temperature
  - d. Humidity
  - e. Wind
4. The measurements from the IoT devices follow the same JSON structure. A sample of the measurements can be seen here:

```

{
  "measurement_id": "88d58a78-95d3-4b9c-b0a4-aab2e2ed73d5",
  "sensor_id": "HUM-001",
  "date": "2024-05-28T15:22:18+02:00",
  "station": "MET-004",
  "info": {
    "category": "Humidity",
    "measurement": 56,
    "unit": "Percentage"
  }
}

{
  "measurement_id": "435k65l345-95d3-4b9c-b0a4-34fsd234fs",
  "sensor_id": "TEM-456",
  "date": "2024-05-28T15:22:18+02:00",
  "station": "MET-009",
  "info": {
    "category": "Temperature",
    "measurement": 56,
    "unit": "Celsius"
  }
}

{
  "measurement_id": "9343h43-95d3-4b9c-b0a4-4543gs546gd",
  "sensor_id": "WND-245",
  "date": "2024-05-28T15:22:18+02:00",
  "station": "MET-020",
  "info": {
    "category": "WIND",
    "measurement": 11,
    "unit": "m/s"
  }
}

```

5. The user should be able to create, edit and delete a meteorological station. The user should also be able to ask and see in the UI the real (average) and the forecasted meteorological properties of a particular date
  - a) Please implement the backend solution that:
    - a. is able to receive data from the IoT sensors asynchronously and from the UI and store them. Please **ignore** any data coming as a batch
    - b. is able to send the user any data they ask.
  - b) Dockerize the services to be runnable independently of the platform
  - c) Mention any future additions that could improve the system

**Guidelines**

All code should be written in Python (Flask, Fast or Django frameworks are accepted)

You should only implement the backend part. No UI is requested

API documentation is required

Testing is not required but it will be appreciated

Good luck!