**Algorithm: IoT-Based Data Retrieval using Arduino and DynamoDB (Conventional Method)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1: | **Inputs:** | | | |  | **Note** |
| 2: |  | SensorData: [Temperature, Humidity] | | | |  |
| 3: |  | Timestamp: Date and Time of reading | | | |
| 4: |  | Δt: Time interval between readings (1 minute) | | | |
| 5: |  | T\_start: Start time for data retrieval | | | |
| 6: |  | T\_end: End time for data retrieval | | | |
| 7: | **Outputs:** | | | |  |
| 8: |  | RetrievedData: Parsed records between T\_start and T\_end | | | |
| 9: |  | ExportedFile: Excel file containing filtered data | | | |
| 10: | **Start:** | | | |  | DHT\_IoT\_AWS.ino |
| 11: |  | 1. [Sensor Layer]- Repeats every Δt (1 minute) | | | |
| 12: |  | | a. Initialize sensors to measure Temperature and Humidity. | | |
| 13: |  | | b. Every Δt, read SensorData and current Timestamp. | | |
| 14: |  | 2. [Field Gateway - Arduino]- Repeats every Δt (1 minute) | | | |
| 15: |  | | a. Use Arduino IDE to send SensorData via MQTT to AWS IoT Core | | |
| 16: |  | | b. Create Payload = {"timestamp": "YYYY-MM-DD+HH:MM:SS",  "temperature": value,"humidity": value} | | |
| 17: |  | 3. [AWS Cloud - DynamoDB]- Repeats every Δt (1 minute) | | | | AWS Rules |
| 18: |  | | a. Store the Payload string in a DynamoDB table with attributes: | | |
| 19: |  | | | - Partition Key: Time | |
| 20: |  | | | - Other Attributes: Full data string | |
| 21: |  | 4. [Data Retrieval - Python Script]- (Run by user periodically or on-demand) | | | | Data\_Retrieval\_DynamoDB.py |
| 22: |  | | a. User inputs T\_start and T\_end. | | |
| 23: |  | | b. Calculate number of hours between T\_start and T\_end. | | |
| 24: |  | | c. Use boto3 to scan all items from DynamoDB. | | |
| 25: |  | | d. Parse the stored strings into structured fields: | | |
| 26: |  | | | - Split timestamp and extract [hour, minute, second] | |
| 27: |  | | | - Reconstruct datetime from split values | |
| 28: |  | | e. Filter records where T\_start ≤ Timestamp ≤ T\_end. | | |
| 29: |  | | f. Save filtered data as an Excel file (ExportedFile). | | |
| 30: | **End.** | | | |  |  |