### **Project Brief: Automated Job Data Ingestion & Enrichment Pipeline**

**Objective:** To operationalize an existing data processing pipeline, transforming it from a Google Colab notebook prototype into a robust, documented, and deployable service on Google Cloud Run.

**Deadline:** Friday, End of Day (PST)

#### **1. Project Context & Business Goal**

We have developed a proof-of-concept (PoC) in a Google Colab notebook that is a POC and not connected, but could successfully automate the ETL (Extract, Transform, Load) process for job postings. This pipeline ingests raw data from various sources and formats, standardizes it, enriches it using AI, and loads it into a database.

The goal of this project is to harden this prototype into a reliable, automated service that can run in the cloud without manual intervention. This will form a core part of our data aggregation strategy.

#### **2. Core Deliverables**

1. **Functional & Refined Codebase:**
   * Refactor the existing notebook script into a clean, modular, and production-ready Python application.
   * Ensure all environment variables and secrets (API keys for Supabase, Gemini, Xano) are handled securely, not hard-coded.
   * Create a requirements.txt file listing all necessary dependencies.
2. **Deployment to Google Cloud Run:**
   * Containerize the Python application using a Dockerfile.
   * Successfully deploy the container as a serverless function on Google Cloud Run.
   * The function should be triggerable (e.g., via an HTTP request) and capable of processing a data file passed to it.
3. **Clear Documentation:**
   * A README.md file detailing:
     + A concise overview of the pipeline's architecture and data flow.
     + Step-by-step instructions for setting up the local development environment.
     + Instructions on how to build the Docker container and deploy it to Cloud Run.
     + An explanation of required environment variables/API keys.

#### **3. Provided Assets**

* The source Google Colab Notebook.
* A set of sample data files (XML, CSV, JSON, and a compressed archive) for testing.
* The target data schema definition for validation.
* All necessary API credentials for the mocked services (Supabase, Gemini, Xano).

#### **4. Acceptance Criteria (How We'll Define "Done")**

* The final Python application runs without errors from start to finish on a local machine.
* The Docker container builds successfully.
* The application is deployed on Cloud Run and is in a healthy, running state.
* The deployed Cloud Run function correctly processes all provided sample files when triggered, with data appearing in the target (mocked) databases as expected.
* The codebase is well-commented, and the README.md is clear and comprehensive.

#### **5. Required Contractor Skills**

To complete this project by the deadline, the contractor must be proficient in the following areas:

**Essential (Must-Have Skills):**

* **Expert Python:** Deep knowledge of Python 3, including file I/O, data structures, error handling, and best practices for writing production code.
* **Google Cloud Platform (GCP) & Cloud Run:** Proven, hands-on experience deploying containerized applications to Google Cloud Run. This is the primary technical requirement.
* **Containerization (Docker):** Ability to write an efficient and clean Dockerfile to containerize the Python application and its dependencies.
* **API Integration:** Experience working with REST APIs, including sending data and handling responses and errors from external services.

**Specialized Expertise (High-Value Skills):**

* **ETL Pipeline Development:** A strong background in building, deploying, and maintaining ETL workflows.
* **Data Parsing & Wrangling:** Expertise in handling and transforming messy data from varied formats (XML, JSON, CSV). Familiarity with libraries like pandas, lxml, or dicttoxml is a plus.
* **AI/LLM Integration:** Experience making API calls to Large Language Models (like Gemini or OpenAI) and implementing logic based on their output.

**Professional Skills:**

* **Problem-Solving:** Ability to independently diagnose and fix issues in the existing code and during deployment.
* **Efficiency & Time Management:** Capable of delivering a high-quality, fully functional result within the tight deadline.
* **Clear Communication:** Ability to provide clear status updates and write excellent documentation.