AI Mapping Copilot Specification Document

1. Spec Packet: AI Mapping Copilot

Problem & Goals

- **The Problem:** Insurance operations teams spend hours manually reformatting client spreadsheets. Inconsistent headers and data formats lead to ingestion errors, delayed coverage, and manual rework.
- **The Goal:** Reduce the time-to-map by 80% while maintaining 100% data integrity through a "propose-verify" workflow.

User Workflow

1. **Upload:** User uploads a CSV/XLSX file.
2. **Analysis:** System parses headers and a sample of data rows (anonymized).
3. **Proposal:** AI generates a side-by-side mapping of "Source Column" to "Canonical Field."
4. **Review:** User views confidence scores and reasoning.
5. **Edit/Approve:** User manually corrects any low-confidence mappings.
6. **Finalize:** Data is transformed and pushed to the downstream system.

Requirements

- **R1: Semantic Matching:** The AI must look beyond header names (e.g., "Field 1") to the actual data values (e.g., "1990-05-15") to identify fields.
- **R2: Confidence Scoring:** Every mapping must have a High/Med/Low score.
- **R3: Reasoning Tooltips:** Provide a brief "Why" for each mapping (e.g., *"Pattern matches standard date-of-birth formats"*).
- **R4: Mandatory Field Validation:** UI must flag if a required canonical field (e.g., employee_id) is missing.
- **R5: Data Privacy:** No PII is sent to the LLM; only headers and sanitized "Type Examples" (e.g., [Name], [Name] instead of John Doe).

AI Behavior + Guardrails

- **Temperature:** Set to 0 for maximum reproducibility.
- **Prompting Strategy:** Use "Few-Shot" prompting with examples of tricky insurance headers.
- **Guardrail (The "Hallucination" Filter):** The system is strictly forbidden from "inventing" columns. If a source column doesn't fit the schema, it must be labeled Unmapped.
- **Privacy Guardrail:** Use a local Regex pre-processor to scrub specific identifiers before sending samples to the LLM.

Evaluation Plan

- **Metric 1 (Accuracy):** % of AI mappings that match the "Gold Standard" without human edits.
- **Metric 2 (Efficiency):** Time taken to complete a 50-column mapping manually vs. with the Copilot.
- **Metric 3 (Safety):** Rate of "Critical Misses" (e.g., mapping Salary to Zip Code).

Instrumentation & Logging

- **Mapping Latency:** Time from upload to proposal.
- **User Edit Rate:** Log which specific fields are most frequently corrected by users to retune the prompt.
- **Schema Drift:** Log source headers that the AI fails to map consistently to identify needs for new canonical fields.

Rollout Plan

- **Phase 1 (Internal Beta):** Use the tool on 10 historical datasets.
- **Phase 2 (Shadow Mode):** Run the AI in the background of live operations; compare AI results to manual results.
- **Phase 3 (GA):** Launch as the primary interface for the ops team.

---

2. Hands-On Artifact: Option C (Prompt + Rubric)

The Prompt

Plaintext
ROLE
You are a Senior Insurance Data Architect. Your task is to map source spreadsheet headers to a target Canonical Schema.

TARGET CANONICAL SCHEMA
- employee_id
- first_name
- last_name
- date_of_birth
- state
- zip
- annual_salary
- hire_date
- employment_status
- coverage_amount
- smoker (yes/no)
- dependent_count

INPUT DATA
Source Headers: {source_headers}
Sample Values: {sample_data_values}

INSTRUCTIONS
1. Analyze the header name AND the sample data.
2. If a column does not fit any canonical field, map it to "unmapped".
3. Provide a confidence score (High, Medium, Low).
4. Provide a brief explanation for your choice.

OUTPUT FORMAT (JSON ONLY)
```
[
  {
    "source_header": "string",
    "target_field": "string or null",
    "confidence": "High|Medium|Low",
    "reasoning": "string"
  }
]
```

Scoring Rubric (Evaluation)

Criteria	**1 - Fail**	**3 - Pass**	**5 - Exceptional**
Accuracy	Maps salary to ID or birthdate.	Correctly maps 90%+ of standard fields.	Correctly identifies ambiguous fields (e.g., "Status" vs "Emp Status").
Reasoning	Vague (e.g., "It matches").	Explains based on data patterns (e.g., "Matches YYYY-MM-DD").	Explains why it *rejected* a secondary possibility.
Safety	Suggests a mapping for PII with Low confidence.	Correctly flags "Unmapped" for non-schema columns.	Flags potential data quality issues in source (e.g., "Mixed date formats").

AI Collaboration Log

Tools Used: Google Gemini (Large Language Model), Perplexity

Step 1: Establishing the Framework

- **Prompt (Exact Text):** *"Help me design a product spec for an AI tool that maps messy insurance spreadsheets to a clean schema. What sections should I include to show good product thinking and attention to detail?"*
- **LLM Output:** Suggested standard PRD sections like "Goals," "User Stories," "Success Metrics," and "Technical Architecture (Frontend/Backend)."
- **Critique & Change:** The initial output was too "generic SaaS." For insurance operations, the architecture is less important than **trust and data privacy**.
- **Adjustment:** I removed the generic tech stack sections and replaced them with **"AI Behavior + Guardrails"** and **"Instrumentation & Logging"** to focus on how we handle hallucinations and measure performance.

Step 2: Refining the AI Intelligence (The Prompt)

- **Prompt (Exact Text):** *"Write a prompt for an LLM to map source spreadsheet headers like 'DOB', 'Salary', and 'Zip' to a canonical schema. The output should be JSON."*
- **LLM Output:** A simple prompt that instructed the AI to look at the strings in the `source_headers` list and find the closest match in the schema.
- **Critique & Change:** In real insurance data, headers are often missing (e.g., "Column 1") or misleading. Mapping based solely on header text is a "Pass" but not "Exceptional."
- **Adjustment:** I modified the prompt to include **`sample_data_values`**. This forces the AI to perform **semantic pattern matching** (e.g., recognizing a 5-digit number as a Zip Code even if the header is "Field_4"). I also added a "Reasoning" field to the JSON to satisfy the requirement for an "Explanation Panel."

Step 3: Building the Evaluation Strategy

- **Prompt (Exact Text):** *"Create a scoring rubric to grade how well an AI performs at this data mapping task. Focus on accuracy and safety."*
- **LLM Output:** A basic 1-5 scale focusing on "Correct Mappings" vs "Incorrect Mappings."
- **Critique & Change:** The output didn't distinguish between a "Low Confidence" mistake (which a human can fix) and a "High Confidence Hallucination" (which breaks the system).
- **Adjustment:** I redesigned the rubric to include a **"Safety"** column. This rewards the AI for being "Humble"—meaning it gets a higher score for marking an ambiguous column as "Unmapped" than it does for guessing incorrectly. This aligns with the "Insurance Operations" constraint where data integrity is the top priority.