AI Automation Engineer – Technical Assessment

SouthDesk

Overview

This technical assessment evaluates your ability to design and implement a real-world, AI-assisted automation system using Python, LLMs, and no-code/low-code tools under realistic constraints.

This is an **open-ended challenge**. There is no single correct solution. We are primarily interested in how you think, design, and communicate.

We care about:

- System architecture and decision-making
- Practical automation skills
- Data quality, validation, and reliability
- Thoughtful and explainable use of AI
- Clear technical communication

Recommended Time Commitment

- **Core requirements:** ~4–6 hours
- **Including optional stretch modules:** ~8–12 hours

You are **not expected to complete everything**. A solid, well-executed core solution is sufficient.

Constraints

- Free tiers only (e.g. Google Gemini, Make.com, n8n)
- Local development is allowed
- No paid SaaS tools

- No business Google accounts required
- No proprietary CRMs

If you cannot use a free Google Gemini API key, we can provide an OpenAI API key upon request.

Business Scenario

You are building a lightweight AI-assisted lead qualification system for ABC Company.

ABC Company builds AI-powered customer support solutions designed for small and mid-sized e-commerce businesses. The platform helps online retailers automate first-line support, reduce response times, and improve customer satisfaction by combining conversational AI with human-in-the-loop workflows.

The system should:

- Collect inbound leads
- Normalize and validate incoming data
- Enrich company information from public sources
- Use AI to qualify and score leads
- Store everything in a simple Google-based system
- Trigger basic downstream actions

Core Requirements (Mandatory)

1. Lead Intake (Google Forms)

Create a Google Form with mixed data types, such as:

- Full name
- Email address
- Company name
- Company website (optional)
- Country

- Lead source (Website, Referral, LinkedIn, Event, Other)
- Estimated budget (ranges)
- Notes (optional)
- Consent to contact (Yes / No)

Form submission must trigger your automation workflow.

2. Automation Orchestration

Choose **one** primary orchestration tool:

- Make.com **or**
- n8n

It must:

- Receive the form submission
- Normalize and validate data
- Handle conditional logic (e.g. based on lead source or consent)
- Create or update records in Google Sheets
- Call the Python enrichment service via HTTP

Using both Make and n8n together is optional and considered a bonus.

3. Data Normalization & Validation

Before storing any data:

- Normalize formats (trim strings, lowercase emails, parse budgets)
- Validate required fields
- Handle missing or malformed inputs gracefully
- Deduplicate leads (email or email + company)

Use a structured schema (JSON or equivalent).

4. Google Sheets as Lead Database

Use Google Sheets as the main data store. Include columns for:

- Lead identifiers
- Raw lead data
- Enrichment status
- AI outputs (summary, score, recommendation)
- Error and log information

Rules:

- No duplicate leads
- Existing leads must be updated, not duplicated
- Errors must be visible and traceable

5. Python Enrichment Service

Build a Python service that:

- Runs locally
- Is exposed via ngrok
- Is called via HTTP from your automation tool

The service must:

1. Discover the company website if missing (simple search is sufficient)
2. Fetch basic public company information
3. Call the provided LLM API to:
    a. Summarize the company
    b. Assess fit for ABC Company
    c. Produce a lead score (0–100)
    d. Recommend a next action

LLM responses must be:

- Structured JSON
- Validated against a schema

Handle:

- Invalid AI responses
- Timeouts and failures gracefully

6. Updating the Sheet

After enrichment:

- Update the correct row in Google Sheets
- Mark enrichment status (pending, done, error)
- Store AI outputs in dedicated columns

Optional Stretch Modules

- **A. Google Apps Script Menu**

Custom menu actions such as enriching selected leads or archiving data.

- **B. Make + n8n Together**

Use both tools in a coordinated workflow.

- **C. Marketing Actions**

Trigger actions like Gmail drafts, Calendar events, or reminders.

- **D. Bulk CSV Import**

Import, validate, deduplicate, and enrich CSV leads.

- **E. Reliability Improvements**

Idempotency, retries, backoff, structured logs, state machines.

Design Artifact (Mandatory)

Provide a workflow diagram (Draw.io, Miro, Lucidchart, etc.) showing:

- Tools involved
- Data flow
- Triggers
- Decision points
- Error paths

Deliverables

1. Workflow diagram
2. Code repository :
    a. Python service
    b. Apps Script code (if any)
    c. README with setup instructions
3. Short write-up:
    a. Architecture overview
    b. Tool choices and tradeoffs
    c. Design decisions and justification
    d. AI prompt and schema strategy
    e. Error handling approach
    f. Improvements and next steps

Evaluation Criteria

We evaluate:

- System-level thinking
- Practical automation decisions
- Robust handling of real-world data
- Clean, explainable Python
- Thoughtful and controlled use of AI
- Clear written and visual communication