### **Mercor Mini-Interview Task: Airtable Multi-Table Form + JSON Automation**

**Goal:** Design an Airtable-based data model and automation system that

1. Collects contractor-application data through a structured, multi-table form flow
2. Local Python script that compresses the collected data into a single JSON object for storage and routing
3. Local Python script that decompresses the JSON back into the original, normalized tables when edits are needed
4. Auto-shortlists promising candidates based on defined, multi-factor rules
5. Uses an LLM endpoint to evaluate, enrich, and sanity-check each application

## **1. Airtable Schema Setup**

Create a base with **three linked tables** plus two helper tables:

| **Table** | **Key Fields** | **Notes** |
| --- | --- | --- |
| **Applicants** (parent) | Applicant ID (primary), Compressed JSON, Shortlist Status, LLM Summary, LLM Score, LLM Follow-Ups | Stores one row per applicant and holds the compressed JSON + LLM outputs |
| **Personal Details** | Full Name, Email, Location, LinkedIn, *(linked to Applicant ID)* | One-to-one with the parent |
| **Work Experience** | Company, Title, Start, End, Technologies, *(linked to Applicant ID)* | One-to-many |
| **Salary Preferences** | Preferred Rate, Minimum Rate, Currency, Availability (hrs/wk), *(linked to Applicant ID)* | One-to-one |
| **Shortlisted Leads** | Applicant (link to Applicants), Compressed JSON, Score Reason, Created At | Auto-populated when rules are met |

All child tables are linked back to **Applicants** by Applicant ID.

## **2. User Input Flow**

Airtable’s native forms can’t write to multiple tables simultaneously, so simulate the flow with **three forms** (one per child table) that each pre-fill or ask for the Applicant ID. Require applicants to submit all three forms.

*Steps 3-4 can be done in a local Python file outside of Airtable. When you run the scripts you can just reflect the updates in Airtable using the API.*

## **3. JSON Compression Automation**

1. **Action:** Write a Python local script that gathers data from the three linked tables, builds a single JSON object, and writes it to Compressed JSON.

{

"personal": { "name": "Jane Doe", "location": "NYC" },

"experience": [

{ "company": "Google", "title": "SWE" },

{ "company": "Meta", "title": "Engineer" }

],

"salary": { "rate": 100, "currency": "USD", "availability": 25 }

}



## **4. JSON Decompression Automation**

Write a separate Python local script that can:

1. Read Compressed JSON.
2. Upsert child-table records so they exactly reflect the JSON state.
3. Update look-ups/links as needed.

## **5. Lead Shortlist Automation**

After compression, evaluate rules:

| **Criterion** | **Rule** |
| --- | --- |
| **Experience** | ≥ 4 years total **OR** worked at a Tier-1 company (Google, Meta, OpenAI, etc.) |
| **Compensation** | Preferred Rate ≤ $100 USD/hour **AND** Availability ≥ 20 hrs/week |
| **Location** | In US, Canada, UK, Germany, or India |

If all criteria are met, create a **Shortlisted Leads** record and copy Compressed JSON. Populate Score Reason with a human-readable explanation.

## **6. LLM Evaluation & Enrichment**

### **6.1 Purpose**

Exercise a modern LLM (e.g., OpenAI, Anthropic, Gemini) to automate qualitative review and sanity checks.

### **6.2 Technical Requirements**

| **Aspect** | **Requirement** |
| --- | --- |
| **Trigger** | After Compressed JSON is written **OR** updated |
| **Auth** | Read API key from an Airtable **Secret** or env variable (do **not** hard-code) |
| **Prompt** | Feed the full JSON and ask the LLM to: • Summarize the applicant in ≤ 75 words • Assign a quality score from 1-10 • Flag any missing / contradictory fields • Suggest up to three follow-up questions |
| **Outputs** | Write to LLM Summary, LLM Score, LLM Follow-Ups fields on **Applicants** |
| **Validation** | If the API call fails, log the error and retry up to 3× with exponential backoff |
| **Budget Guardrails** | Cap tokens per call and skip repeat calls unless input JSON has changed |

### **6.3 Sample Prompt (pseudo-code)**

****You are a recruiting analyst. Given this JSON applicant profile, do four things:

1. Provide a concise 75-word summary.

2. Rate overall candidate quality from 1-10 (higher is better).

3. List any data gaps or inconsistencies you notice.

4. Suggest up to three follow-up questions to clarify gaps.

Return exactly:

Summary: <text>

Score: <integer>

Issues: <comma-separated list or 'None'>

Follow-Ups: <bullet list>

### **6.4 Expected Results**

| **Field** | **Example Value** |
| --- | --- |
| LLM Summary | *“Full-stack SWE with 5 yrs experience at Google and Meta…”* |
| LLM Score | 8 |
| LLM Follow-Ups | • “Can you confirm availability after next month?”• “Have you led any production ML launches?” |

## **Deliverables**

1. **Airtable base** (share link) with all tables, automations, and scripts.
2. **Documentation** (Markdown or Google Doc) explaining:  
   * Setup steps and field definitions
   * How each automation works, including script snippets
   * How the LLM integration is configured and secured
   * How to extend or customize the shortlist criteria

**No emojis** should appear in any field names, table names, or documentation.