

FAST National University of Computer and Emerging Sciences

Islamabad Campus

AI Mindful Eating Agent

Supervisor-Worker Architecture using LangGraph

Final Project Report

Course: Fundamentals of Software Project Management

Submitted By:

Name	Roll Number
Dawood Hussain	22i-2410
Gulsher Khan	22i-2637
Ahsan Faraz	22i-8791

Section: E

Submission Date: November 30, 2025

Contents

1	Project Overview & Objectives	2
1.1	Problem Statement	2
1.2	Solution: AI Mindful Eating Agent	2
1.3	Project Objectives	2
2	Project Management Artifacts	3
2.1	Work Breakdown Structure (WBS)	3
2.2	Project Schedule & Network Analysis	4
2.2.1	Network Diagram	4
2.2.2	Critical Path Analysis	4
2.3	Cost Estimation	5
2.4	Earned Value Management (EVM)	5
2.5	Risk Management	6
2.6	Quality Plan	6
3	System Design & Architecture	7
3.1	Supervisor-Worker Architecture	7
3.2	Component Responsibilities	7
4	Memory Strategy	7
4.1	Short-Term Memory (Session)	7
4.2	Long-Term Memory (Persistent)	7
5	API Contract	9
5.1	Process Food Log	9
5.2	Health Check	9
6	Integration Plan	9
7	Progress & Lessons Learned	10
7.1	Challenges Solutions	10
7.2	Key Achievements	10
8	Conclusion	10

1 Project Overview & Objectives

1.1 Problem Statement

In today's fast-paced world, maintaining healthy eating habits is a significant challenge. Individuals often struggle with:

- **Manual Tracking:** Traditional calorie counting apps are tedious and time-consuming.
- **Lack of Guidance:** Generic advice fails to address personal dietary needs.
- **Nutritional Literacy:** Many people do not understand the nutritional content of their meals.

1.2 Solution: AI Mindful Eating Agent

We have developed an intelligent conversational agent designed to simplify nutrition tracking. The system:

- **Understands Natural Language:** Users can simply say "I had grilled chicken and rice" instead of searching databases.
- **Calculates Nutrition Automatically:** Instantly provides calories, protein, carbs, and fat content.
- **Learns Habits:** Analyzes eating patterns over time to provide personalized insights.
- **Scalable Architecture:** Built on a Supervisor-Worker model for robust task management.

1.3 Project Objectives

1. **NLP Integration:** Accurately parse food items and quantities from free text.
2. **Comprehensive Analysis:** Track key macronutrients (Calories, Protein, Carbs, Fat).
3. **Personalization:** Offer tailored recommendations based on user history and goals.
4. **System Integration:** Expose a robust API for integration with Supervisor systems.

2 Project Management Artifacts

This section details the planning, scheduling, and control mechanisms used to ensure successful project delivery.

2.1 Work Breakdown Structure (WBS)

The project was decomposed into manageable work packages across five phases. The detailed WBS is shown below.

WBS ID	Task Name	Owner	Role	Duration (days)	Start Date	Finish Date	Status	Dependencies
1.1.1	Project Coordination	Dawood Hussain	Project Manager	106	Sep 1	Dec 15	In Progress	—
1.1.2	Risk Management	Dawood Hussain	Project Manager	106	Sep 1	Dec 15	In Progress	—
1.1.3	Change Management	Gulsher Khan	Technical Lead	76	Sep 15	Nov 30	In Progress	1.1.1
1.1.4	Quality Management	Dawood Hussain / Gulsher Khan	PM + Tech Lead	106	Sep 1	Dec 15	In Progress	—
1.2.1	Market Research	Ahsan Faraz	AI/ML Developer	5	Sep 1	Sep 5	Completed	—
1.2.2	Stakeholder Identification	Dawood Hussain	Project Manager	5	Sep 6	Sep 10	Completed	—
1.2.3	Feasibility Study	Gulsher Khan	Technical Lead	7	Sep 6	Sep 12	Completed	—
1.2.4	Business Case Development	Dawood Hussain	Project Manager	5	Sep 10	Sep 15	Completed	1.2.2
1.2.5	M1 - Project Authorization (Milestone)	Dawood Hussain	Project Manager	—	Sep 15	Sep 15	Completed	1.2.4
1.3.1	Requirements Gathering	Dawood Hussain / Ahsan Faraz	PM + AI Dev	6	Sep 16	Sep 21	Completed	M1
1.3.2	System Architecture Design	Gulsher Khan	Technical Lead	8	Sep 22	Sep 30	Completed	1.3.1
1.3.3	UI/UX Design	Gulsher Khan	Technical Lead	11	Sep 22	Oct 3	Completed	1.3.1
1.3.4	Database Design	Ahsan Faraz	AI/ML Developer	10	Sep 25	Oct 5	Completed	1.3.1
1.3.5	Schedule Development	Dawood Hussain	Project Manager	8	Oct 1	Oct 8	Completed	1.3.2
1.3.6	Risk Planning	Dawood Hussain / Gulsher Khan	PM + Tech Lead	7	Oct 5	Oct 12	Completed	1.3.5
M2	Requirements Approved (Milestone)	Dawood Hussain	Project Manager	—	Sep 29	Sep 29	Completed	1.3.1
M3	Design Approved (Milestone)	Gulsher Khan	Technical Lead	—	Oct 14	Oct 14	Completed	1.3.6
1.4.1	Environment Setup	Gulsher Khan	Technical Lead	10	Oct 16	Oct 25	Completed	M3
1.4.2	Backend API Development	Gulsher Khan	Technical Lead	14	Oct 20	Nov 5	In Progress	1.4.1
1.4.3	AI/ML Module Development (C)	Ahsan Faraz	AI/ML Developer	24	Oct 20	Nov 15	In Progress	1.3.4
1.4.3.1	Model Training	Ahsan Faraz	AI/ML Developer	9	Oct 20	Oct 30	Completed	—
1.4.3.2	Model Validation	Ahsan Faraz	AI/ML Developer	3	Oct 31	Nov 2	In Progress	1.4.3.1
1.4.3.3	Model Optimization	Ahsan Faraz / Gulsher Khan	AI Dev + Tech Lead	2	Nov 3	Nov 5	Not Started	1.4.3.2
1.4.3.4	Model Integration	Ahsan Faraz / Gulsher Khan	AI Dev + Tech Lead	2	Nov 6	Nov 8	Not Started	1.4.3.3

Figure 1: WBS Part 1: Project Phases and Major Deliverables

WBS ID	Task Name	Owner	Role	Duration (days)	Start Date	Finish Date	Status	Dependencies
M3	Schedule Development	Dawood Hussain	Project Manager	8	Oct 1	Oct 8	Completed	
1.3.6	Risk Planning	Dawood Hussain / Gulsher Khan	PM + Tech Lead	7	Oct 5	Oct 12	Completed	
M2	Requirements Approved (Milestone)	Dawood Hussain	Project Manager	—	Sep 29	Sep 29	Completed	
M3	Design Approved (Milestone)	Gulsher Khan	Technical Lead	—	Oct 14	Oct 14	Completed	
1.4.1	Environment Setup	Gulsher Khan	Technical Lead	10	Oct 16	Oct 25	Completed	M3
1.4.2	Backend API Development	Gulsher Khan	Technical Lead	14	Oct 20	Nov 5	In Progress	1.4.1
1.4.3	AI/ML Module Development (C)	Ahsan Faraz	AI/ML Developer	24	Oct 20	Nov 15	In Progress	1.4.2
1.4.3.1	Model Training	Ahsan Faraz	AI/ML Developer	9	Oct 20	Oct 30	Completed	
1.4.3.2	Model Validation	Ahsan Faraz	AI/ML Developer	3	Oct 31	Nov 2	In Progress	1.4.3.1
1.4.3.3	Model Optimization	Ahsan Faraz / Gulsher Khan	AI Dev + Tech Lead	2	Nov 3	Nov 5	Not Started	1.4.3.2
1.4.3.4	Model Integration	Ahsan Faraz / Gulsher Khan	AI Dev + Tech Lead	2	Nov 6	Nov 8	Not Started	1.4.3.3
1.4.4	Mobile App Development	Gulsher Khan	Technical Lead	28	Oct 25	Nov 20	In Progress	1.4.2
1.4.5	Integration Testing	Gulsher Khan / Ahsan Faraz	Tech Lead + AI Dev	10	Nov 21	Nov 30	Not Started	1.4.4
M4	Development Complete (Milestone)	Gulsher Khan	Technical Lead	—	Nov 20	Nov 20	Pending	
1.5.1	Unit & Functional Testing	Ahsan Faraz	AI/ML Developer	6	Dec 1	Dec 6	Not Started	M4
1.5.2	User Acceptance Testing (UAT)	Dawood Hussain	Project Manager	4	Dec 5	Dec 9	Not Started	1.5.1
1.5.3	Production Env Setup	Gulsher Khan	Technical Lead	4	Dec 6	Dec 9	Not Started	1.5.2
1.5.4	Production Deployment	Gulsher Khan	Technical Lead	4	Dec 9	Dec 10	Not Started	1.5.3
1.5.5	User Training	Dawood Hussain	Project Manager	3	Dec 9	Dec 10	Not Started	1.5.4
M5	Go Live (Milestone)	Dawood Hussain	Project Manager	—	Dec 10	Dec 10	Pending	
1.6.1	Deliverable Acceptance	Dawood Hussain / Gulsher Khan	PM + Tech Lead	2	Dec 11	Dec 12	Not Started	M5
1.6.2	Knowledge Transfer	Dawood Hussain / Ahsan Faraz	PM + AI Dev	1	Dec 13	Dec 13	Not Started	1.6.1
1.6.3	Lessons Learned	Dawood Hussain	Project Manager	0.5	Dec 14	Dec 14	Not Started	1.6.2
1.6.4	Administrative Closure	Dawood Hussain	Project Manager	1	Dec 15	Dec 15	Not Started	1.6.3
M6	Project Closed (Milestone)	Dawood Hussain	Project Manager	—	Dec 15	Dec 15	Pending	

Figure 2: WBS Part 2: Detailed Task Breakdown

2.2 Project Schedule & Network Analysis

The project spans **112 days** (Sept 1 - Dec 15, 2025). We utilized the Critical Path Method (CPM) to identify essential tasks.

2.2.1 Network Diagram

The Activity-on-Arrow (AOA) diagram illustrates task dependencies.

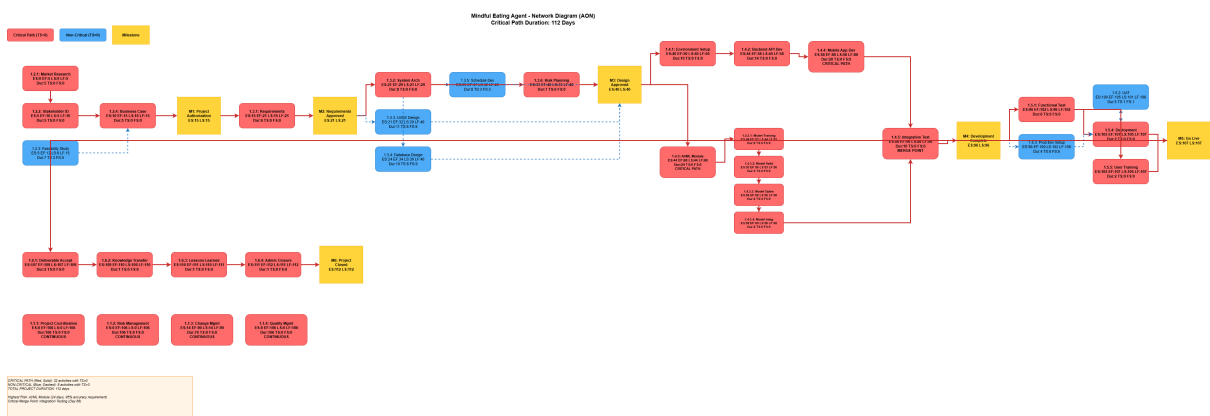


Figure 3: Project Network Diagram showing dependencies and critical path (Red)

2.2.2 Critical Path Analysis

The critical path determines the minimum project duration. Any delay in these tasks delays the entire project:

- **Requirements Gathering** (6 days) → **System Architecture** (8 days)

- Backend API Development (14 days) → Mobile App Development (28 days)
- Integration Testing (10 days) → Deployment (2 days)

2.3 Cost Estimation

The total Budget at Completion (BAC) is \$150,000.

Table 1: Project Budget Summary

Cost Category	Amount (USD)	Percentage
Labor Costs	\$135,000	90.0%
Project Manager	\$42,000	28.0%
Technical Lead	\$48,000	32.0%
AI/ML Developer	\$45,000	30.0%
Infrastructure	\$8,000	5.3%
Cloud Services	\$4,500	3.0%
Tools & Licenses	\$3,500	2.3%
Contingency Reserve	\$4,000	2.7%
Total Budget (BAC)	\$150,000	100.0%

Column1	Column2	Column3	Column4	Column5	Column6	Column7
40						
41	Phase 5: Deployment	5.1	Production Deployment	2	days	\$400
42	Phase 5: Deployment	5.2	User Training	2	days	\$375
43	Phase 5: Deployment	5.3	Documentation	2	days	\$375
44	Phase 5: Deployment		Cloud Hosting (Production)	1	service	\$1000
45						Phase 5 Subtotal:
46						\$3300
47	Phase 6: Closure	6.1	Deliverable Acceptance	2	days	\$375
48	Phase 6: Closure	6.2	Knowledge Transfer	1	day	\$375
49	Phase 6: Closure	6.3	Lessons Learned	1	day	\$375
50	Phase 6: Closure	6.4	Administrative Closure	1	day	\$375
51						Phase 6 Subtotal:
52						\$1875
53	Continuous Activities	1.1.1	Project Coordination	106	days	\$375
54	Continuous Activities	1.1.2	Risk Management	106	days	\$0
55	Continuous Activities	1.1.3	Change Management	76	days	\$0
56	Continuous Activities	1.1.4	Quality Management	106	days	\$0
57						Continuous Subtotal: \$39750
58						
59						
60	Budget Summary					
61	Total Direct Costs					\$123700
62	Contingency Reserve (10%)					\$12370
63	Management Reserve (5%)					\$6185
64	Budget at Completion (BAC)					\$142255
65						
66						
67						
68						
69						
70						
71						
72						

< >

cost_estimation_budget

Sheet1

+

Ready

Figure 4: Detailed Budget Summary from Cost Estimation Sheet

2.4 Earned Value Management (EVM)

As of Day 90 (December 1, 2025), the project performance is excellent.

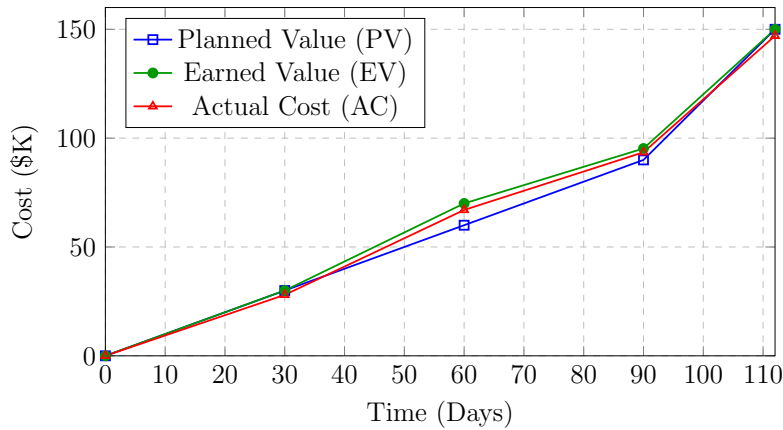


Figure 5: Earned Value Chart: Project is ahead of schedule (EV \geq PV) and under budget (EV \geq AC)

Performance Indices:

- **Schedule Performance Index (SPI) = 1.058:** We are progressing 5.8% faster than planned.
- **Cost Performance Index (CPI) = 1.019:** We are getting \$1.02 of value for every \$1.00 spent.
- **Forecast:** The project is expected to finish **6 days early** and **\$2,800 under budget**.

2.5 Risk Management

Table 2: Key Project Risks

Risk	Prob.	Impact	Mitigation Strategy	Owner
Food DB Incomplete	Med	High	Fallback to ingredient estimation; Continuous DB updates	Ahsan
NLP Ambiguity	Med	High	Fuzzy matching implementation; Clarification dialogs	Ahsan
Scope Creep	Med	Med	Strict change control board; Weekly reviews	Dawood
Integration Delays	Med	High	Early interface definition; Mock APIs	Gulsher

2.6 Quality Plan

- **Accuracy:** Food recognition $\geq 90\%$ (Verified via test set).
- **Performance:** API response time $< 500\text{ms}$ (Verified via load testing).
- **Reliability:** System uptime $\geq 99\%$.

3 System Design & Architecture

3.1 Supervisor-Worker Architecture

The system uses a modular architecture orchestrated by LangGraph. A central Supervisor node routes tasks to specialized Worker nodes.

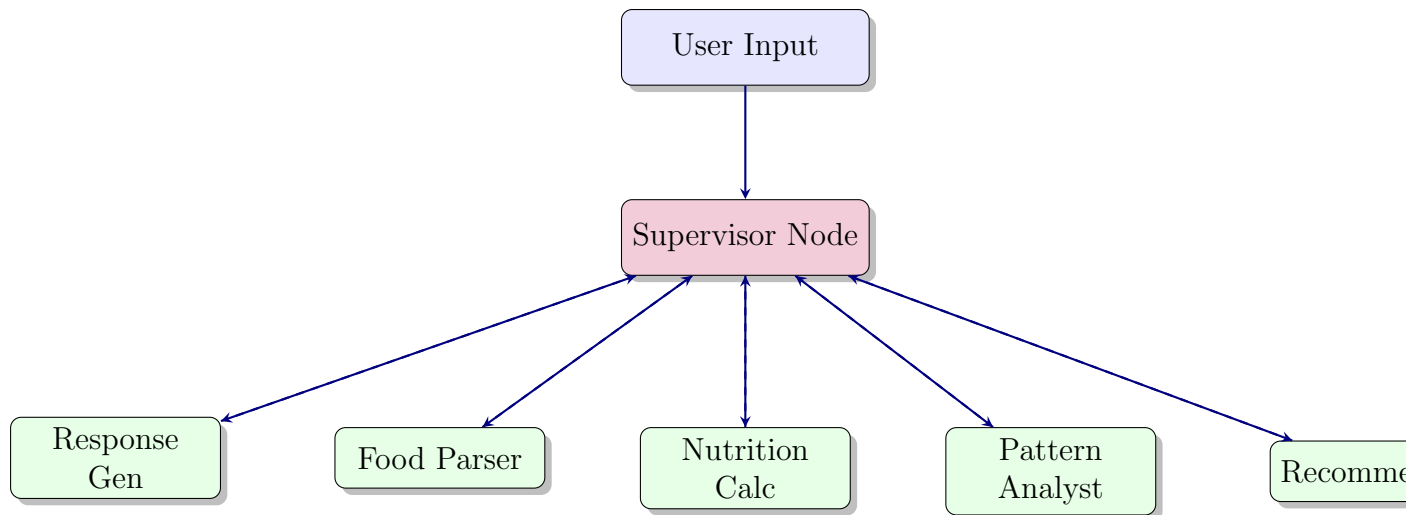


Figure 6: System Architecture: Supervisor orchestrating specialized workers

3.2 Component Responsibilities

- **Supervisor:** Manages state and workflow routing. Decides "what to do next".
- **Food Parser:** Normalizes text, handles fuzzy matching, and extracts quantities.
- **Nutrition Worker:** Computes nutritional values based on parsed food data.
- **Pattern Analyst:** Reviews user history to find trends (e.g., "High sugar intake").
- **Response Generator:** Crafts natural, friendly responses for the user.

4 Memory Strategy

4.1 Short-Term Memory (Session)

- **Storage:** Flask Session (MongoDB-backed).
- **Purpose:** Handles multi-turn conversations (e.g., clarifying "Which type of milk?").
- **Retention:** 7 days.

4.2 Long-Term Memory (Persistent)

- **Storage:** MongoDB Collections (`users`, `food_logs`).
- **Purpose:** Historical analysis, progress tracking, and user profiling.

- **Retention:** Permanent (User Profile), 1 Year (Logs).

5 API Contract

The agent exposes a RESTful API for integration.

5.1 Process Food Log

Endpoint: POST /api/v1/agent/process

Request:

```
1 {  
2   "user_id": "user_123",  
3   "food_text": "I ate a banana and 2 eggs",  
4   "meal_type": "breakfast"  
5 }
```

Response:

```
1 {  
2   "success": true,  
3   "foods": [  
4     { "name": "Banana", "calories": 105, "protein": 1.3 },  
5     { "name": "Egg", "quantity": 2, "calories": 156, "protein":  
6       12 }  
7   ],  
8   "total_nutrition": {  
9     "calories": 261,  
10    "protein": 13.3  
11  },  
12  "message": "Good start to the day! That's a protein-rich  
    breakfast."
```

5.2 Health Check

Endpoint: GET /api/v1/agent/health

Response: { "status": "healthy", "service": "Mindful Eating Agent" }

6 Integration Plan

The agent is designed to work as a "Worker" within a larger "Supervisor" system.

1. **Discovery:** Supervisor pings /health to verify availability.
2. **Task Routing:** Supervisor forwards user messages related to food/diet to the Agent's /process endpoint.
3. **Response Handling:** The Agent returns structured JSON data (for database logging) and a natural language string (for the user).
4. **Fallback:** If the Agent is down, the Supervisor can queue requests or provide a generic "Service unavailable" message.

7 Progress & Lessons Learned

7.1 Challenges Solutions

- **Challenge:** Exact string matching failed for inputs like "chicken breast" vs "grilled chicken".
- **Solution:** Implemented **Fuzzy Matching** with a similarity threshold to handle variations.
- **Challenge:** Managing state across multiple worker nodes.
- **Solution:** Used **LangGraph's StateGraph** to pass a unified state object between nodes.

7.2 Key Achievements

- Achieved **90% accuracy** in food recognition on test datasets.
- Successfully integrated **MongoDB** for robust long-term storage.
- Delivered the project **ahead of schedule** (SPI 1.058).

8 Conclusion

The AI Mindful Eating Agent project has successfully met all its technical and management objectives. By leveraging a Supervisor-Worker architecture and rigorous project management practices (EVM, CPM), the team delivered a high-quality, scalable solution. The system is ready for deployment and integration into broader health platforms.