

Project 10: AI Agents Fundamentals

Objective (Why?)

Build a simple AI agent that can use tools and reason through problems in 3 days. This project introduces AI agents using LangChain with a focus on practical, working examples rather than complex enterprise features. You will practice:

- Simple Agent Creation: Using LangChain to create a basic reasoning agent
- Tool Integration: Building 3-4 simple tools (calculator, web search, file reader)
- Basic Reasoning: Understanding how agents think through problems step-by-step
- User Interface: Creating a simple chat interface to interact with the agent

Core Requirements (Simplified)

Component	Requirement
Basic Agent	LangChain agent with OpenAI that can use tools and show reasoning
Simple Tools	Calculator, web search, file reader, and one custom tool
Chat Interface	Streamlit interface where users can talk to the agent
Reasoning Display	Show the agent's thought process to users

Development Approach: Milestone-Based Progression

Philosophy: Focus on deliverable quality and comprehensive review compliance rather than rigid timelines. Each milestone must pass all relevant review templates before proceeding.

Milestone 1: Agent Foundation & Basic Tools

Estimated Time: 4-6 hours (flexible based on learning pace)

Deliverables:

- LangChain agent setup with OpenAI integration
- Basic agent that can respond to questions and show reasoning
- Calculator tool implementation with mathematical operations
- Agent reasoning display in console with step-by-step process
- Basic error handling and agent conversation management

Review Requirements (Must Pass to Proceed):

- AI Integration Review: Agent reasoning quality and tool usage
- Architecture Review: Clean agent system design and modularity
- Security Review: Safe tool execution and input validation

Milestone 2: Advanced Tools & Multi-Step Logic

Estimated Time: 4-6 hours (flexible based on Milestone 1 completion)

Deliverables:

- Web search tool integration using DuckDuckGo or simple API
- File reading tool for text files with content processing
- Custom tool implementation (weather, joke generator, or domain-specific)
- Multi-step problem solving requiring multiple tools
- Enhanced error handling and edge case management

Review Requirements (Must Pass to Proceed):

- AI Integration Review: Multi-tool coordination and reasoning quality
- Performance Review: Tool execution efficiency and response times
- Security Review: Safe file access and web request handling

Milestone 3: User Interface & Production Features

Estimated Time: 4-6 hours (flexible based on previous milestones)

Deliverables:

Streamlit chat interface with agent interaction
Agent reasoning steps displayed to users in real-time
File upload capability for document analysis
Testing with non-technical users and UX improvements
Comprehensive documentation and usage examples

Review Requirements (Must Pass for Project Completion):

AI Integration Review: Complete agent system with user interface
Architecture Review: Scalable agent architecture and tool ecosystem
Code Quality Review: Production-ready code and documentation
Performance Review: Responsive user experience and tool performance

Milestone Progression Rules:

- Cannot advance to next milestone without passing all review requirements
- Flexible timing allows for learning at individual pace
- Quality gates ensure each milestone meets professional standards
- Mentor support available for concept clarification and review failures

Success Examples

- User: "What's 15% of 250, and can you search for current inflation rates?"
- Agent should use calculator for math, then web search for inflation data

Phase Progression Requirements

Project 10 → Project 11 Advancement Requirements

Mandatory Review Template Compliance:

AI Integration Review: Minimum 8.5/10 score (agent reasoning and tool usage)
Architecture Review: Minimum 8.0/10 score (agent system design)
Security Review: Minimum 8.5/10 score (safe tool execution)
Code Quality Review: Minimum 8.0/10 score (maintainable agent code)

```
advancement_requirements = {
```

```
"review_compliance": {  
    "ai_integration": {"minimum_score": 8.5, "weight": 40},  
    "architecture": {"minimum_score": 8.0, "weight": 25},  
    "security": {"minimum_score": 8.5, "weight": 25},  
    "code_quality": {"minimum_score": 8.0, "weight": 10}  
,  
    "functional_requirements": {  
        "multi_tool_coordination": True,  
        "reasoning_transparency": True,  
        "user_interface_functional": True  
    }  
}
```