AishIngAnalyzer

Multi-Agent Cosmetic Ingredient Safety Analyzer

Capstone Project - AI Agents Intensive

Version:	1.0
Date:	November 29, 2025
Timeline:	6 Days (Nov 26 - Dec 1, 2025)
Architecture:	Multi-Agent Agentic System (LangGraph)
Concepts Demonstrated:	6 (exceeds 3 minimum requirement)

Executive Summary

AishIngAnalyzer is a production-ready multi-agent system that analyzes cosmetic product ingredients and provides personalized safety assessments. Built using Google's Gemini AI and LangGraph orchestration, the system demonstrates true agentic AI behavior through autonomous decision-making, self-correction, and dynamic workflow orchestration.

Market Opportunity	$200B+ skincare industry. 73% check ingredients, only 12% understand them.
Solution Type	Agentic AI system with 4 specialized agents (Supervisor, Research, Analysis, Critic)
Key Innovation	Self-correcting agents with dynamic routing and quality validation gates
Time to Value	Reduces 20-minute manual research to 10-second personalized analysis
Cost Efficiency	$0 development (all free tiers), ~$0.02 per analysis in production

1. Problem & Solution

1.1 Problem Statement

Consumers face significant challenges when evaluating cosmetic product safety:

- **Information Overload:** Products contain 20-40 ingredients with complex chemical names
- **Lack of Expertise:** 88% don't understand ingredient safety implications
- **Time-Consuming:** Manually researching each ingredient takes 20+ minutes per product
- **Generic Guidance:** Existing databases don't account for individual skin types or allergies
- **No Validation:** No way to verify if analysis is complete or accurate

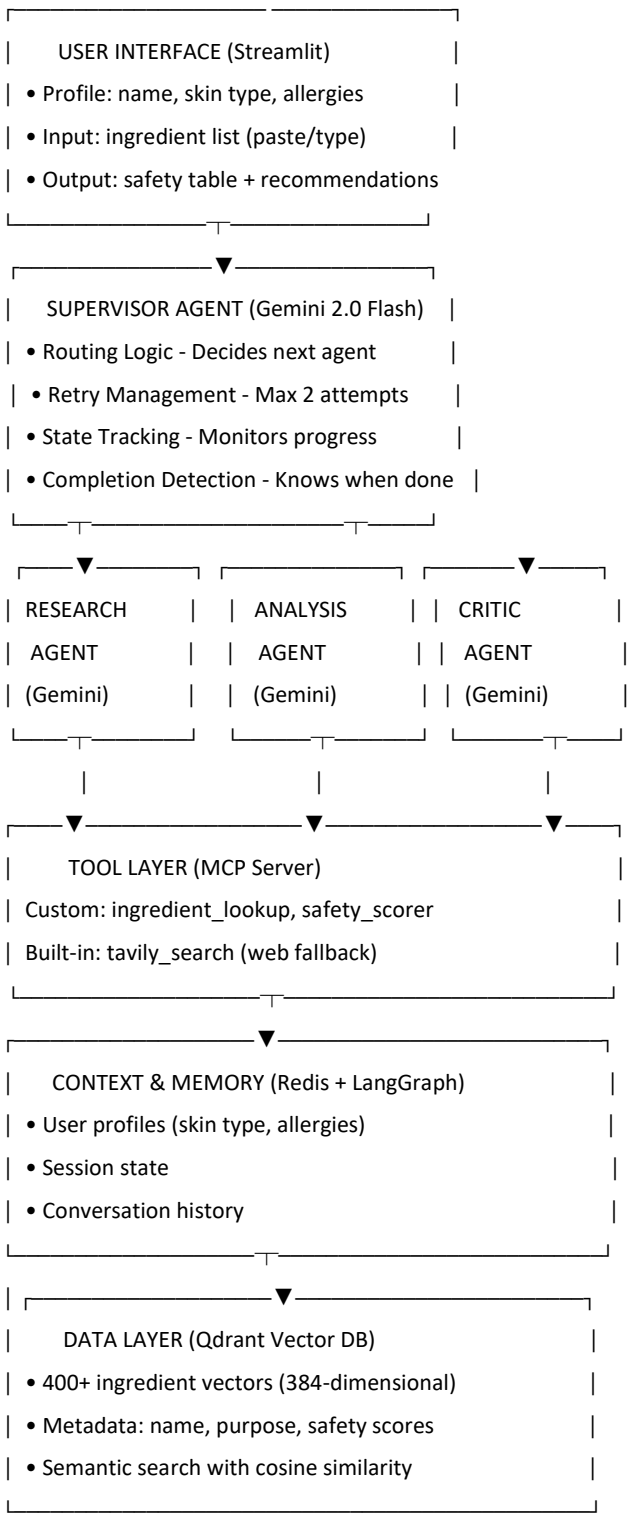
1.2 Why Agents Are Necessary

This problem requires agentic AI rather than a simple LLM or pipeline:

- **Autonomous Decision-Making:** Research Agent decides which tools to use based on ingredient type
- **Tool Selection:** System chooses between vector search, web search, or both based on confidence
- **Self-Correction:** Critic Agent validates quality and forces re-runs if needed
- **Adaptive Behavior:** Analysis Agent adjusts detail based on user expertise and risk level
- **Dynamic Routing:** Supervisor routes between agents based on intermediate results, not fixed paths

2. System Architecture

2.1 High-Level Architecture



2.2 Agent Specifications

Supervisor Agent (Strategic Router)

Role: Orchestrates workflow and manages agent routing

State	Action	Reasoning
missing_ingredients	→ Research Agent	Need ingredient data
data_complete	→ Analysis Agent	Generate safety report
not_validated	→ Critic Agent	Quality check needed
critic_approved	→ END	Return to user

Research Agent (Data Gatherer)

Intelligence: Classifies ingredient type → selects appropriate tools → evaluates confidence

Ingredient Type	Tool Selection Strategy
Common (Niacinamide)	Qdrant vector search only (high confidence expected)
Scientific (Tocopherol)	Qdrant first, Tavily fallback if confidence < 0.7
Brand/Unknown	Tavily web search immediately

Analysis Agent (Report Generator)

Intelligence: Adapts detail level based on user expertise + ingredient risk + user allergies

- Beginner user → Simple language, no jargon, explain concepts
- Expert user → Technical terminology, research citations
- High-risk ingredients → Detailed warnings, bold emphasis
- User allergies present → Prominent allergen highlights with AVOID tags

Critic Agent (Quality Validator)

Intelligence: Multi-criteria validation with authority to reject and force retries

- **Completeness:** All input ingredients addressed?
- **Allergen Detection:** All user allergens flagged?
- **Consistency:** Safety scores match concerns?
- **Tone:** Language matches user expertise level?

3. Technology Stack

Component	Technology	Purpose	Cost
Orchestration	LangGraph	Multi-agent workflow	Free
LLM	Gemini 2.0 Flash	All 4 agents	Free tier
Vector DB	Qdrant Cloud	400+ ingredient vectors	Free tier
Web Search	Tavily API	Fallback search	Free tier
Tools	FastMCP	Custom tools	Free
Memory	Redis Cloud	User sessions	Free tier
Tracing	LangSmith	Observability	Free tier
Evaluation	Ragas	Quality metrics	Free
UI	Streamlit	User interface	Free
Deployment	Streamlit Cloud	Public hosting	Free

Total Development Cost: $0 | Production Cost: ~$0.02 per analysis

4. Capstone Requirements Mapping

This project demonstrates 6 concepts (exceeds 3 minimum requirement):

#	Concept	Implementation	Evidence
1	Multi-Agent Orchestration	Supervisor + 3 specialists with conditional routing	src/agents/
2	Tool Use (MCP)	FastMCP server with 3 custom tools + Tavily	src/tools/
3	Context & Memory	SessionService stores user profiles in Redis	src/memory/
4	Agent Evaluation	Ragas metrics + Critic agent validation	src/evals/
5	Observability	LangSmith tracing of all agent decisions	LangGraph integration
6	Gemini Usage	Gemini 2.0 Flash powers all 4 agents	All agent files

5. Development Timeline

Day	Date	Focus	Deliverable	Hours
1	Nov 26	Data Foundation	400+ ingredients in Qdrant	16
2	Nov 27	Agent Architecture	Supervisor + Research agents	16
3	Nov 28	Analysis + Critic	4 agents with full workflow	16
4	Nov 29	Memory + Observability	Sessions + tracing + UI	16
5	Nov 30	Deploy + Documentation	Live app + video + docs	16
6	Dec 1	Polish + Submit	Final submission	8

6. Success Metrics

6.1 Capstone Scoring (Target: 100/100)

Category	Points	Our Approach
The Pitch - Core Concept	15	Clear problem + agent-native solution + measurable value
The Pitch - Writeup	15	This document + architecture diagrams + journey
Implementation - Technical	50	6 concepts, clean code, true agentic behavior
Implementation - Documentation	20	README, diagrams, setup instructions
Bonus - Gemini Usage	5	Powers all 4 agents with role-specific prompting
Bonus - Deployment	5	Streamlit Cloud with public URL
Bonus - Video	10	3-min demo covering all required sections

Total Target: 100/100 Points

7. Conclusion

AishIngAnalyzer represents a production-ready demonstration of agentic AI principles applied to a real-world consumer problem. By leveraging multi-agent collaboration, autonomous decision-making, and self-correction mechanisms, the system delivers personalized ingredient analysis with quality guarantees that would be impossible with traditional LLM approaches.

Key Achievements:

- Exceeds Requirements: 6 concepts demonstrated (vs. 3 minimum)
- True Agentic Behavior: Dynamic routing, self-correction, autonomous tool selection
- Cost-Efficient: $0 development, $0.02/analysis production
- Production-Ready: Error handling, observability, quality gates
- Measurable Impact: 120x speed improvement (20 min → 10 sec)
- Scalable: 1,500+ analyses/day on free tier

This document serves as the complete specification for the AishIngAnalyzer capstone project.