Dear Muhammad, (Dost)
Your Phase 2 submission demonstrates a very good understanding of batch data ingestion and system design. The workflow is well-structured and clearly explained, covering the full process from raw data cleaning to containerized deployment using Docker.

1. Pipeline Design → Clear modular structure from cleaning to indexing and storage → Adding a flow diagram would make the process more visually understandable.

What has changed:

Created and added a new flow diagram.

2. Data Flow & Fault Tolerance → Handles data chunking and ensures idempotency using unique IDs → Consider including a retry or logging mechanism for failed insertions.

What has changed:

- Rewrote the ingest.py script with a robust retry mechanism for transient MongoDB write errors (up to 3 attempts).
- Added structured logging using Python's logging module, storing detailed logs both on-screen and in /logs/ingestion.log.
- Implemented lightweight metrics output (metrics.json) recording total rows, inserted count, duplicates, and duration.
- Improved idempotency logic by using SHA-1 hashes of device + timestamp for unique _id fields.
  Result: The data ingestion process is now fully fault-tolerant, transparent, and auditable.

3. Containerization → Proper use of Docker Compose for reproducibility and integration with Mongo Express → Add comments or instructions inside the docker-compose.yml for clarity.

What has changed:

Rewrote the docker-compose.yml file with comprehensive inline comments explaining each service (MongoDB, Mongo Express, and App).

4. Monitoring & Visualization → Mongo Express gives a simple GUI for checking data → Consider adding lightweight logging or metrics collection for better visibility.

What has changed:

- Added persistent logging and metrics tracking inside the app/logs/ directory.
- Logs capture chunk-by-chunk ingestion status, errors, and duplicate counts.
- A metrics.json file is automatically created summarizing ingestion performance (rows processed, duration, duplicates).

5. Documentation & GitHub → GitHub is clean with readable scripts and clear structure → Expanding the README with expected input/output samples would be helpful.

What has changed:

- Rewrote the **README.md** completely to a professional standard.
- Added **expanded input and output examples** (multiple CSV rows and MongoDB document samples).
- Included **clear explanations of each script's function** (clean_csv.py, ingest.py, 01-create-indexes.js).
- Added **sample log and metrics outputs** for transparency.
- Integrated **verification steps** for users to check results in Mongo Express.

GitHub repository: https://github.com/Dost-DS/task1_mongodb_batch_ingest.git

(Best regards,
Sahar)

Best,

Dost