Test Case Backend Developer

Test Case: Sensor Event Processing Microservice

Overview

The candidate is required to develop a Java micro service that interfaces with RESTful endpoints to receive sensor data, stores the data in MongoDB, and optionally integrates with Kafka for event streaming. The project should be committed to a public GitHub repository to demonstrate the candidate's proficiency with version control and project documentation.

Objectives

- Implement RESTful endpoints to handle sensor data.
- Integrate MongoDB for data persistence.
- Optionally, integrate Kafka to produce and consume events.
- Commit the project to a GitHub repository with proper documentation.

Technical Requirements

Micro service Framework

- Utilise Spring Boot for creating the micro service.
- Configure the application to run on localhost, port 8090.

RESTful Endpoint Implementation

- Develop a POST endpoint `/api/sensor-events` for receiving sensor event data in JSON format, such as:

```
{
  "sensorId": "sensor_123",
  "timestamp": "2024-02-15T12:34:56Z",
  "type": "temperature",
```

```
  "value": 25.3
}
```

- Ensure proper validation for incoming data.

MongoDB Integration

- Use Spring Data MongoDB to facilitate MongoDB interactions.
- Store the sensor event data in a collection named `sensor_events`.

Kafka Integration (Optional)

- Implement Kafka producer configuration to send events to a topic named `sensor_events`.
- Implement a Kafka consumer in the same or a separate microservice to consume events from the `sensor_events` topic and store them in a collection named `processed_sensor_events`.

GitHub Repository

- Initialize a Git repository for your project and commit your code in logical increments with clear, descriptive messages.
- Push the code to a new, public GitHub repository under your personal account.
- Ensure the repository contains a `.gitignore` file suitable for a Java project.

Documentation

- Provide a README.md file in the root of the repository with the following:
  - Instructions for setting up and running the application, including MongoDB and Kafka (if used).
  - Detailed API endpoint documentation.
  - A brief overview of the project architecture and design choices.
  - Any necessary information on configuring Kafka and MongoDB.