

## Objective

Build an intelligent **Startup Profiler Agent** that gathers and stores contextual information about any company/startup. The agent should use **Gemini** to analyze and infer insights, **SerpAPI** to fetch recent news, **web scraping** to extract website content, and store the structured data in **Supabase**.

This task tests your real-world skills across agents, tool integration, LLM prompt engineering, and data persistence.

## Tech Stack

- **Language:** Python 3.10+
- **LLM:** Google Gemini (via REST API)
- **Database:** Supabase (PostgreSQL)
- **Web Search:** SerpAPI (Free account works)
- **Web Scraping:** Any library (e.g., `requests` + `BeautifulSoup`, `Playwright`, or `Selenium`)
- **Containerization:** Docker
- **Framework:** Langchain

## Agent Task Flow

### Input:

User provides:

- a `website_url` (e.g., `https://notion.so`)

### Agent Responsibilities:

On receiving a request, check if the entry already exists for this company in Supabase. If yes, skip re-processing.

If no, follow the following steps:

## 1. Website Scraping:

- Use web scraping to extract:
  - Page title
  - Meta description
  - All H1/H2 tags
  - Outbound links from the homepage
  - Content

## 2. Fetch Latest News (SerpAPI):

- Use SerpAPI to fetch top 5 **latest news articles** or web results related to the company.
- Extract:
  - Title
  - Snippet
  - Source (URL)
  - Date (if available)

## 3. Gemini-Based Enrichment:

Use Google Gemini to analyze:

- What does the company do? (100-word summary)
- Industry category
- Target audience
- Key problem(s) it solves
- Suggest 2–3 potential competitors
- Summarize the latest news (from SerpAPI results) into a short paragraph

## 4. Add entry to Supabase

## Project Requirements

- Add logging and exception handling  
Project must be **Dockerized**:
  - Provide **Dockerfile** and **docker-compose.yml**
  - Project should run with **docker-compose up**

## Deliverables

- GitHub/GitLab repo with:
  - Source code
  - `README.md` with setup and usage instructions
  - Sample `.env.template` for config vars (e.g., Gemini API key, Supabase URL, SerpAPI key)
  - Screenshots or example JSON output

## Deliverables

- GitHub/GitLab repo with:
  - Source code
  - `README.md` with setup and usage instructions
  - Sample `.env.template` for config vars (e.g., Gemini API key, Supabase URL, SerpAPI key) but remove actual values
  - Screenshots or example JSON output
- Optional (Bonus): Streamlit or Gradio UI

## Submission Instructions

- Submit a GitHub link with clear commit history (remove the actual API keys)
- Include a Loom video (2–3 mins) demoing your work
- Deadline: **5th June 2025 (11:59 PM SGT)**