Procedure explanation

The implementation of Task 1 ensures that IoT sensor data is cleaned, transformed, and stored in a database efficiently and reproducibly. The workflow consists of four main stages: data cleaning, ingestion, indexing, and containerization.

1. **Data cleaning**
   The raw IoT dataset, collected in CSV format, contains inconsistencies, including empty rows, extra columns, and irregular headers. To address this, a Python script (*clean_csv.py*) is used. It removes unnecessary entries, drops misaligned columns, and standardizes column names. The result is a cleaned CSV file (*cleaned_IoT_data.csv*), which becomes the input for ingestion.
2. **Data ingestion**
   The cleaned dataset is loaded into MongoDB in batches using the Python script (*ingest.py*). Instead of bulk-loading the entire file, which may cause performance issues, the script processes the data in configurable chunks. Numeric fields are converted into floats, boolean values are standardized, and timestamps are parsed into a consistent UTC format. Each record is assigned a unique identifier (*device* + *timestamp*), ensuring idempotency and preventing duplicates.
3. **Indexing**
   To optimize queries, indexes are created on frequently used fields with the script (*01-create-indexes.js*). Indexes on timestamp, device, and their combination improve query efficiency, making searches scalable as the dataset grows.
4. **Containerization**
   All components (MongoDB, Mongo Express, and the Python scripts) are deployed with Docker using (*docker-compose.yml*). This ensures portability and reproducibility, allowing anyone to run the setup in a consistent environment. Mongo Express provides a web interface for verifying and exploring the ingested data.

Result

This structured approach reliably handles IoT data from preprocessing to storage. It ensures data quality, efficient querying, and scalability for future extensions.

🔗 GitHub Repository: *[Insert your repository link here]*