The AI Research Agent

The Problem: We are drowning in information but starving for wisdom. Professionals, from market analysts to academic researchers, spend up to 40% of their work week manually searching for information, vetting sources, and synthesizing findings. The process is slow, inefficient, and susceptible to bias. A simple question can lead to hours of sifting through search results, ads, and conflicting reports.

The Challenge: Your task is to build Project Galileo, an autonomous AI research agent. Given a single complex question, your agent must intelligently navigate the web, critically evaluate information, and produce a trustworthy, evidence-backed report. This isn't just about finding answers; it's about building a system that can reason, verify, and synthesize knowledge on its own.

Use Case & Core Requirements:

Your AI agent must execute a full research workflow from query to report.

1. Query Decomposition (The "Planner"):
o Given a high-level query (e.g., “What are the top risks to India’s electronics supply chain in the next 12 months?”), the agent must first break it down into a logical set of smaller, searchable sub-questions.
o Example Sub-Questions: "major electronics component suppliers to India," "geopolitical stability in Taiwan," "Indian government policies on semiconductor manufacturing," "global logistics and shipping risks 2025."
2. Intelligent Search & Data Collection (The "Scout"):
o For each sub-question, the agent must programmatically execute web searches using a search engine API (e.g., Google, Bing, Serper).
o It should then crawl the top-ranking source URLs and scrape the relevant text content from each page.
3. Information Extraction & Synthesis (The "Analyst"):
o The agent must analyze the scraped content from multiple sources to find answers to its sub-questions.
o Crucially, it must identify and store the source URL for every piece of information it extracts.
o It should be designed to synthesize information from various pages to form a more complete picture, discarding irrelevant or low-quality content.
4. Evidence-Backed Report Generation (The "Writer"):
o The final step is to use a Large Language Model (LLM) to generate a concise, well-structured report in Markdown format.
o This is the most critical requirement: Every key statement, fact, or figure in the report must be immediately followed by an inline citation linking back to the source URL where the information

was found. Example: The Indian government recently launched a $10 billion PLI scheme for semiconductor manufacturing [https://example-source.com/article].

Bonus Features & Stretch Goals (For Top Teams):

- Cross-Verification & Contradiction Detection: The agent actively compares facts from different sources. If it finds conflicting information (e.g., Source A reports 15% growth, Source B reports 8%), it should highlight this contradiction in its final report.
- Interactive Chat Interface: Instead of a single report, create a conversational interface. After the initial report is generated, the user can ask follow-up questions like, "Can you tell me more about the geopolitical risks from Taiwan?" and the agent uses its collected knowledge to answer.
- Structured JSON Output: In addition to the Markdown report, generate a structured JSON output that categorizes the findings. This makes the data machine-readable and ready for other applications.
  - Example: { "risk": "Geopolitical Tension", "impact": "High", "timeline": "6-12 months", "evidence": ["url1", "url2"] }
- Autonomous Depth Control: Allow the user to specify a "research depth" (e.g., shallow, normal, deep). A "deep" search would generate more sub-questions and analyze more sources, resulting in a more comprehensive report.

Key Skills to Look For:

- Agentic AI & LLM Orchestration: Ability to design and implement a multi-step AI workflow. Experience with frameworks like LangChain or LlamaIndex is a strong indicator of this skill.
- API Integration: Proficiency in using various APIs (LLMs like GPT/Gemini, Search APIs, etc.).
- Web Scraping & Data Extraction: Skills in pulling and cleaning unstructured data from websites using tools like BeautifulSoup.
- System Design & Critical Thinking: How did the team structure the agent's logic? How did they ensure the final output was reliable and well-supported by evidence?

Suggested Tech Stack & Resources:

- Language: Python is the de facto standard for this type of project.
- AI/LLM Frameworks: LangChain, LlamaIndex, or building a custom agent loop.
- LLM APIs: OpenAI (GPT-4/GPT-3.5), Google AI Platform (Gemini), Anthropic (Claude).
- Search APIs: Google Custom Search API, Bing Web Search API, or a developer-friendly alternative like Serper.dev.
- Web Scraping: BeautifulSoup, Requests, Scrapy.