

Generative AI Agents – Task Automation with LLM Reasoning

LAB Assignment on CrewAI

Assignment 1 – Multi-Agent Research Crew for Research Report Generation

Implement a **sequential CrewAI workflow** where agents work together to research a topic, analyze findings, and produce a final structured report saved to disk.

Requirements:

1. **At least three agents:**
 - **Research Agent:** Searches the web for given topics (tool use – search API).
 - **Analysis Agent:** Reads the research findings, extracts key insights.
 - **Report Writer Agent:** Generates a structured report (JSON & Markdown formats).
2. **Passing context:**

The analysis agent should receive the raw research data from the research agent, and the report writer should receive the analysis output.
3. **Sequential process:**
 - Step 1: Research → Step 2: Analysis → Step 3: Report Writing.
4. **Tool use:**
 - Create a custom "Web Search Tool" that returns top N results.
5. **Structured output:**

The report should have fields: {topic, key_findings, conclusion, references}.
6. **File writing:**

Save the report in both .json and .md formats.
7. **Task callback:**

Use a task callback to print progress after each agent completes its task.

=====

Assignment 2 – Multi-Agent Curriculum Designer - AI-Driven Education Module Builder

Build a hierarchical CrewAI system where a Lead Curriculum Designer Agent delegates tasks to domain-specific agents to create an educational module for a given topic and target audience.

Requirements:

1. **Agents:**
 - **Lead Curriculum Designer Agent** – coordinates overall process, assigns subtasks.
 - **Content Creator Agent** – generates learning material in structured sections.
 - **Assessment Designer Agent** – creates quizzes and assignments for the module.
 - **Resource Curator Agent** – finds reference links and suggested readings.
2. **Process:**
 - **Hierarchical flow:** Lead Agent → three Specialist Agents.
 - Each specialist's output is collected, combined into the final module.
3. **Memory Use:**
 - Persistent memory storing previously created modules.
 - If the same topic is requested again, recall and reuse past content with option to "update."
4. **Structured Output:**
 - {topic, target_audience, module_outline, learning_material, assessments, resources}
5. **File Output:**
 - Save module as .json and .docx formats.
6. **Custom Tool:**
 - ResourceSearchTool – returns curated list of relevant books, articles, and videos.
7. **Callbacks:**
 - **Step callback** logs when each specialist completes their work.
 - **Task callback** confirms when the full module is ready.