Task: Employee Attendance Log Processing

Overview

Your company collects daily attendance logs from multiple devices as files. Each file contains raw punch records for employees. Your task is to process these files to generate a **clean summary report**, including **working hours per employee per day**, and save it in **Excel format**.

Files Provided

- Folder: `attendance_logs/` containing multiple log files (`.log` or `.csv`) ● Each file contains rows with the following columns:

`emp_code first_name last_name timestamp device`

- `timestamp` may be an **integer Unix timestamp**.
- Some rows may have missing or corrupted values.
- **Shift Period:** 09 AM - 6 PM
- **Late Entry:** Check-in after 09:30 AM
- **Early Exit:** Check-out before 05:00 PM

Task Requirements

1. File Handling

- Read **all log files** from the folder.
- Skip rows with **missing or invalid data** and log them in `error_log.txt`.
- Merge all data into a **single dataset**.
- Remove **duplicate records** (same `emp_code`, timestamp, device).

2. Data Processing

- For each employee, calculate per day:
    1. **First punch time**
    2. **Last punch time**
    3. **Total number of punches**
    4. **Working hours**
    5. **Late Entry(Yes/NO)**
    6. **Early Exit(Yes/No)**
- Identify employees who have **only one punch in a day**.
- Sort results efficiently by **employee code and date**.

3. Output

- Generate a **JSON summary**:

```
{
    "2025-09-24": [
        {"emp_code": "10023", "first_punch": "09:05", "last_punch": "18:10",
"total_punches": 4, "working_hours": "09:05", "late_entry":1, "early_exit":0},
        {"emp_code": "10024", "first_punch": "09:00", "last_punch": "17:55",
"total_punches": 3, "working_hours": "08:55", "late_entry":1, "early_exit":0}
    ]
}
```

- Generate an **Excel file** (attendance_summary.xlsx) with columns:

Date | Emp Code | First Punch | Last Punch | Total Punches | Working Hours| Late Entry| Early Exit

Bonus (Optional)

1. Writing code with better time complexity will be considered a plus.
2. Implement a **search function** to quickly retrieve summary by employee and date.
3. Support **large files** efficiently using generators or chunked processing.

Deliverables

1. Python script(s) implementing the requirements.
2. JSON summary report.
3. Excel summary file with working hours.
4. error_log.txt for skipped or invalid rows.

Evaluation Criteria

- Correctness of summary report and Excel file.
- Accurate working hours calculation.
- Efficient handling of large files.
- Proper use of **Python data structures and algorithms**.
- Clean, readable, and well-documented code.
- Bonus points for optional features.

Note:

1. **DEADLINE: the task must be completed within 2 days of assigning & the deadline mentioned in the email.**
2. Submit your code by providing the Git repository URL.
3. This task must be completed without any assistance from ChatGPT or other AI tools; otherwise, it will be disqualified.