

Dragify Engineering Assessment

Role: Software Engineer

Company: Cloudilic

Product: Dragify – AI Agent Automation Platform



Assessment Objective

Build a **modular AI Agent Template** that automates a realistic business workflow involving:

1. Trigger

Agent listens to one of the following real-time events (pick one from the following list):

- A new message in **Slack**
- A new **email** (via Gmail or Outlook)
- A **custom webhook** from a web form, CRM, or external tool

2. Lead Data Collection

Fetch lead data from realistic business sources (pick one from the following list):

- **Google Sheets** (connected to a form)
- **NoSQL** document (e.g., Firebase, MongoDB)
- **Relational database** (e.g., PostgreSQL, MySQL)
- **Webform** submissions via webhook
- **Airtable**

3. CRM Integration

Insert the lead data into a CRM system (pick one from the following list):

- **Zoho CRM**
- **Odoo CRM**
- **Salesforce**
- (Or use mock API endpoints)

4. Status Notification

Send an email confirmation about the lead creation status (success or failure) using Gmail or Outlook integrations.



Core Requirements

- Build an **AI Agent Template**, not a fixed script.
- The flow (trigger, data source, CRM) must be **dynamically configurable**.
- **Frameworks Allowed:**
 - **LangChain**, **LangGraph**, or custom orchestration
 - **Zapier**, **N8N**, **Make.com**, **AgentBuilder**, etc.

Engineering Standards

- Clean code with **SOLID**, **OOP**, and **modular architecture**
- Documented via **README.md** and code comments
- Token-based **OAuth 2.0** with secure capture.
- Scalable architecture for multi-user support and extensibility

Auth Handling

- Implement full OAuth 2.0 flows for:
 - **Slack**
 - **Google (Gmail, Sheets, etc.)**
 - **Outlook**
- Store tokens securely in **PostgreSQL** or **MongoDB**



Frontend Interface (Required)

Create a minimal UI using **React + TailwindCSS** that displays:

- **Trigger Logs** (event type, timestamp)
- **Fetches Lead Data**
- **CRM Status** (success or failure)

Must be **clean, simple, responsive**, and match Dragify's standards for usable internal tools.

Deployment (Required)

Deploy the frontend to one of the following and share a **public link** in your README:

- [Vercel](#) – Recommended
- Netlify
- Render
- Any modern frontend hosting service

Tech Stack (Must Be Used)

Category	Technology
Backend Lang	Python or TypeScript (Node.js)
Backend Framework	FastAPI or Express.js
Agent Logic	LangChain, LangGraph
Database	PostgreSQL (preferred) or MongoDB
Frontend	React + TailwindCSS
OAuth	Full token exchange + callback handling



Bonus Points

- Retry logic for CRM failures
 - Multi-user credential handling
 - An interface to modify the user story in real time
 - Basic unit or mock tests
 - Short Loom video walkthrough
-



Submission Checklist

- Codebase on GitHub
- Public deployment of frontend (e.g., Vercel)
- Clear README with:
 - Setup instructions
 - OAuth instructions
 - API endpoints
 - Frontend live link
- (Optional) Demo video (Loom, 5 min max)