36Zero Vision Automation Engineer Take-Home Assignment

Overview

In this assignment, you will demonstrate your ability to parse different data formats (JSON and XML), process the data according to specific conditions, and integrate with various systems including REST APIs, Industrial PCs (IPC), and OPC UA. You'll also have the opportunity to showcase your skills with low-code platforms (optional).

Tasks

Task 1: JSON Processing and REST API Integration

Requirements:

1. Parse the provided JSON files (1.json and 2.json)
2. Analyze the 'fi_confidence' values:
   - Count instances where fi_confidence > 0.99
   - Make a POST request to an API endpoint with the count
3. Extract and process 'tr_values':
   - Extract all tr_values
   - Use these values as serial IDs in REST API calls
4. Implement proper error handling and logging
5. Create a modular and maintainable solution

Task 2: IPC Integration with Threshold Detection

Requirements:

1. Parse 3.json file
2. Monitor ‘numberMinDetectedObjects’ field
3. When threshold ≥ 2, send DIO signal to IPC
4. Implement error handling and logging
5. Create a configuration system for IPC settings
6. Implement reconnection logic for IPC communication

Task 3: OPC UA Integration with XML Processing

Requirements:

1. Parse the provided XML file (4.xml file provided)
2. Extract Fault tag and the severity of it and if available, go to step 3
3. Send signals to OPC UA server
4. Implement proper error handling and reconnection logic

Additional Requirements

Low-Code Platform Integration (Optional)

- You may use platforms like Node-RED or n8n to implement parts of the solution
- If using a low-code platform, provide:
  - Flow documentation
  - Export of the flow configuration
  - Instructions for deployment

Documentation Requirements

1. README.md with:
   - Setup instructions
   - Dependencies list
   - Configuration parameters
   - Example usage
2. Comments in code explaining logic
3. Error handling
4. API documentation if creating additional endpoints

Evaluation Criteria

1. Code Quality
2. Documentation
3. Functionality
4. Extra Credit:
   - Low-code implementation
   - Unit tests
   - Docker containerization

Submission Instructions

1. Create a private GitHub repository
2. Include all source code and documentation
3. Provide access to repository when submitting
4. Include any additional setup instructions
5. Or you can also create zip file and send it to us.

Good luck! If you have any questions, please don’t hesitate to ask.