# Developer Assignment: SEO Directory Submission AI Agent

## Objective:

Build a smart, scalable, and autonomous agent that automates manual SEO directory submissions for startups and businesses. This tool should collect business data once, use AI agents to autofill directory submission forms, handle login, captchas, and dynamic fields, and track listing statuses weekly, stopping once successful.

## Core Features & Functionalities:

### 1. Startup Data Intake Module

- Collect and store company name, tagline, logo, website URL, email, phone, business description, social media links, founder name, business category & keywords, address, and location.  
- Save data to structured JSON or database.

### 2. CSV URL Input & Queue Builder

- Accepts a CSV with directory submission URLs.  
- Parses each line and builds a submission queue.

### 3. Web Agent Capabilities (Core Automation Agent)

- Visit each URL, detect login/registration, solve captcha (integrate with 2Captcha or similar).  
- Locate 'Submit Your Site' buttons, autofill form fields using startup data.  
- Submit the form and record result: success/failure, timestamp, error logs.

### 4. Submission Response Tracker

- Save submission data including URL, HTTP response, screenshot (optional), status.

### 5. Weekly Listing Checker Bot

- Search for company name weekly on submitted directories.  
- If found, mark as 'Live'; if not, requeue.  
- Stop rechecking if listing is found.

## Bonus Features (Optional but Valuable):

- Rate limiting & retries  
- Proxy support/user-agent rotation  
- Email alerts  
- Multi-language directory handling  
- Submission template matcher  
- Dashboard to track status  
- Plugin architecture for adding directories

## Technical Requirements:

- Python preferred (Playwright/Selenium, SQLite/MongoDB, FastAPI/Streamlit)  
- Scheduler (Celery/APScheduler)  
- Captcha solver integration (2Captcha, anti-captcha)

## Deliverables (For Interview Assignment):

1. MVP that accepts data and attempts form submission on 3 real directories.  
2. Video walkthrough.  
3. Code repository with README.  
4. Proposal for scaling to 100s of directories.

## Evaluation Criteria:

- Code clarity & reusability  
- Smart form mapping  
- Real-world success  
- Error handling  
- UX usability  
- Scalability approach