Table Extraction Task: Replicating a Transaction Log

Welcome! Your first task is to develop an automated solution for extracting structured data from an image and transforming it to precisely match a provided output file. This task focuses on exact data extraction and formatting.

The Goal: Your primary objective is to **design and build a robust, reproducible pipeline** that can identify and extract **only the transaction-related data** from a document image, and format it to **exactly match a provided CSV file**. You are free to use any tools you find helpful, including LLMs like ChatGPT or Gemini, OCR tools like Google Vision AI or Tesseract, or any other programming libraries.

The Task:

1. **Image Input:** I will provide you with a single image file of a document that contains a lot of text, including a distinct section formatted as a table of transactions.
2. **Desired Output:** The desired output is a CSV file that **is an exact replica of the provided `csv_sample.csv` file**. Your solution must produce a file with the same number of rows, columns, and data values, formatted identically.
3. **The Challenge:** Your final submission should include the following:
    - **The Extracted Data:** Your final CSV file, which should be named `transactions.csv`. It must be identical to `csv_sample.csv`.
    - **The Code/Script:** A script (e.g., Python, JavaScript) that demonstrates your extraction and formatting pipeline. This script should be well-commented and easy to run.
    - **A "ReadMe" File:** A markdown file (`README.md`) that explains your approach. This is the most crucial part of the task. It should describe:
        - **The Tools Used:** List and briefly explain the tools or libraries you used (e.g., "Used Tesseract for OCR, followed by the `pandas` library for data manipulation.").
        - **The Pipeline Steps:** Detail the steps of your pipeline, with a specific focus on how you achieved the exact output format. For example: "1. Perform OCR on the image. 2. Filter the raw text to isolate the transaction

data. 3. Parse and structure the data into a table format. 4. Clean and format the columns (e.g., convert dates to a specific format, ensure numeric values are correct). 5. Export the final data to `transactions.csv` to match the target file."

- **Challenges and Solutions:** What were the difficulties you encountered in matching the output exactly (e.g., date format issues, whitespace discrepancies), and how did you solve them?
- **Improvements:** How would you improve this pipeline to handle slight variations in the input image or to be more robust?

Deliverables:

Please submit a zip file or a folder containing:

1. `transactions.csv`
2. `extraction_script.py` (or similar script)
3. `README.md`