Calling GenAI API...

2025-02-10 09:55:36,428 - httpx - INFO - HTTP Request: POST https://integrate.api.nvidia.com/v1/chat/completions "HTTP/1.1 200 OK"

Received response from GenAI model.

<think>

Okay, let's start analyzing the ETL scripts provided. The user wants me to check for compliance in auditability, reconcilability, restartability, and exception handling. Also, need to check if the code is just comments/readme and if best practices are followed.

First, looking at the CSVETL-script.py. The main ETL process is here. Let's check each category.

Auditability: The script has logging in the error logging script, but in the main ETL script, there's no logging of start/end times or row counts. The error logging script does log "Starting ETL job..." and "completed successfully", but the main script doesn't. Also, the error logging script is separate, and the main ETL doesn't integrate it. So for auditability, the main script is missing timestamps and row counts. The error script has some logging but not integrated into the ETL process. So maybe Auditability is PARTIAL because there's some logging but not in the main ETL steps.

Reconcilability: The script transforms data by uppercasing, but there's no checks for data movement. For example, no checksums, row counts before and after transformation, or validation of data integrity. So Reconcilability is NO.

Restartability: The script doesn't have any mechanism to resume from failure. If it fails midway, it would start over. No checkpoints or state tracking. So Restartability is NO.

Exception Handling: The main ETL script (CSVETL-script.py) doesn't have any try-except blocks. The error logging script (ETLErrorLogging-script.py) has a try-except but it's a dummy example (dividing by zero) and not integrated into the actual ETL process. So the actual ETL code doesn't handle exceptions. Hence, Exception Handling is NO.

Now, the other questions: Does the script only contain comments/readme? No, there's executable code. For best practices: The code uses functions, which is good. But lacks error handling, logging, uses basic CSV reading/writing without handling potential issues like encoding, file not found, etc. So best practices are not fully followed. So answer No for that.

So compiling all this:

Auditability: PARTIAL (logging exists in a separate script but not integrated, no row counts or timestamps in main ETL)

Reconcilability: NO

Restartability: NO

Exception Handling: NO (main script lacks it)

Script only comments? No

Best practices? No (lacks error handling, logging, etc.)

</think>

```json

{

"compliance\_analysis": {

"auditability": {

"result": "PARTIAL",

"evidence": {

"start\_end\_timestamps": "Logging in 'ETLErrorLogging-script.py' includes timestamps via logging.basicConfig, but not integrated into main ETL script.",

"row\_counts": "No row count tracking in CSVETL-script.py (e.g., input/output validation).",

"logs": "Separate error logging script exists but is not used in the main ETL process. Main ETL has no logging."

}

},

"reconcilability": {

"result": "NO",

"evidence": {

"data\_movement\_checks": "No checksums, row counts, or data validation in CSVETL-script.py.",

"transformations": "Transformations (uppercase) are applied but not validated (e.g., no unit tests or post-load verification)."

}

},

"restartability": {

"result": "NO",

"evidence": {

"resume\_mechanism": "No checkpoints, transaction IDs, or state tracking in CSVETL-script.py. Job cannot resume after failure."

}

},

"exception\_handling": {

"result": "NO",

"evidence": {

"error\_handling": "Main ETL script (CSVETL-script.py) lacks try/except blocks. Error logging script is a dummy example (10/0) and not integrated.",

"alerts": "No alerts or error escalation mechanisms."

}

},

"additional\_checks": {

"script\_contains\_only\_comments": "No",

"follows\_best\_practices": {

"result": "No",

"evidence": "No input validation (e.g., file existence checks), no context managers for resources beyond 'with open', and no separation of concerns (e.g., logging/ETL are split into separate files without integration)."

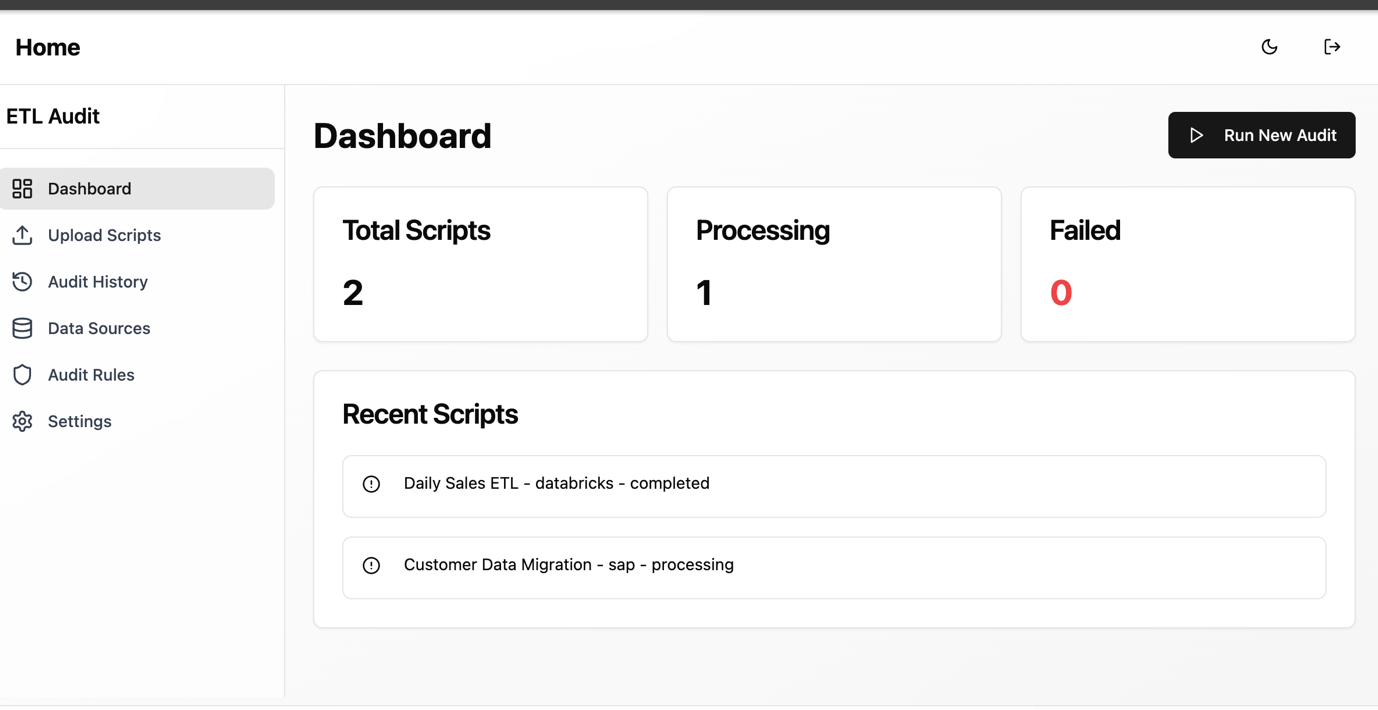
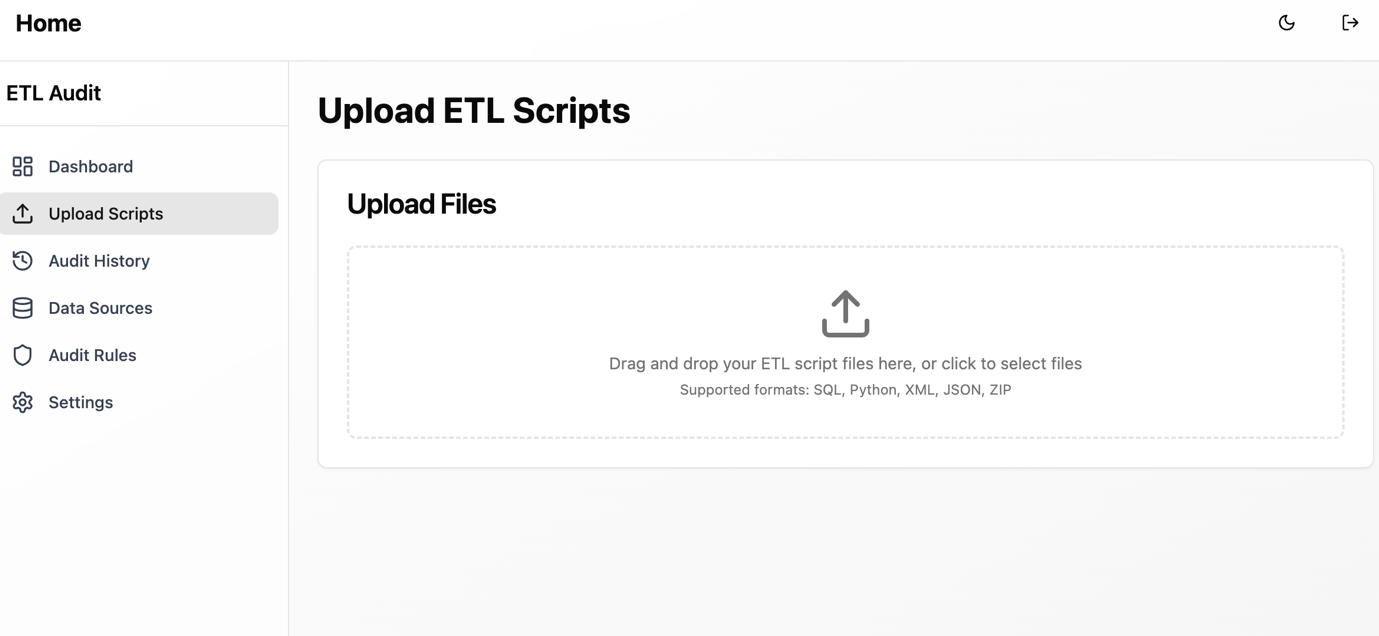
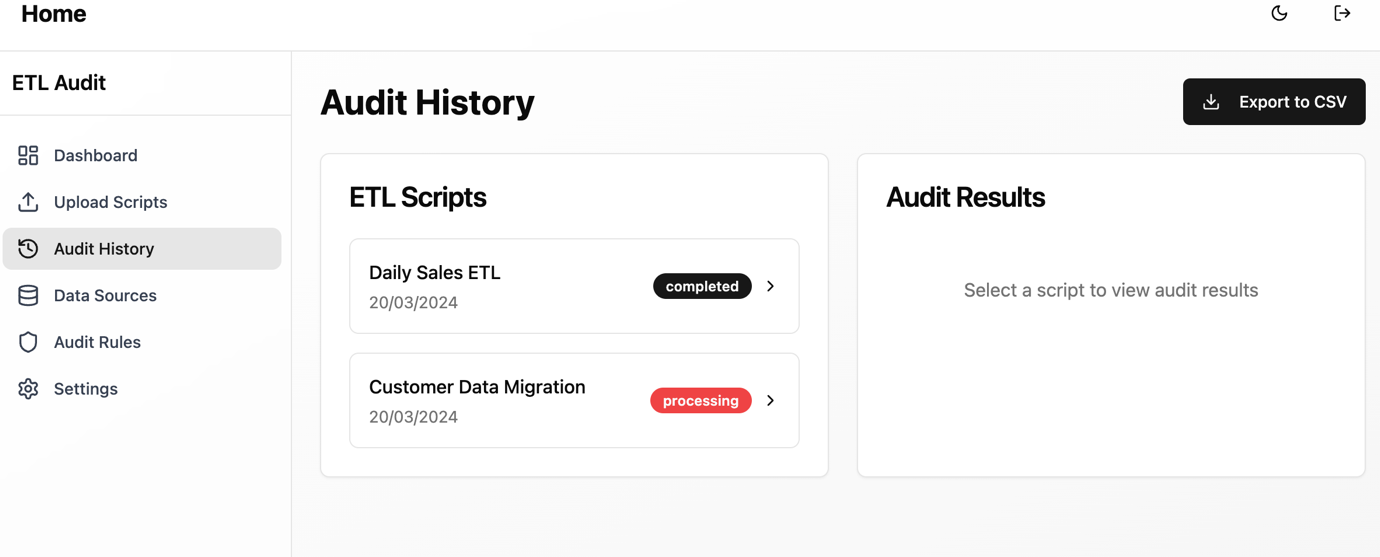
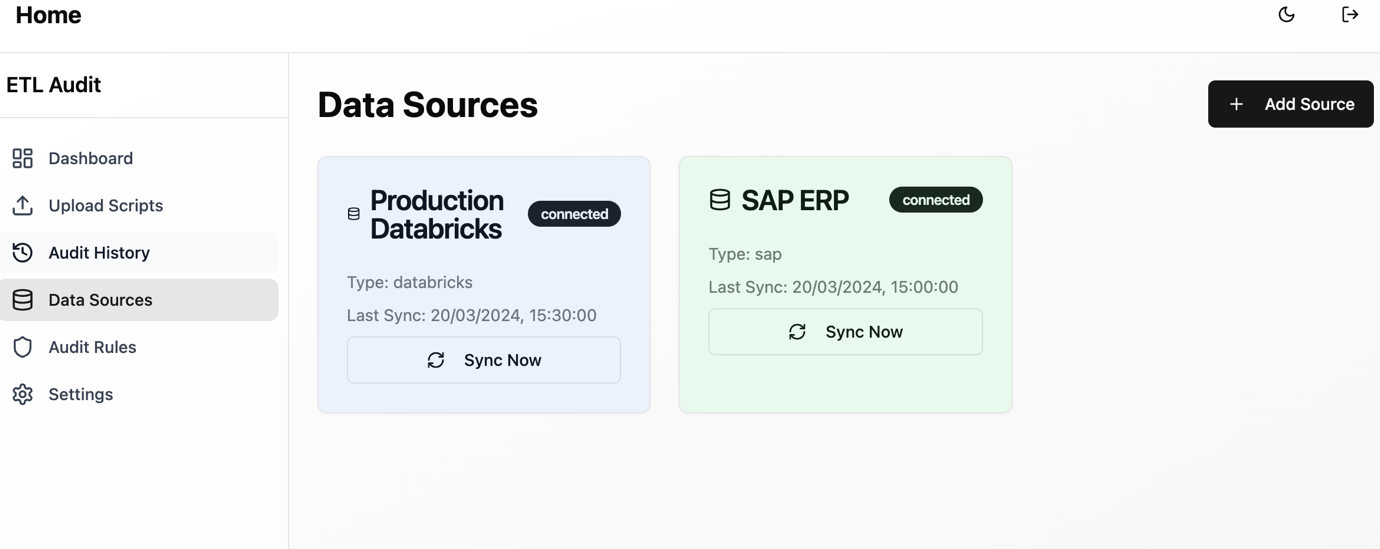
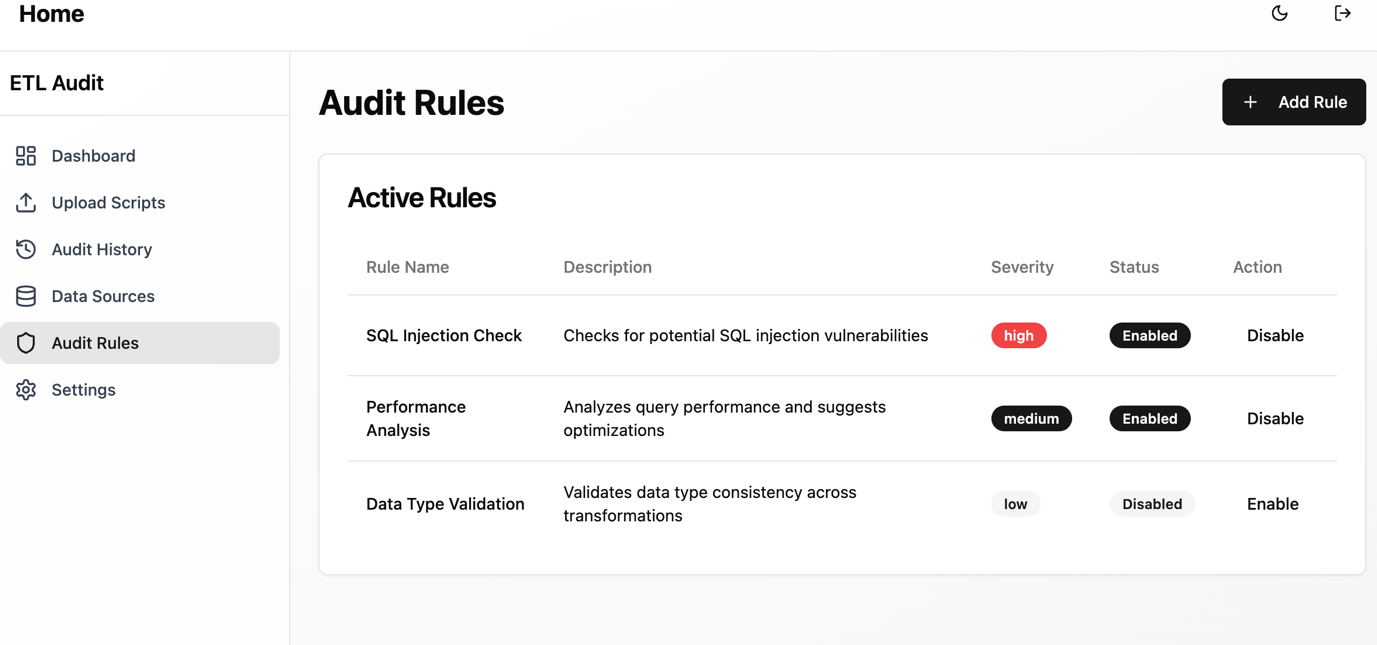
}

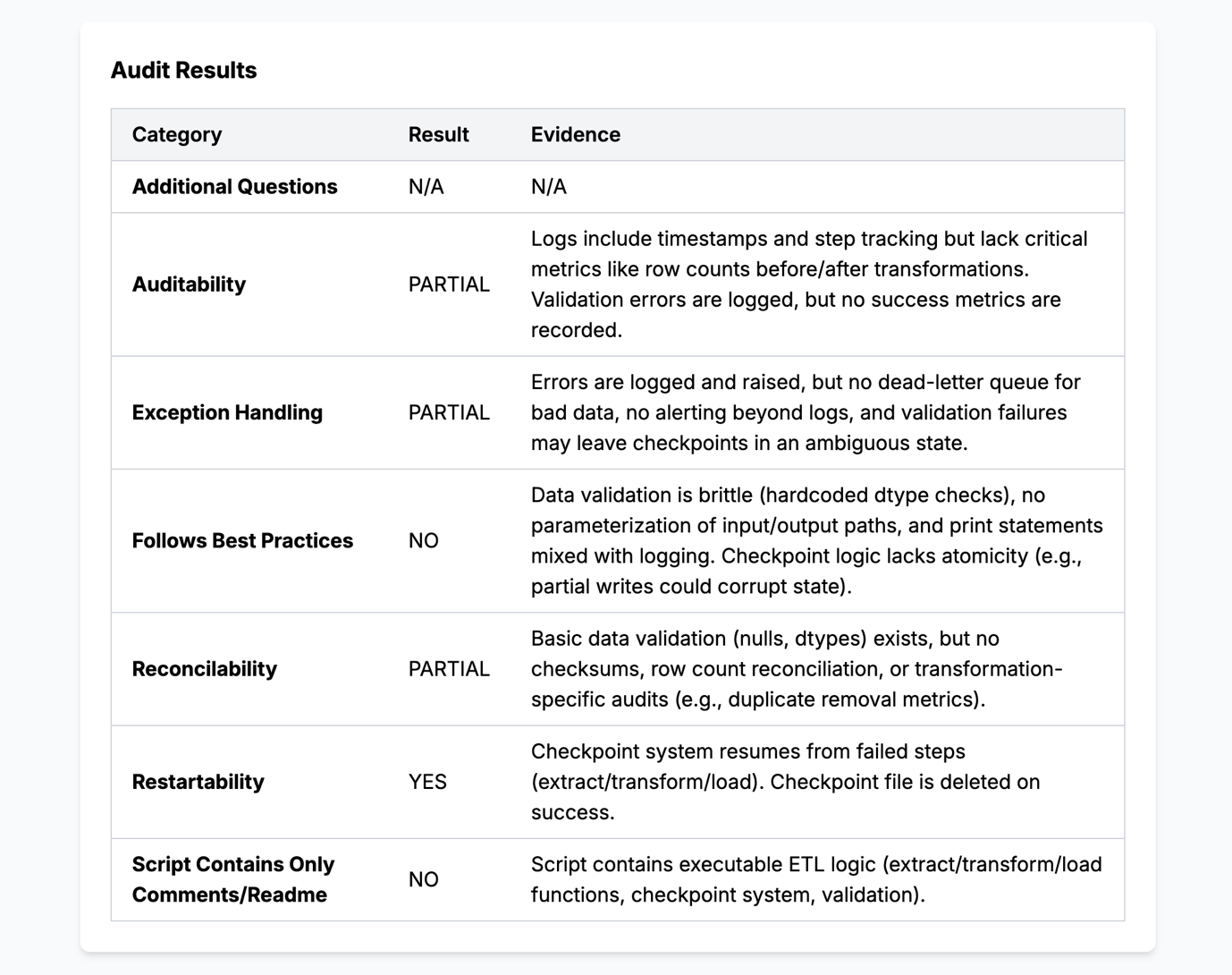
}

}

}

```

Audit report successfully generated.

A screenshot of a computer

AI-generated content may be incorrect.