Evaluation Guidelines: LLM-Powered Term Sheet Reconciliation Assignment

Overview

This document outlines the evaluation framework for candidate submissions on the LLM-powered term sheet reconciliation assignment. Solutions will be assessed across technical accuracy, automation, code quality, and production readiness.

Category	Max Points
1. Parsing & Field Extraction	3
2. LLM & API Integration	3
3. Booking Data Ingestion & Comparison Logic	3
4. Reconciliation Output	3
5. Code Quality, Modularity, and Documentation	4
6. DevOps/Productionization Awareness	2
7. Robustness & Error Handling	2

(Total 20 Points)

1. Parsing & Field Extraction (0–3 points)

- **3:** Robust, automated extraction across both PDF and Word term sheets; handles diverse formats and noise adeptly.
- **2:** Reasonable extraction covering majority of fields and formats but with some misses.
- **1:** Extraction brittle or reliant on overly rigid assumptions; struggles with real-world document variation.

2. LLM & API Integration (0–3 points)

- **3:** Fully automated, programmatic use of an LLM with free/public API; clean prompt engineering; handles API errors gracefully.
- **2:** Mostly automated but with some manual steps or fragile integration.
- **1:** Manual usage or incomplete API-based workflow.

3. Booking Data Ingestion & Comparison Logic (0–3 points)

- **3:** Flexible ingestion supporting multiple formats (CSV/JSON); systematic, extendable reconciliation logic.
- **2:** Supports one format well; partial field comparison or hardcoding present.
- **1:** Limited or error-prone logic.

4. Reconciliation Output (0–3 points)

- **3:** Clear, precise output table/file indicating all matches and mismatches; business-useful format.
- **2:** Generally clear, but room for improved clarity or completeness.
- **1:** Confusing, incomplete, or incorrect output.

5. Code Quality, Modularity, and Documentation (0–4 points)

- **4:** Exceptional code clarity and structure; comprehensive README with environment setup, usage, API key config, and troubleshooting; well-handled secrets; modular and reusable components.
- **3:** Clean, modular, adequately documented code; good README and handling of details.
- **2:** Some clarity or documentation gaps but mostly understandable.
- **1:** Unruly, monolithic, or poorly documented code.
- **0:** Code unusable.

6. DevOps/Productionization Awareness (0–2 points)

- **2:** Thoughtful automation suggestions, CI/CD awareness, error logging, scaling, data security in write-up.
- **1:** Basic mention but lacking depth.
- **0:** No consideration.

7. Robustness & Error Handling (0–2 points)

- **2:** Graceful handling of missing/ambiguous fields, malformed inputs; informative logs and recoverability.
- **1:** Partial or unreliable error management.
- **0:** No error handling.