**🧠 Job Posting Creator – Multi-Agent AI Project with CrewAI**

**📌 Project Overview**

**Job Posting Creator** is an intelligent multi-agent system built using the CrewAI framework. It automates the process of generating professional job description posts based on user input (e.g., job title, industry, location). By coordinating a team of role-based agents, the system performs research, writing, and review tasks to produce high-quality hiring posts.

This project is part of a hands-on self-learning journey focused on mastering **agentic AI**, **multi-agent collaboration**, and **LLM-powered automation** using frameworks like CrewAI and tools such as Ollama LLaMA 3.1 and Serper API.

**💡 Objectives**

* Automatically generate job descriptions tailored to a specified job title, industry, and location.
* Simulate human-like task delegation across multiple AI agents.
* Practice structured AI development using CrewAI YAML project scaffolding.
* Test file-based CrewAI agent deployment in a local environment.

**🧱 Architecture**

**Agents**

Defined in agents.yaml, each agent has a **role**, **goal**, and a rich **backstory**:

* **Job Researcher**
  + **Goal:** Search for insights and requirements related to the given job title.
  + **Backstory:** A skilled market analyst who gathers deep job-related data using AI tools.
* **Job Writer**
  + **Goal:** Draft a detailed, structured job description using research findings.
  + **Backstory:** A creative, professional content writer with HR experience.
* **HR Reviewer**
  + **Goal:** Polish, validate, and refine the final post.
  + **Backstory:** An HR specialist ensuring clarity, compliance, and appeal.

**Tasks**

Defined in tasks.yaml, each task is bound to an agent and outlines what that agent must do.

* **Research Task**
  + Uses tools to analyze job market trends and gather responsibilities/qualifications.
* **Writing Task**
  + Crafts the full job description with markdown formatting.
* **Review Task**
  + Edits and improves the job post to align with HR standards and tone.

**🧰 Tools Used**

* **Ollama** with Meta LLaMA 3.1 – open-source LLM for all agent tasks.
* **Serper API** – search tool used by the researcher to find web-based information.
* **CrewAI File Scaffolding** – YAML-based configuration for agents and tasks.
* **Python (PyCharm)** – for coding, running, and managing the system.

**⚙️ Files and Structure**

Job\_post\_creator\_project/

├── .env # Environment file with API keys and model name

├── pyproject.toml # Project config for CrewAI

├── crew.py # Crew definition: loads agents & tasks

├── main.py # Entry point, runs the crew with user inputs

├── agents.yaml # YAML definition of agents

├── tasks.yaml # YAML definition of tasks

└── output/

└── job\_posting.md # Final job post generated

**✅ Sample Output**

A complete job post is generated and saved in output/job\_posting.md. Example:  
**Senior Python Developer** job post with:

* Title & Summary
* Responsibilities
* Qualifications
* Benefits
* How to Apply

👉 This output mimics a high-quality HR post ready for platforms like LinkedIn or Glassdoor.

**🌍 How It Works (Execution Flow)**

1. User provides job title, industry, and location via main.py.
2. Agents are loaded from YAML and assigned to tasks via crew.py.
3. Each agent completes its task using the LLM and tools.
4. Final result is printed and saved to Markdown.

**🔐 Environment Setup**

* .env should include:

MODEL=ollama/llama3.1

SERPER\_API\_KEY=your-serper-api-key

* Install dependencies:

pip install crewai crewai\_tools

* Run the system:

python main.py

**📚 Learning Goals Achieved**

* ✅ CrewAI project structuring with YAML
* ✅ Multi-agent system planning and execution
* ✅ Custom prompt engineering and role definition
* ✅ LLM-based research, generation, and review
* ✅ Tool integration and API handling
* ✅ Building AI systems end-to-end using Python