Autonomous Logistic Research Agent

Course Name: Agentic AI

Institution Name: Medicaps University – Datagami Skill Based Course

Submitted by:

Sr no	Student Name	Enrolment Number
1.	Kunal Singh Bhadouriya	EN22CS301539
2.	Komal Khanuja	EN22CS301511
3.	Kundan Singh	EN22CS301543
4.	Kartik Gawshinde	EN22CS301487
5.	Madhavi Patidar	EN23CS301559

Group Name: Group 05D5

Project Number: AAI-05

Industry Mentor Name:

University Mentor Name: Ass. Prof. Maya Baniya

Academic Year: 2026

Table of Contents

1. Introduction

The Autonomous Logistics Researcher Agent is an AI-powered Agentic system designed to automate logistics research and industry intelligence generation. The system leverages Large Language Models (LLMs), LangChain, and CrewAI to autonomously search, validate, analyze and summarize logistics-related information from multiple reliable sources.

Unlike traditional AI chat systems, this solution follows an Agentic AI approach where multiple specialized agents collaborate to perform structured research tasks. The system produces professional, structured research reports and stores them in a persistent knowledge repository for future retrieval and continuous learning.

Problem Statement

The logistics industry is rapidly evolving due to advancements in AI, automation, and digital technologies. However, logistics research is still largely manual, time-consuming, and fragmented across multiple sources, making it difficult to extract structured and actionable insights efficiently.

There is a need for an intelligent, autonomous system that can understand logistics-related queries, gather and analyze information from credible sources, generate structured research reports, and maintain a persistent knowledge repository to support faster and more informed decision-making.

1.1 Scope of the Document

This High-Level Design document covers:

- Overall system architecture and technology stack
- Major components and their responsibilities
- Information and process flows
- API design and endpoint catalogue
- Data model and storage strategy
- Authentication, session management, and CORS strategy
- Deployment architecture on Railway and Vercel
- Non-functional requirements: security and performance
- Known limitations and planned future enhancements

This document does not cover detailed low-level implementation code.

1.2 Intended Audience

This document is intended for:

- Faculty evaluators
- Industry mentors
- System architects
- Developers
- Project Stakeholders

1.3 System Overview

The system enables:

Strategic Intelligence Queries

- Autonomous Multi-Agent Processing
- Verified Research Insights
- Structured Knowledge Outputs

The system consists of a multi-agent orchestration engine where each agent performs a specialized task such as:

- Query interpretation
- Web search
- Source validation
- Data analysis
- Report generation
- Knowledge storage

It ensures accurate, scalable, and continuously improving logistics intelligence generation.

2. System Design

2.1 Application Design

The system follows a modular micro-component architecture:

Modules:

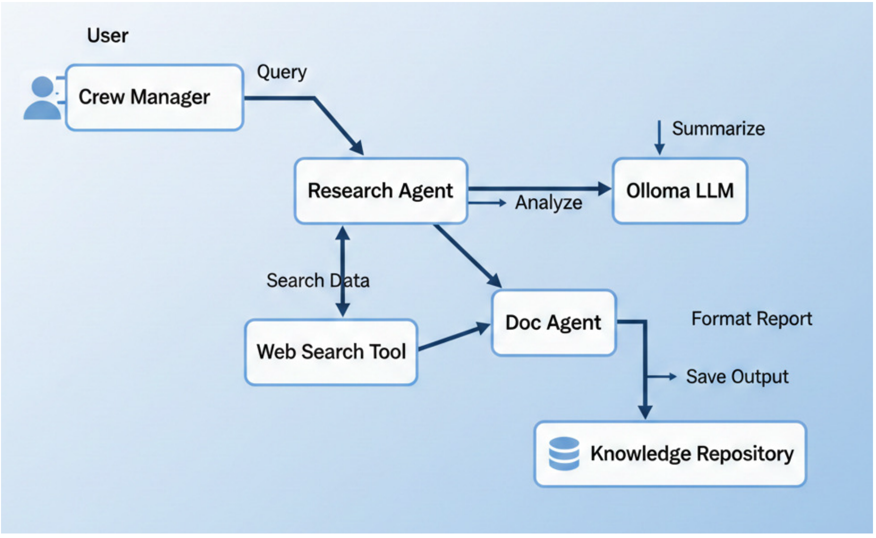
- Query Intake Module
- Orchestrator Engine
- Search & Retrieval Agent
- Source Validation Agent
- Analysis & Insight Engine
- Summary Generator
- Knowledge Storage Module

Each module is independently scalable and loosely coupled.

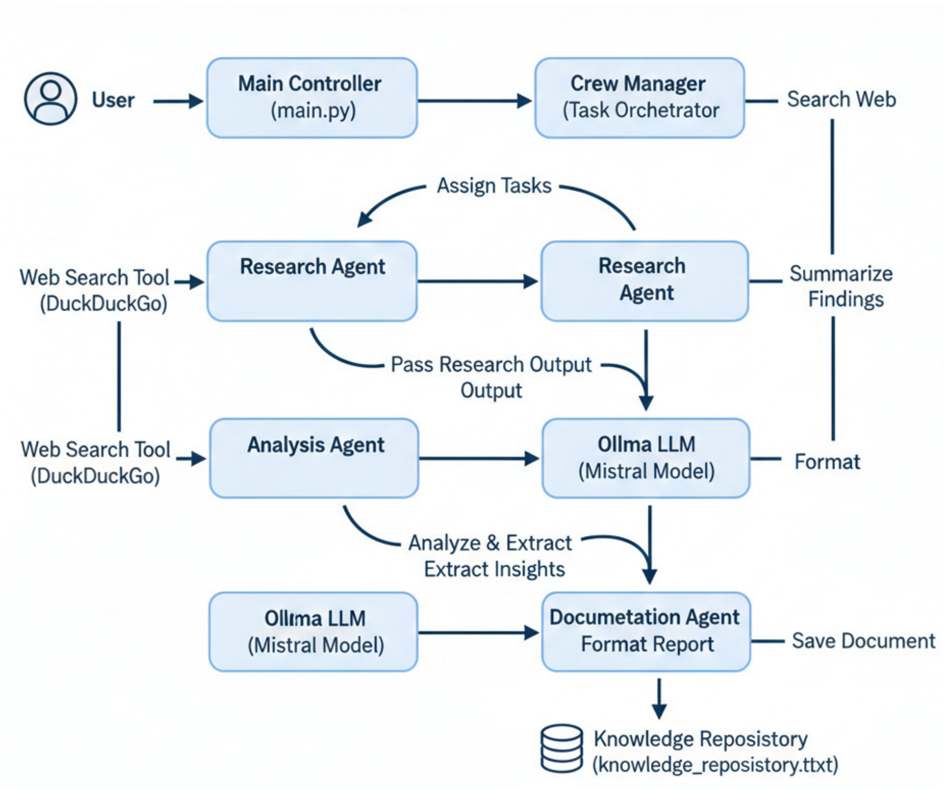
2.2 Process Flow

User Query
↓
Query Intake Module
↓
Orchestrator Engine
↓
Search & Retrieval Agent
↓
Source Validation Agent
↓
Analysis & Insight Engine
↓
Summary Generator
↓
Knowledge Storage Module

2.3 Information Flow



2.4 Components Design



2.5 Key Design Considerations

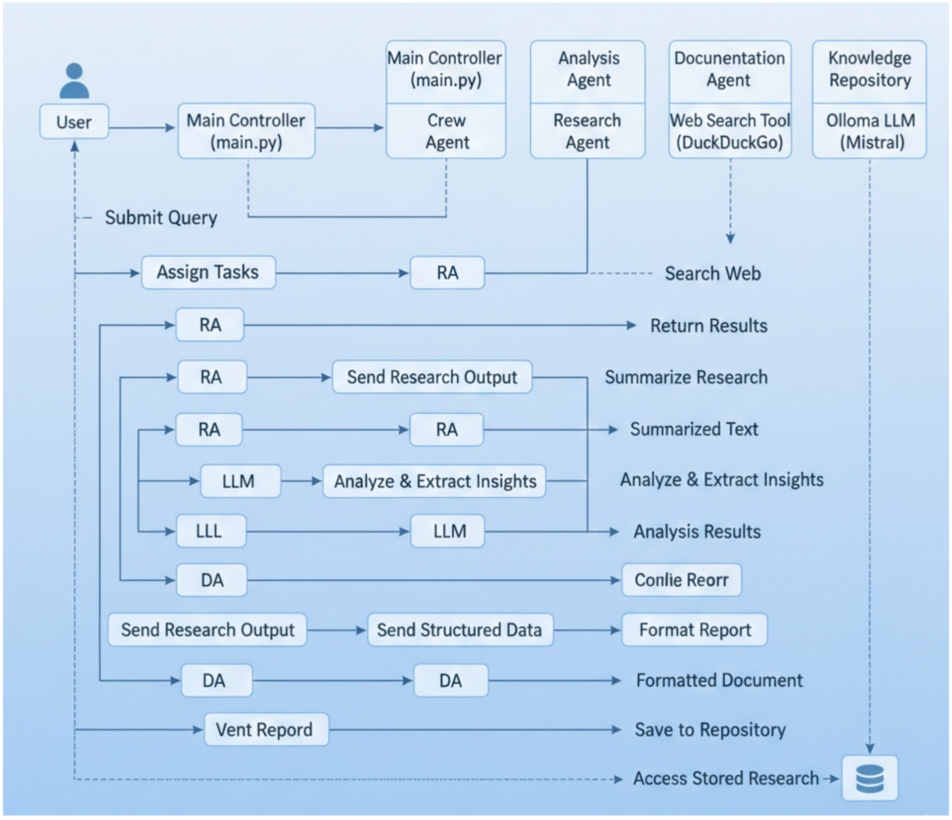
- Modular and scalable architecture
- Agent-based orchestration
- Source reliability validation
- Persistent knowledge storage
- Low hallucination through multi-source verification
- Extendable for real-time monitoring

2.6 API Catalogue

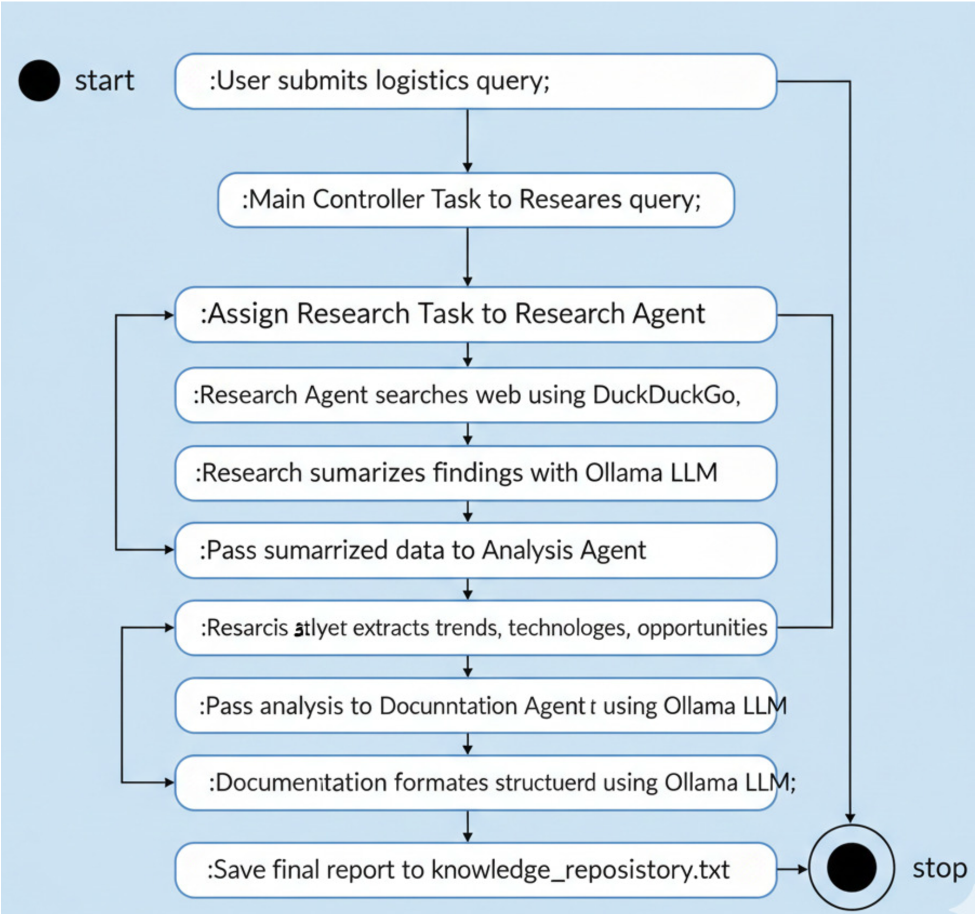
API	Purpose
OpenAI GPT API	Reasoning & summarization
Tavily Search API	Web research
News APIs	Industry news
Vector DB API	Memory storage

UML Diagrams:-

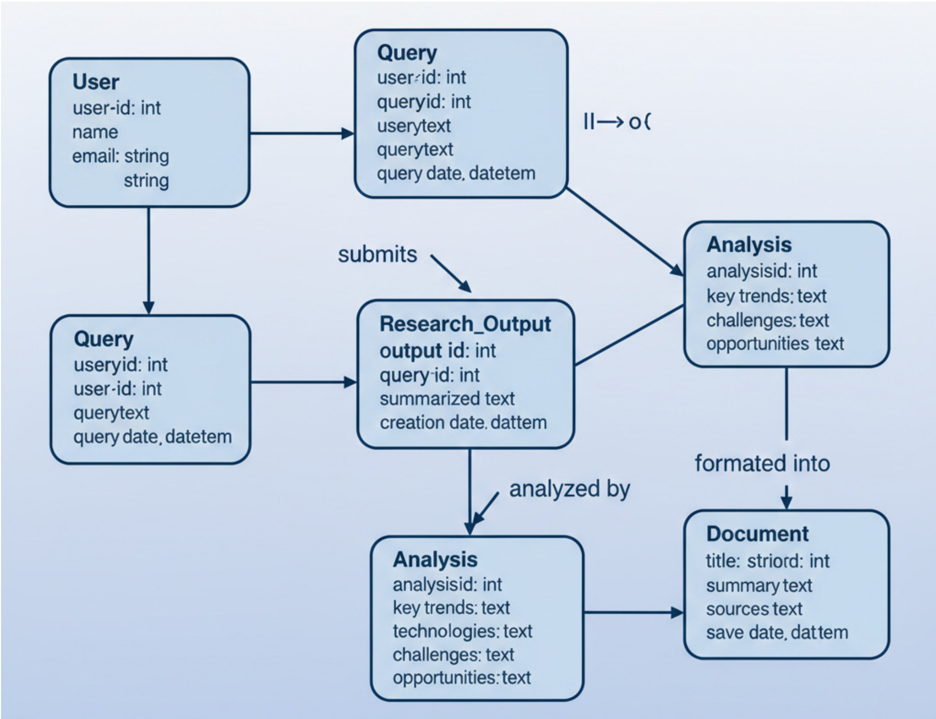
Sequence Diagram



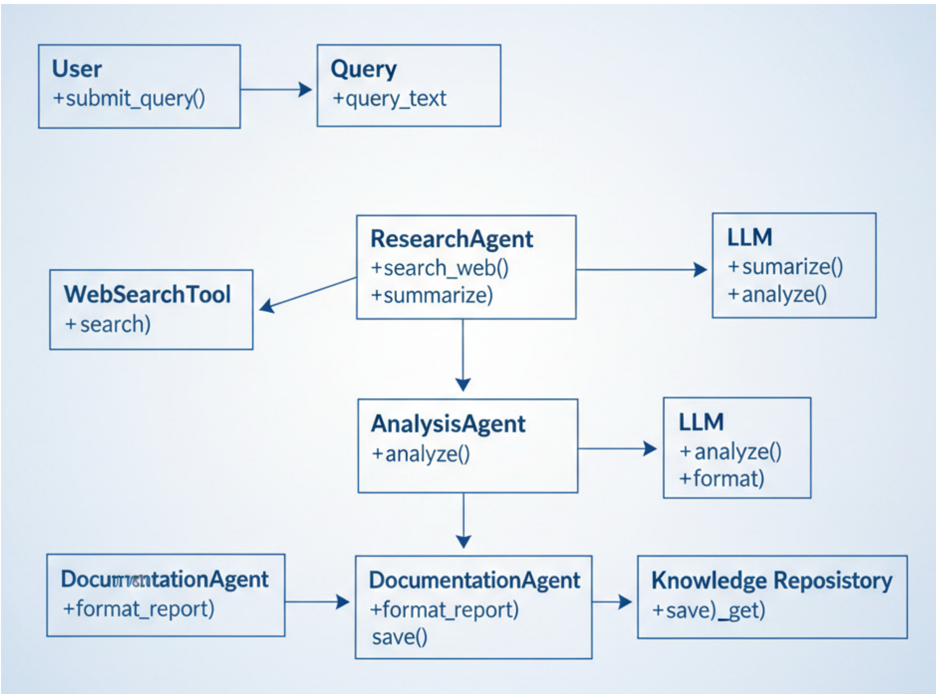
Activity Diagram



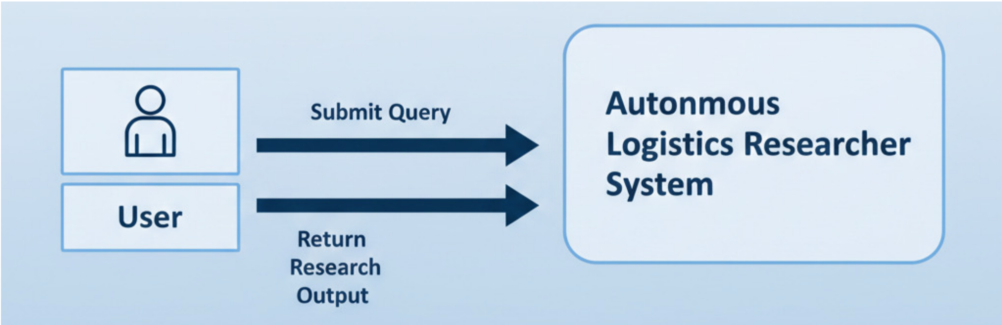
Entity Relationship Diagram



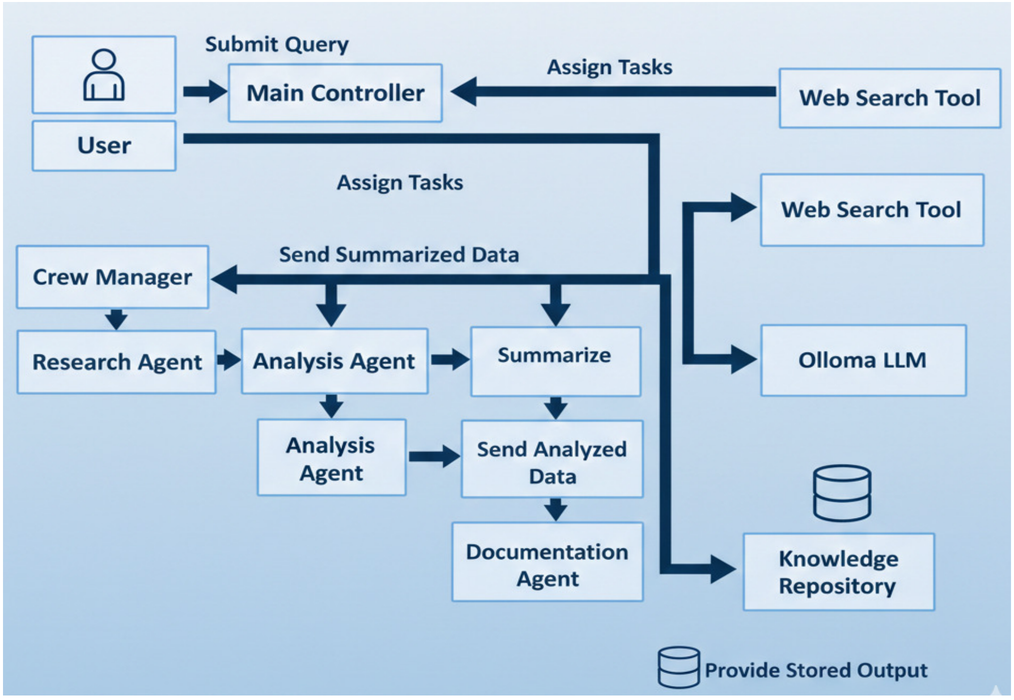
Class Diagram



Data Flow Diagram - Level 0

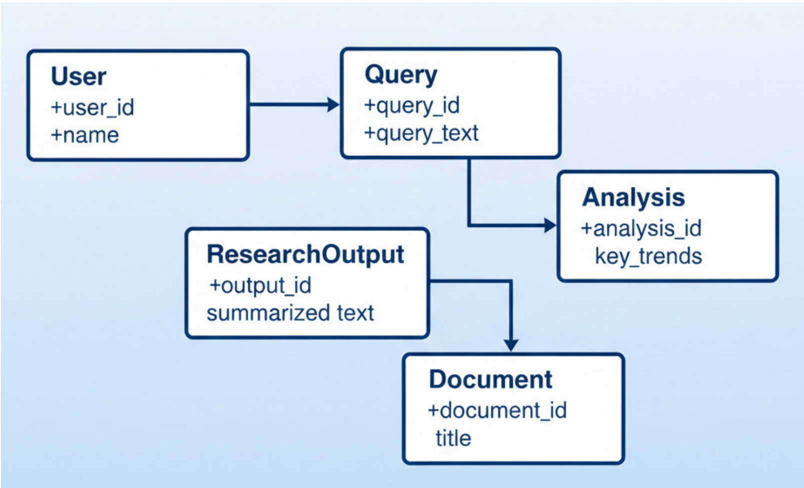


Data Flow Diagram – Level 1

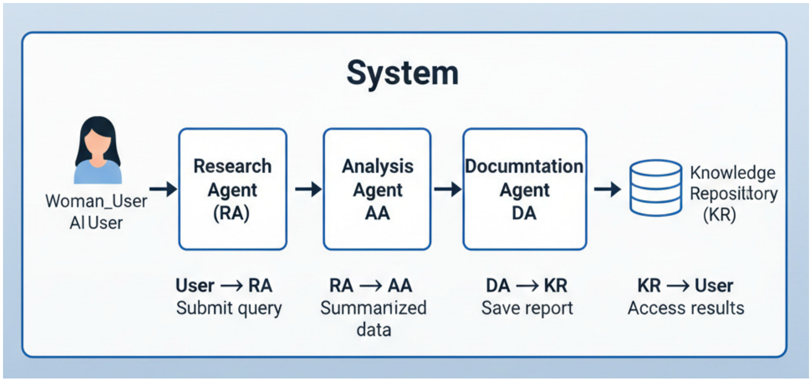


3. Data Design

3.1 Data Model



3.2 Data Access Mechanism

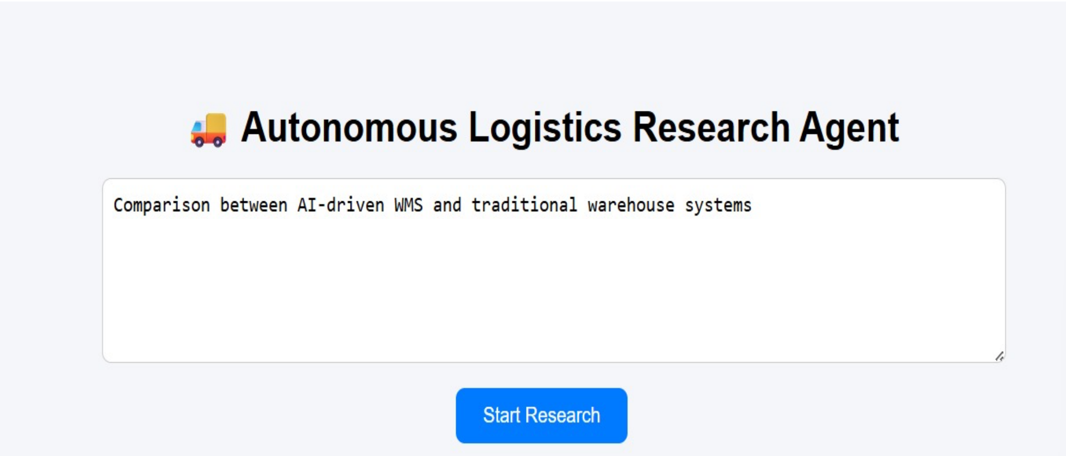


3.3 Data Retention Policies

- Research embeddings are stored per session in ChromaDB and remain until manually cleared.
- Chat and query history is retained until the user deletes it via the system API.
- Guest session data is not linked to a user ID and may be automatically cleaned in the future.
- No sensitive personal data is stored beyond Firebase-managed email and UID.

4. Interfaces

- **Research Input Interface**



- **Research Output Interface**

5. State and Session Management

- Each query treated as independent session
- Conversation memory stored using vector embeddings
- Session context maintained during execution

6. Caching & Future Optimizations

- Frequently searched queries cached
- Vector embeddings reused
- Reduced API calls
- Improved response time

7. Deployment Architecture

Service	Platform	Details
Frontend	Vercel	Auto-deploys on git push to main; preview URL per PR; SPA fallback via vercel.json
Backend	Render	Python 3.11 auto-detected; start command via Procfile; root directory: /backend

8. Non-Functional Requirements

- High reliability
- Low latency response
- Scalable agent architecture
- Modular design
- Fault tolerance
- Easy maintainability

9. Known Limitations & Future Enhancements

- **Known Limitations:**
  Limited scalability, basic security, partial feature implementation, minimal error handling, and limited testing coverage.
- **Future Enhancements:**
  Improve performance, add advanced security, enhance UI/UX, expand features, integrate more APIs, and implement automation with CI/CD.

10. References

- OpenAI API Documentation
- LangChain Documentation
- CrewAI Documentation
- Tavily API Documentation
- Research papers on Agentic AI systems