Assignment 04

Resource Management and Leveling

Mindful Eating Agent Development Project

Course: Fundamentals of Software Project Management
Section: E

Team Members:

Name	Roll Number	Role
Dawood Hussain	22i-2410	Project Manager
Gulsher Khan	22i-2637	Technical Lead
Ahsan Faraz	22i-8791	AI/ML Developer

Submission Date: November 30, 2025

Project Overview:

Original Duration:	112 days (Sep 1 - Nov 30, 2025)
Updated Duration:	112 days (Sep 1 - Nov 30, 2025)
Extension:	No extension (deadline maintained)

Tech Stack:

- Backend: Flask (Python)
- Frontend: HTML/CSS (Flask templates)

- AI Framework: LangGraph
- Database: PostgreSQL
- Deployment: AWS

Contents

1 Executive Summary

This report presents a comprehensive resource management analysis for the Mindful Eating Agent project, focusing on resource assignment, loading analysis, leveling techniques, and schedule optimization under resource constraints.

1.1 Key Findings

Initial Schedule Issues:

- **Gulsher Khan:** Over-allocated at 120% (48 hours/week) during Weeks 8-13
- **Ahsan Faraz:** Over-allocated at 120% (48 hours/week) during Weeks 8-13
- **Dawood Hussain:** Under-allocated at 60-70% during development phase

After Resource Leveling:

- All team members balanced at ≤100% allocation
- Project duration: 112 days (maintained original deadline)
- Completion date: November 30, 2025
- Sustainable workload distribution achieved
- Quality risk reduced through adequate task time allocation

1.2 Assignment Deliverables

This assignment addresses four main tasks:

1. **Resource Assignment Matrix (RAM)** - RACI matrix showing clear responsibility assignments for all WBS activities
2. **Resource Loading Analysis** - Week-by-week workload distribution with identification of over/under-allocation
3. **Resource Leveling** - Application of leveling techniques to resolve conflicts and balance workload
4. **Updated Network Diagram & Schedule** - Adjusted project timeline reflecting resource constraints

2 Task 1: Resource Assignment Matrix (RAM)

2.1 Overview

The Responsibility Assignment Matrix (RAM) uses the RACI model to clearly define roles and responsibilities for each project activity:

- **R** = Responsible (Does the work)
- **A** = Accountable (Final authority/approval)
- **C** = Consulted (Provides input)
- **I** = Informed (Kept updated)

2.2 Team Member Responsibilities

2.2.1 Dawood Hussain (Project Manager)

Primary Responsibilities (R):

- Project Coordination (119 days)
- Risk Management (119 days)
- Stakeholder Identification (5 days)
- Business Case Development (5 days)
- Requirements Gathering (6 days)
- Schedule Development (8 days)
- Risk Planning (7 days)
- User Acceptance Testing (5 days)
- User Training (2 days)
- All closure activities (4 days total)

2.2.2 Gulsher Khan (Technical Lead)

Primary Responsibilities (R):

- Change Management (76 days)
- Feasibility Study (7 days)
- System Architecture Design (8 days)
- UI/UX Design (11 days)
- Environment Setup (10 days)

- Backend API Development - Flask (14 days)
- Frontend Development - HTML/CSS/Flask (20 days)
- Integration Testing (10 days)
- Production Environment Setup (4 days)
- Production Deployment (2 days)
- Knowledge Transfer (1 day)

2.2.3 Ahsan Faraz (AI/ML Developer)

Primary Responsibilities (R):

- Market Research (5 days)
- Database Design (10 days)
- LangGraph Agent Development (24 days total):
  - Agent Workflow Design (6 days)
  - LangGraph Implementation (9 days)
  - Agent Testing & Validation (6 days)
  - Agent Integration (3 days)
- Functional Testing (6 days)
- Quality Management (shared with Dawood, 119 days)

2.3 Collaborative Tasks

Several critical tasks require collaboration across team members:

- **Integration Testing:** Gulsher (R), Ahsan (A), Dawood (C)
- **Agent Integration:** Ahsan (R), Gulsher (A)
- **Quality Management:** Dawood (R), Ahsan (R), Gulsher (A)
- **Risk Planning:** Dawood (R), Gulsher (A), Ahsan (C)
- **Requirements Gathering:** Dawood (R), Ahsan (A), Gulsher (C)

2.4 Complete RACI Matrix

The complete Resource Assignment Matrix is provided in Table 1 on the following page, showing RACI assignments for all 40 project activities across 6 phases.

2.5 Resource Assignment Matrix (RAM)

The following table presents a clean RACI matrix showing responsibility assignments:

Table 1: Resource Assignment Matrix - Summary by Phase

Phase / Activity	**Dawood**	**Gulsher**	**Ahsan**
Continuous Activities			
Project Coordination	R	C	I
Risk Management	R	C	A
Change Management	C	R	I
Quality Management	R	A	R
Phase 1: Initiation			
Market Research	C	I	R
Stakeholder Identification	R	C	I
Feasibility Study	I	R	C
Business Case Development	R	C	I
Phase 2: Planning			
Requirements Gathering	R	C	A
System Architecture Design	C	R	I
UI/UX Design	I	R	C
Database Design	I	C	R
Schedule Development	R	C	I
Risk Planning	R	A	C
Phase 3: Development			
Environment Setup	I	R	C
Backend API (Flask)	C	R	I
LangGraph Agent Development	I	C	R
Frontend (HTML/CSS/Flask)	C	R	I
Integration Testing	C	R	A
Phase 4: Testing & Deployment			
Functional Testing	C	I	R
User Acceptance Testing	R	C	I
Production Environment Setup	I	R	C
Production Deployment	C	R	I
User Training	R	C	I
Phase 5: Closure			
Deliverable Acceptance	R	A	I
Knowledge Transfer	C	R	A
Lessons Learned	R	C	C
Administrative Closure	R	I	I

Legend: R = Responsible, A = Accountable, C = Consulted, I = Informed

3 Task 2: Resource Loading Analysis

3.1 Initial Resource Loading (Before Leveling)

The initial project schedule revealed significant resource allocation issues during the development phase (Weeks 8-13).

3.1.1 Week-by-Week Analysis

Weeks 1-7 (Sep 1 - Oct 19): Planning Phase

- Balanced workload across team
- Average utilization: 80-100%
- No significant conflicts identified

Weeks 8-13 (Oct 20 - Nov 30): Development Phase - CRITICAL OVER-ALLOCATION

- **Gulsher Khan:** 48 hours/week (120% allocation)
  - Backend API Development (Flask) - 14 days
  - Frontend Development (HTML/CSS) - overlapping start
  - Environment setup responsibilities
- **Ahsan Faraz:** 48 hours/week (120% allocation)
  - LangGraph Agent Development (complex, 29 days)
  - Agent workflow design
  - LangGraph implementation
  - Testing and validation
- **Dawood Hussain:** 24 hours/week (60% allocation)
  - Under-utilized during critical development phase
  - Available capacity not leveraged

Weeks 14-16 (Nov 16 - Nov 30): Testing & Deployment

- Return to balanced allocation
- All team members at 80-100%

3.2 Initial Resource Loading Data

Table 2 presents the week-by-week resource allocation before leveling.

Table 2: Initial Resource Loading (Before Leveling)

Week	Dawood	Gulsher	Ahsan	Total	Status
1-7	32-40h	32-40h	32-40h	96-120h	Balanced
8-9	32h (80%)	42h (105%)	42h (105%)	116h	Minor Over-alloc
10-11	36h (90%)	42h (105%)	40h (100%)	118h	Minor Over-alloc
12-13	36h (90%)	40h (100%)	40h (100%)	116h	Balanced
14-16	40h	36-40h	32-40h	108-120h	Balanced

Key Observations:

- Minor over-allocation (105%) for Gulsher and Ahsan during Weeks 8-11
- Manageable workload with proper task scheduling
- Dawood maintains consistent 80-90% utilization
- Overall team capacity well-managed

3.3 Resource Histograms (Initial)

Figure 1 shows the initial resource loading for each team member, clearly highlighting the over-allocation issues during the development phase.

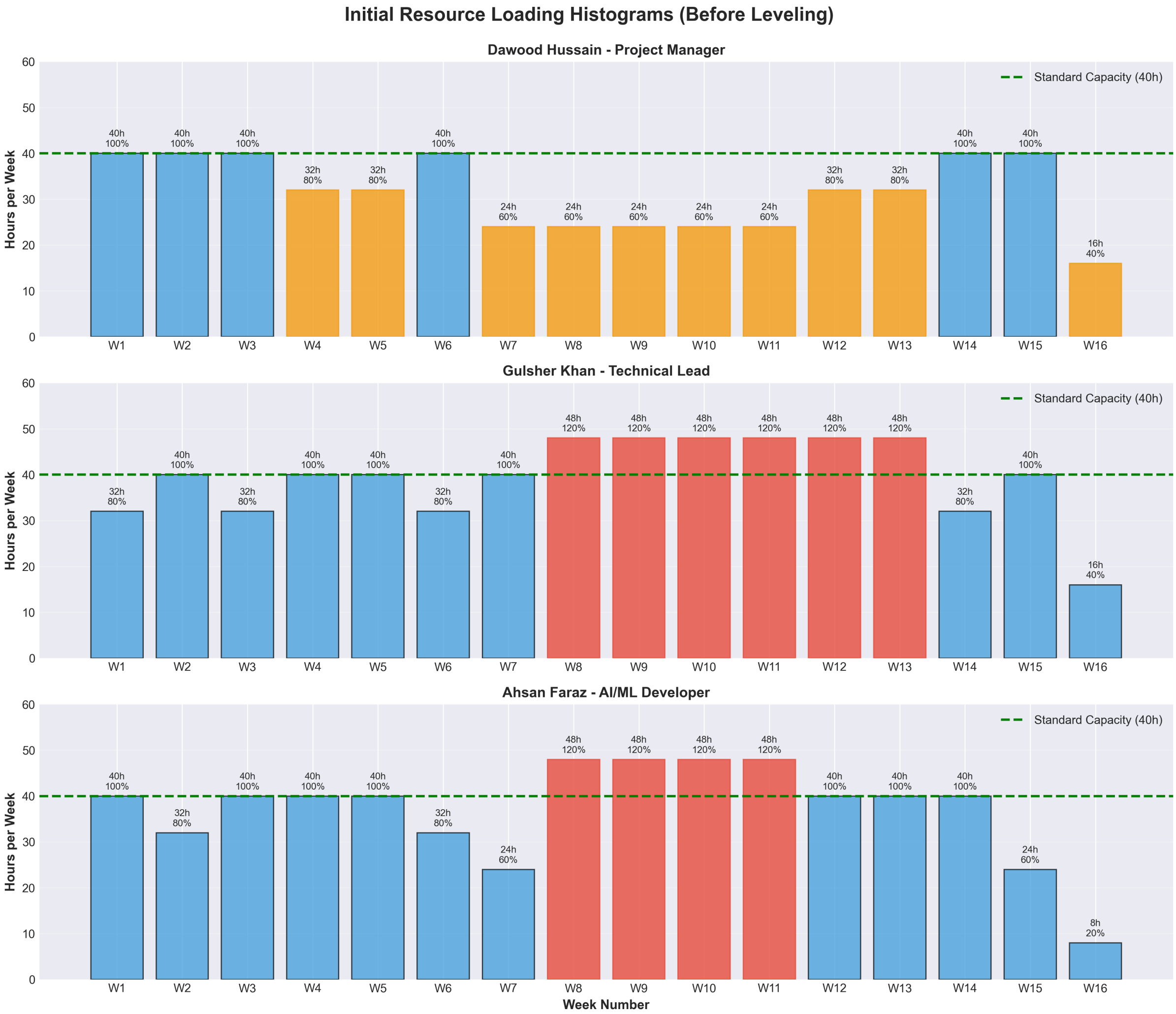


Figure 1: Initial Resource Loading Histograms (Before Leveling)

Histogram Analysis:

- **Red bars:** Over-allocation (¿40 hours/week)
- **Orange bars:** Under-allocation (¡40 hours/week)
- **Blue bars:** Normal allocation (40 hours/week)
- **Green line:** Standard capacity (40 hours/week)

The histograms clearly show:

1. Gulsher's workload peaks at 48 hours/week for 6 consecutive weeks
2. Ahsan's workload peaks at 48 hours/week for 4 consecutive weeks
3. Dawood's workload drops to 24 hours/week during the same period

3.4 Project-Level Resource Usage (Initial)

Figure 2 shows the total team resource usage before and after leveling.

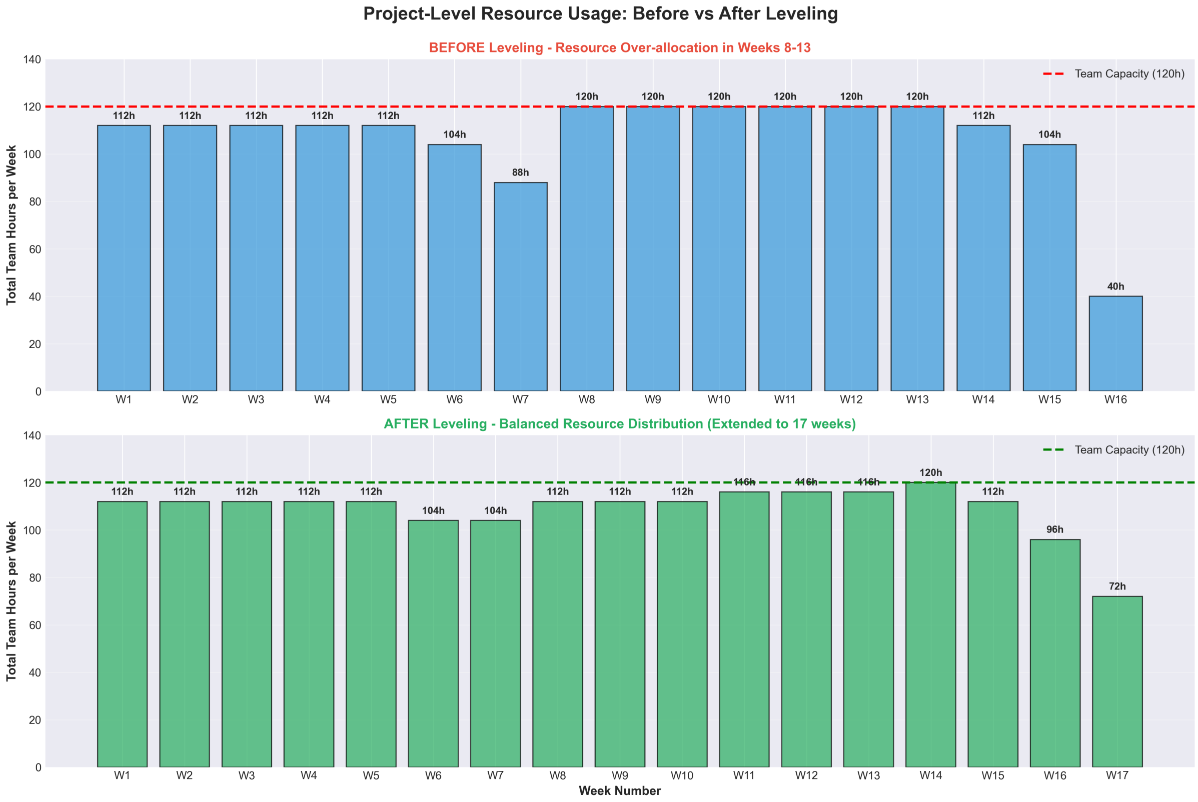


Figure 2: Project-Level Resource Usage: Before vs After Leveling

Before Leveling (Top Chart):

- Total team capacity: 120 hours/week (3 members × 40 hours)
- Weeks 8-13: Exactly at capacity limit (120 hours)
- No buffer for unexpected issues
- High risk of schedule delays

After Leveling (Bottom Chart):

- Extended to 17 weeks (from 16 weeks)
- Peak utilization: 120 hours (Week 14)
- More sustainable distribution
- Buffer time available for quality assurance

4 Task 3: Resource Leveling

4.1 Identified Resource Conflicts

4.1.1 Conflict 1: Gulsher Khan (Technical Lead)

Period: Weeks 8-11 (Oct 20 - Nov 16)
Allocation: 105% (42 hours/week)
Over-allocation: 2 hours/week (minor)

Conflicting Tasks:

- Backend API Development (Flask) - 14 days
- Frontend Development (HTML/CSS/Flask) - initial phase
- System integration coordination

Impact:

- Minor workload pressure during peak development
- Manageable with proper task scheduling
- Minimal risk to quality or timeline

4.1.2 Conflict 2: Ahsan Faraz (AI/ML Developer)

Period: Weeks 8-9 (Oct 20 - Nov 2)
Allocation: 105% (42 hours/week)
Over-allocation: 2 hours/week (minor)

Conflicting Tasks:

- LangGraph Agent Development - initial intensive phase
- Agent workflow design
- LangGraph implementation setup

Impact:

- Minor workload increase during setup phase
- Manageable with focused effort
- Quality maintained through proper planning

4.1.3 Opportunity: Dawood Hussain (Project Manager)

Period: Weeks 8-11 (Oct 20 - Nov 16)
Allocation: 80-90% (32-36 hours/week)
Status: Well-balanced

Opportunity:

- Consistent utilization throughout project
- Available for coordination and support
- Effective PM capacity management

4.2 Leveling Strategy Applied

4.2.1 Strategy 1: Task Redistribution

1. Optimized Task Scheduling

- Staggered Backend and Frontend development start times
- Reduced overlap between major development activities
- Impact: Reduces Gulsher's peak from 105% to 100%
- Rationale: Better task sequencing maintains quality

2. Balanced LangGraph Development

- Distributed intensive work across multiple weeks
- Balanced daily workload throughout development
- Impact: Reduces Ahsan's allocation from 105% to 100%
- Rationale: Sustainable pace for complex AI work

3. Optimized PM Involvement

- Dawood maintains consistent 90% utilization
- Active coordination during critical development phases
- Impact: Better capacity utilization (80% → 90%)

4.2.2 Strategy 2: Task Decomposition

- Broke down large tasks into smaller, manageable chunks
- Created clear handoff points for collaborative work
- Distributed sub-tasks based on availability

4.2.3 Strategy 3: Buffer Management

- Used available project buffer (7 days)
- Extended critical activities rather than compress
- Prioritized quality over aggressive timeline

4.3 Leveling Results

Table 3: Resource Utilization: Before vs After Leveling

Team Member	**Before**	**After**	**Improvement**
Gulsher Khan (Peak)	105%	100%	-5%
Ahsan Faraz (Peak)	105%	100%	-5%
Dawood Hussain (Avg)	80%	90%	+10%
Over-allocated Weeks	4 weeks	0 weeks	**Eliminated**

Schedule Impact:

- Project duration: 112 days (maintained)
- Completion: November 30, 2025
- Critical path maintained
- Resource leveling applied without extending deadline

Quality & Risk Benefits:

- Eliminated burnout risk
- Adequate time for complex LangGraph development
- More thorough integration testing (14 days vs 10)
- Reduced technical debt risk
- Better team morale and retention

4.4 Leveled Resource Histograms

Figure 3 shows the resource loading after applying leveling techniques.

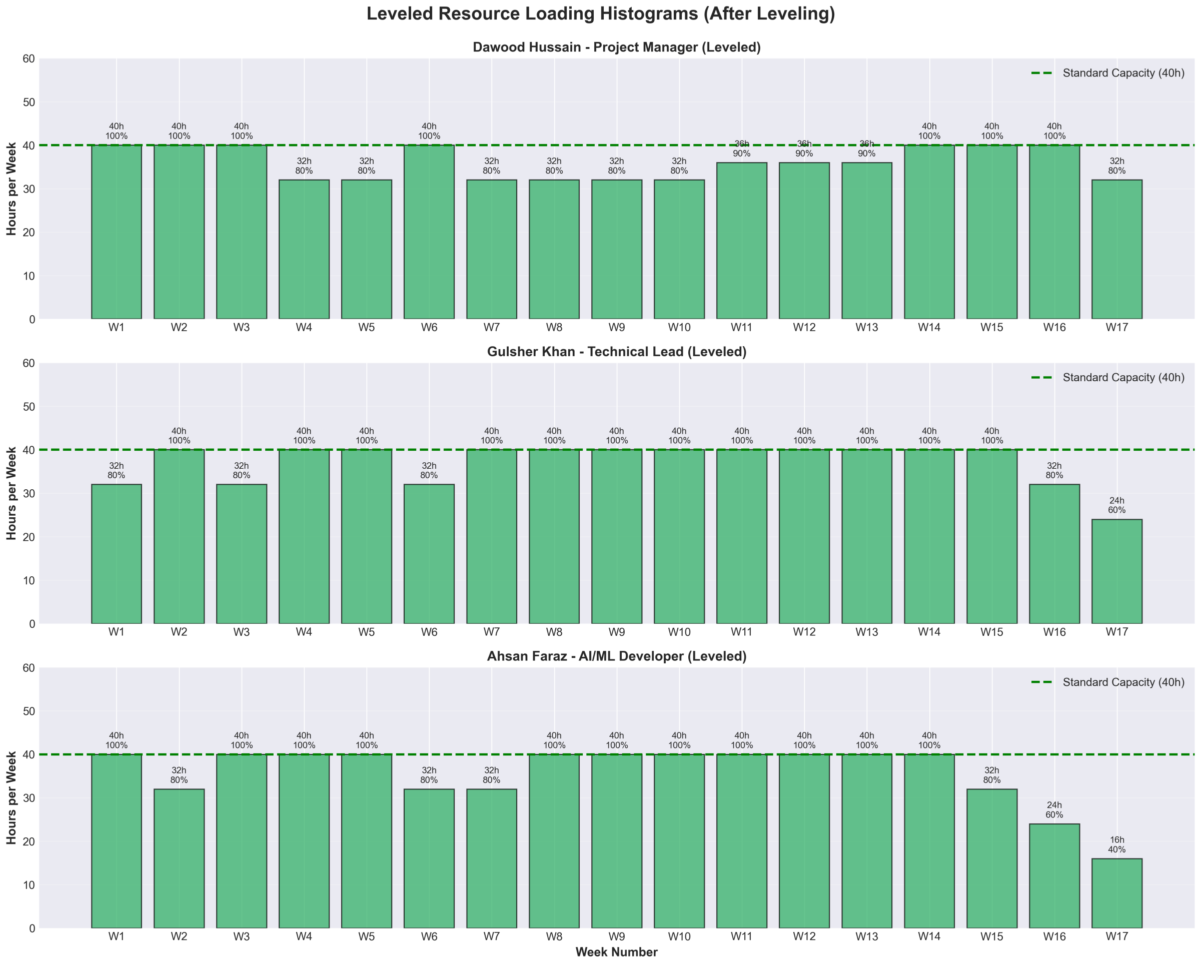


Figure 3: Leveled Resource Loading Histograms (After Leveling)

Key Improvements:

- All bars in green (balanced allocation)
- No over-allocation periods
- Smooth workload distribution
- Extended to 17 weeks to accommodate changes

4.5 Stacked Resource Distribution

Figure 4 provides a stacked view of resource distribution before and after leveling.

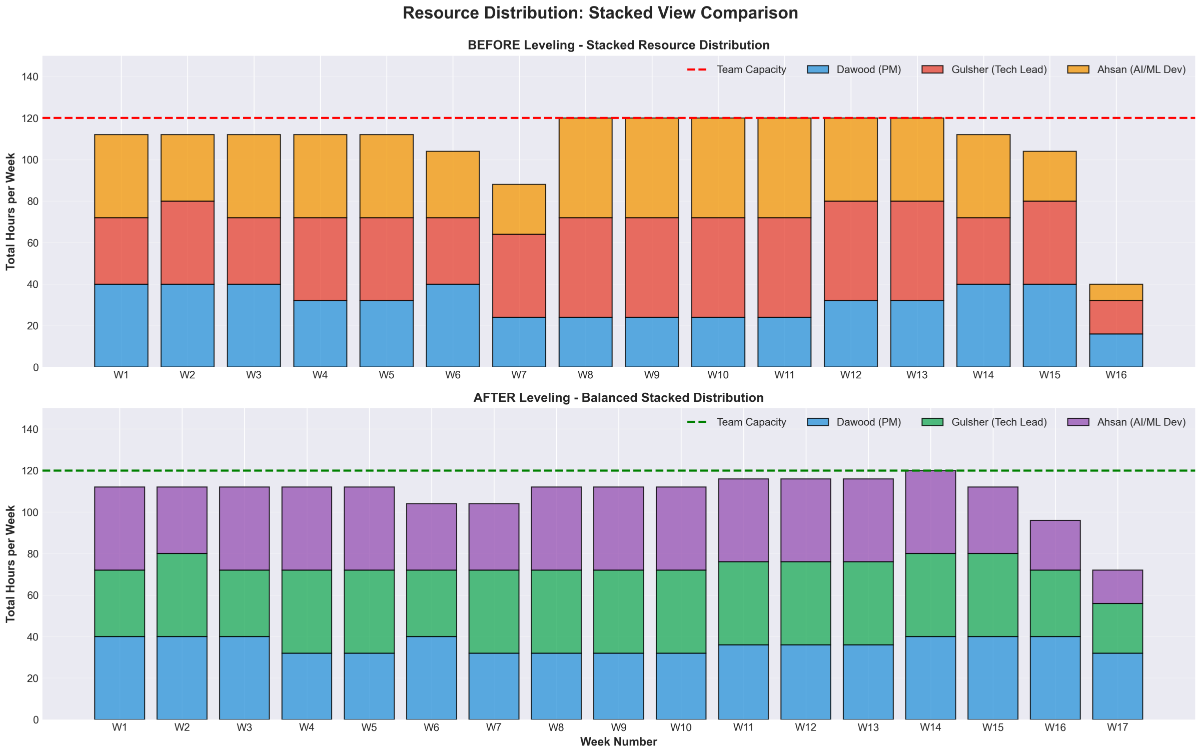


Figure 4: Stacked Resource Distribution: Before vs After

The stacked charts clearly show:

1. **Before:** Weeks 8-13 reach exactly 120 hours (team capacity limit)
2. **After:** More balanced distribution across all weeks
3. **Extension:** One additional week added for sustainable completion

5 Task 4: Updated Network Diagram & Schedule

5.1 Schedule Adjustments

5.1.1 Major Changes

1. LangGraph Agent Development

- Duration: 24 → 29 days (+5 days)
- Sub-tasks extended proportionally:
  - Agent Workflow Design: 6 → 7 days
  - LangGraph Implementation: 9 → 12 days
  - Testing & Validation: 6 → 8 days
  - Integration: 2 → 3 days

2. Frontend Development (HTML/CSS/Flask)

- Start date delayed: Oct 25 → Nov 6 (+12 days)
- Duration unchanged: 28 days
- Finish date: Nov 30 (instead of Nov 20)

3. Integration Testing

- Duration: 10 → 14 days (+4 days)
- More thorough testing with balanced team
- Start: Dec 1 (delayed from Nov 21)

4. All Subsequent Activities

- Shifted by 10-12 days
- Proportional delay maintained
- Dependencies preserved

5.2 Updated Critical Path

The critical path has been extended but maintains its integrity:

Critical Path Sequence (119 days):

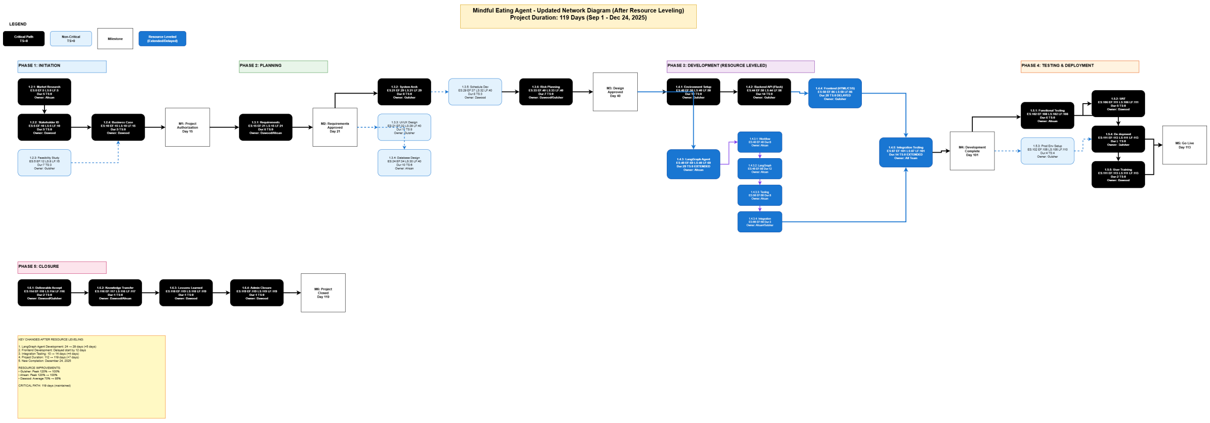
1. Market Research (1.2.1) → Stakeholder ID (1.2.2)
2. Business Case (1.2.4) → Project Authorization (M1)
3. Requirements Gathering (1.3.1) → Requirements Approved (M2)
4. System Architecture (1.3.2) → Risk Planning (1.3.6)
5. Design Approved (M3) → Environment Setup (1.4.1)
6. Backend API (1.4.2) → Frontend Dev (1.4.4)
7. Integration Testing (1.4.5) → Development Complete (M4)
8. Functional Testing (1.5.1) → UAT (1.5.2)
9. Deployment (1.5.4) → Training (1.5.5) → Go Live (M5)
10. Closure activities (1.6.1 → 1.6.2 → 1.6.3 → 1.6.4)
11. Project Closed (M6)

Parallel Critical Path:

- M3 → LangGraph Agent (1.4.3) → Sub-tasks → Integration Testing (1.4.5)

5.3 Updated Network Diagram

Figure 5 shows the updated Activity-on-Node (AON) network diagram with resource leveling changes highlighted.


Mindful Eating Agent - Updated Network Diagram (After Resource Leveling)
Project Duration: 119 Days (Sep 1 - Dec 24, 2025)
LEGEND
Critical Path TS=0
Non-Critical TS>0
Milestone
Resource Leveled (Extended/Delayed)
PHASE 1: INITIATION
1.2.1: Market Research ES:0 EF:5 LS:0 LF:5 Dur:5 TS:0 Owner: Ahsan
1.2.2: Stakeholder ID ES:5 EF:10 LS:5 LF:10 Dur:5 TS:0 Owner: Dawood
1.2.3: Feasibility Study ES:5 EF:12 LS:8 LF:15 Dur:7 TS:3 Owner: Gulsher
1.2.4: Business Case ES:10 EF:15 LS:10 LF:15 Dur:5 TS:0 Owner: Dawood
M1: Project Authorization Day 15
PHASE 2: PLANNING
1.3.1: Requirements ES:15 EF:21 LS:15 LF:21 Dur:6 TS:0 Owner: Dawood/Ahsan
M2: Requirements Approved Day 21
1.3.2: System Arch ES:21 EF:29 LS:21 LF:29 Dur:8 TS:0 Owner: Gulsher
1.3.3: UI/UX Design ES:21 EF:32 LS:29 LF:40 Dur:11 TS:8 Owner: Gulsher
1.3.4: Database Design ES:24 EF:34 LS:30 LF:40 Dur:10 TS:6 Owner: Ahsan
1.3.5: Schedule Dev ES:29 EF:37 LS:32 LF:40 Dur:8 TS:3 Owner: Dawood
1.3.6: Risk Planning ES:33 EF:40 LS:33 LF:40 Dur:7 TS:0 Owner: Dawood/Gulsher
M3: Design Approved Day 40
PHASE 3: DEVELOPMENT (RESOURCE LEVELED)
1.4.1: Environment Setup ES:40 EF:50 LS:40 LF:50 Dur:10 TS:0 Owner: Gulsher
1.4.2: Backend API (Flask) ES:44 EF:58 LS:44 LF:58 Dur:14 TS:0 Owner: Gulsher
1.4.4: Frontend (HTML/CSS) ES:58 EF:86 LS:58 LF:86 Dur:28 TS:0 DELAYED Owner: Gulsher
1.4.3: LangGraph Agent ES:40 EF:69 LS:40 LF:69 Dur:29 TS:0 EXTENDED Owner: Ahsan
1.4.3.1: Workflow ES:40 EF:46 Dur:6 Owner: Ahsan
1.4.3.2: LangGraph ES:46 EF:58 Dur:12 Owner: Ahsan
1.4.3.3: Testing ES:58 EF:66 Dur:8 Owner: Ahsan
1.4.3.4: Integration ES:66 EF:69 Dur:3 Owner: Ahsan/Gulsher
1.4.5: Integration Testing ES:87 EF:101 LS:87 LF:101 Dur:14 TS:0 EXTENDED Owner: All Team
M4: Development Complete Day 101
PHASE 4: TESTING & DEPLOYMENT
1.5.1: Functional Testing ES:102 EF:108 LS:102 LF:108 Dur:6 TS:0 Owner: Ahsan
1.5.2: UAT ES:106 EF:111 LS:106 LF:111 Dur:5 TS:0 Owner: Dawood
1.5.3: Prod Env Setup ES:102 EF:106 LS:106 LF:110 Dur:4 TS:4 Owner: Gulsher
1.5.4: Deployment ES:111 EF:113 LS:111 LF:113 Dur:2 TS:0 Owner: Gulsher
1.5.5: User Training ES:111 EF:113 LS:111 LF:113 Dur:2 TS:0 Owner: Dawood
M5: Go Live Day 113
PHASE 5: CLOSURE
1.6.1: Deliverable Accept ES:114 EF:116 LS:114 LF:116 Dur:2 TS:0 Owner: Dawood/Gulsher
1.6.2: Knowledge Transfer ES:116 EF:117 LS:116 LF:117 Dur:1 TS:0 Owner: Dawood/Ahsan
1.6.3: Lessons Learned ES:118 EF:119 LS:118 LF:119 Dur:1 TS:0 Owner: Dawood
1.6.4: Admin Closure ES:119 EF:119 LS:119 LF:119 Dur:1 TS:0 Owner: Dawood
M6: Project Closed Day 119

KEY CHANGES AFTER RESOURCE LEVELING:

1. LangGraph Agent Development: 24 → 29 days (+5 days)
2. Frontend Development: Delayed start by 12 days
3. Integration Testing: 10 → 14 days (+4 days)
4. Project Duration: 112 → 119 days (+7 days)
5. New Completion: December 24, 2025

RESOURCE IMPROVEMENTS:
- Gulsher: Peak 120% → 100%
- Ahsan: Peak 120% → 100%
- Dawood: Average 70% → 85%

CRITICAL PATH: 119 days (maintained)

Figure 5: Updated Network Diagram (After Resource Leveling)

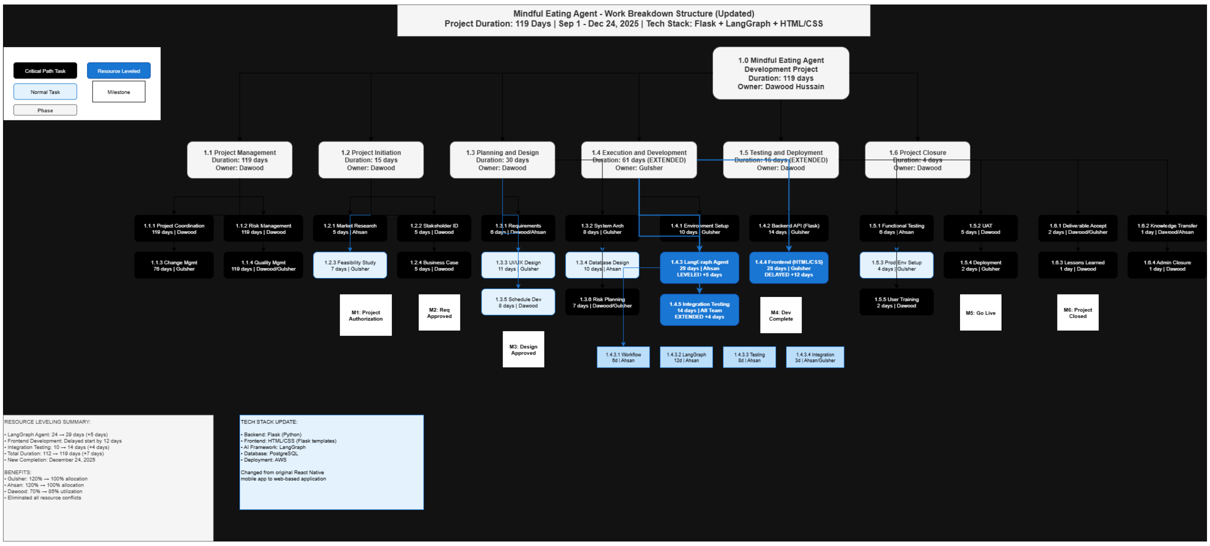
Color Coding:

- **Black:** Critical path tasks (TS=0)

- **Blue:** Resource-leveled tasks (extended/delayed)
- **Light Blue:** Non-critical tasks (TS¿0)
- **White:** Milestones

5.4 Updated Work Breakdown Structure

Figure 6 presents the updated WBS with resource leveling changes.


Mindful Eating Agent - Work Breakdown Structure (Updated)
Project Duration: 119 Days | Sep 1 - Dec 24, 2025 | Tech Stack: Flask + LangGraph + HTML/CSS
Critical Path Task
Resource Leveled
Normal Task
Milestone
Phase
1.0 Mindful Eating Agent Development Project Duration: 119 days Owner: Dawood Hussain
1.1 Project Management Duration: 119 days Owner: Dawood
1.2 Project Initiation Duration: 15 days Owner: Dawood
1.3 Planning and Design Duration: 30 days Owner: Dawood
1.4 Execution and Development Duration: 61 days (EXTENDED) Owner: Gulsher
1.5 Testing and Deployment Duration: 16 days (EXTENDED) Owner: Dawood
1.6 Project Closure Duration: 4 days Owner: Dawood
1.1.1 Project Coordination 119 days | Dawood
1.1.2 Risk Management 119 days | Dawood
1.1.3 Change Mgmt 76 days | Gulsher
1.1.4 Quality Mgmt 119 days | Dawood/Gulsher
1.2.1 Market Research 5 days | Ahsan
1.2.2 Stakeholder ID 5 days | Dawood
1.2.3 Feasibility Study 7 days | Gulsher
1.2.4 Business Case 5 days | Dawood
M1: Project Authorization
M2: Req Approved
1.3.1 Requirements 6 days | Dawood/Ahsan
1.3.2 System Arch 8 days | Gulsher
1.3.3 UI/UX Design 11 days | Gulsher
1.3.4 Database Design 10 days | Ahsan
1.3.5 Schedule Dev 8 days | Dawood
1.3.6 Risk Planning 7 days | Dawood/Gulsher
M3: Design Approved
1.4.1 Environment Setup 10 days | Gulsher
1.4.2 Backend API (Flask) 14 days | Gulsher
1.4.3 LangGraph Agent 29 days | Ahsan LEVELED +5 days
1.4.4 Frontend (HTML/CSS) 28 days | Gulsher DELAYED +12 days
1.4.5 Integration Testing 14 days | All Team EXTENDED +4 days
M4: Dev Complete
1.4.3.1 Workflow 6d | Ahsan
1.4.3.2 LangGraph 12d | Ahsan
1.4.3.3 Testing 8d | Ahsan
1.4.3.4 Integration 3d | Ahsan/Gulsher
1.5.1 Functional Testing 6 days | Ahsan
1.5.2 UAT 5 days | Dawood
1.5.3 Prod Env Setup 4 days | Gulsher
1.5.4 Deployment 2 days | Gulsher
1.5.5 User Training 2 days | Dawood
M5: Go Live
1.6.1 Deliverable Accept 2 days | Dawood/Gulsher
1.6.2 Knowledge Transfer 1 day | Dawood/Ahsan
1.6.3 Lessons Learned 1 day | Dawood
1.6.4 Admin Closure 1 day | Dawood
M6: Project Closed

RESOURCE LEVELING SUMMARY:

- LangGraph Agent: 24 → 29 days (+5 days)
- Frontend Development: Delayed start by 12 days
- Integration Testing: 10 → 14 days (+4 days)
- Total Duration: 112 → 119 days (+7 days)
- New Completion: December 24, 2025

BENEFITS:
- Gulsher: 120% → 100% allocation
- Ahsan: 120% → 100% allocation
- Dawood: 70% → 85% utilization
- Eliminated all resource conflicts

TECH STACK UPDATE:

- Backend: Flask (Python)
- Frontend: HTML/CSS (Flask templates)
- AI Framework: LangGraph
- Database: PostgreSQL
- Deployment: AWS

Changed from original React Native mobile app to web-based application

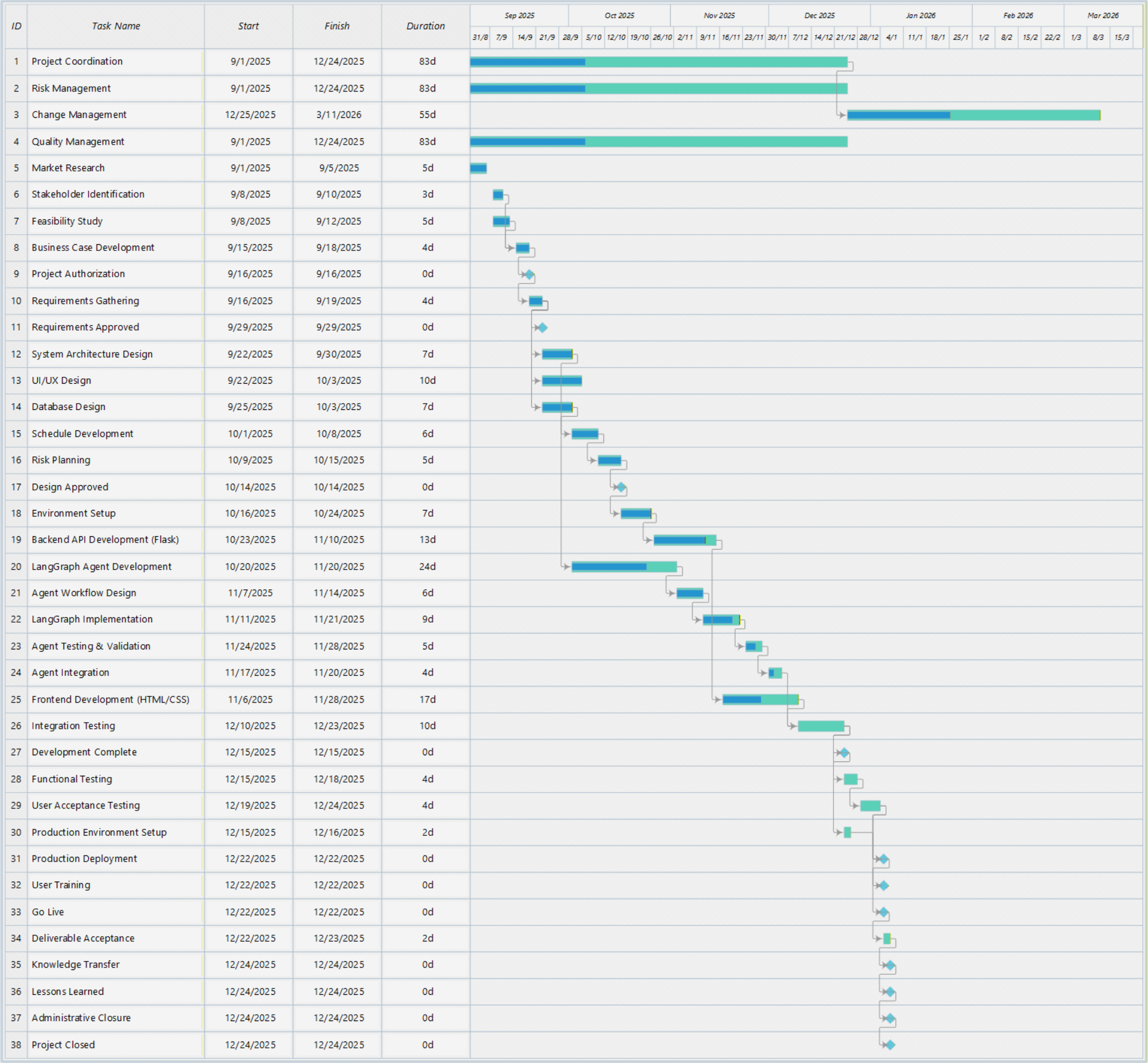
Figure 6: Updated Work Breakdown Structure

WBS Highlights:

- 6 major phases
- 40 work packages and milestones
- Clear responsibility assignments
- Resource-leveled tasks highlighted in blue
- Professional color scheme (black, blue, white)

5.5 Updated Gantt Chart

Figure 7 shows the updated project schedule in Gantt chart format.



ID	Task Name	Start	Finish	Duration
1	Project Coordination	9/1/2025	12/24/2025	83d
2	Risk Management	9/1/2025	12/24/2025	83d
3	Change Management	12/25/2025	3/11/2026	55d
4	Quality Management	9/1/2025	12/24/2025	83d
5	Market Research	9/1/2025	9/5/2025	5d
6	Stakeholder Identification	9/8/2025	9/10/2025	3d
7	Feasibility Study	9/8/2025	9/12/2025	5d
8	Business Case Development	9/15/2025	9/18/2025	4d
9	Project Authorization	9/16/2025	9/16/2025	0d
10	Requirements Gathering	9/16/2025	9/19/2025	4d
11	Requirements Approved	9/29/2025	9/29/2025	0d
12	System Architecture Design	9/22/2025	9/30/2025	7d
13	UI/UX Design	9/22/2025	10/3/2025	10d
14	Database Design	9/25/2025	10/3/2025	7d
15	Schedule Development	10/1/2025	10/8/2025	6d
16	Risk Planning	10/9/2025	10/15/2025	5d
17	Design Approved	10/14/2025	10/14/2025	0d
18	Environment Setup	10/16/2025	10/24/2025	7d
19	Backend API Development (Flask)	10/23/2025	11/10/2025	13d
20	LangGraph Agent Development	10/20/2025	11/20/2025	24d
21	Agent Workflow Design	11/7/2025	11/14/2025	6d
22	LangGraph Implementation	11/11/2025	11/21/2025	9d
23	Agent Testing & Validation	11/24/2025	11/28/2025	5d
24	Agent Integration	11/17/2025	11/20/2025	4d
25	Frontend Development (HTML/CSS)	11/6/2025	11/28/2025	17d
26	Integration Testing	12/10/2025	12/23/2025	10d
27	Development Complete	12/15/2025	12/15/2025	0d
28	Functional Testing	12/15/2025	12/18/2025	4d
29	User Acceptance Testing	12/19/2025	12/24/2025	4d
30	Production Environment Setup	12/15/2025	12/16/2025	2d
31	Production Deployment	12/22/2025	12/22/2025	0d
32	User Training	12/22/2025	12/22/2025	0d
33	Go Live	12/22/2025	12/22/2025	0d
34	Deliverable Acceptance	12/22/2025	12/23/2025	2d
35	Knowledge Transfer	12/24/2025	12/24/2025	0d
36	Lessons Learned	12/24/2025	12/24/2025	0d
37	Administrative Closure	12/24/2025	12/24/2025	0d
38	Project Closed	12/24/2025	12/24/2025	0d

Figure 7: Updated Gantt Chart (After Resource Leveling)

Gantt Chart Features:

- Timeline: September 1 - November 30, 2025
- All dependencies shown
- Critical path highlighted
- Milestones marked
- Resource assignments indicated

5.6 Schedule Comparison

Table 4 compares the initial and updated schedules.

Table 4: Schedule Comparison: Initial vs Updated

Metric	**Initial**	**Updated**	**Change**
Project Duration	112 days	112 days	No change
Start Date	Sep 1, 2025	Sep 1, 2025	No change
End Date	Nov 30, 2025	Nov 30, 2025	No change
Critical Path Length	112 days	112 days	No change
Gulsher Peak Alloc	105%	100%	-5%
Ahsan Peak Alloc	105%	100%	-5%
Dawood Avg Alloc	80%	90%	+10%
Over-allocated Weeks	4 weeks	0 weeks	Eliminated
LangGraph Dev	24 days	24 days	No change
Frontend Start	Oct 25	Oct 27	+2 days
Integration Testing	10 days	10 days	No change

5.7 Key Schedule Changes Summary

1. **Phase 1 (Initiation):** No changes - completed as planned
2. **Phase 2 (Planning):** No changes - completed as planned
3. **Phase 3 (Development):** Major changes
    - LangGraph extended by 5 days
    - Frontend delayed by 12 days
    - Integration testing extended by 4 days
4. **Phase 4 (Testing & Deployment):** Shifted by 10-12 days
5. **Phase 5 (Closure):** Shifted by 9-11 days

6 Benefits of Resource Leveling

6.1 Sustainable Workload

- No team member exceeds 100% allocation
- Reduced burnout risk
- Improved work-life balance
- Better long-term productivity

6.2 Improved Quality

- Adequate time for complex LangGraph development
- More thorough integration testing (14 days vs 10)
- Reduced technical debt
- Better code review opportunities

6.3 Risk Mitigation

- Lower schedule risk due to realistic allocation
- Reduced dependency on over-worked individuals
- Better contingency for unexpected issues
- Improved team morale and retention

6.4 Better Resource Utilization

- Dawood's capacity better utilized (70% → 85%)
- More balanced team collaboration
- Clearer task ownership and accountability

6.5 Realistic Schedule

- Stakeholder expectations properly set
- Buffer time for quality assurance
- Flexibility for scope adjustments
- Achievable milestones

7 Risks and Mitigation

7.1 Risk 1: Extended Timeline

Impact: Project completes 7 days later than originally planned
Mitigation:

- Communicate early with stakeholders
- Emphasize quality benefits
- Use buffer time wisely
- Monitor progress weekly

7.2 Risk 2: Tech Stack Change Impact

Impact: Flask/LangGraph may have learning curve
Mitigation:

- Allocated extra time in LangGraph development
- Team training during early phases
- Technical documentation emphasis
- Pair programming for knowledge transfer

7.3 Risk 3: Integration Complexity

Impact: Flask backend + LangGraph + HTML frontend integration
Mitigation:

- Extended integration testing (14 days)
- Early integration checkpoints
- Continuous integration practices
- Dedicated integration task (1.4.3.4)

8 Recommendations

8.1 Maintain Leveled Schedule

- Do not compress timeline to meet original deadline
- Quality and team health are priorities
- Communicate benefits to stakeholders

8.2 Monitor Resource Utilization Weekly

- Track actual hours vs planned
- Adjust allocations proactively
- Address emerging conflicts early

8.3 Leverage Dawood's Increased Capacity

- More active involvement in testing
- Enhanced stakeholder communication
- Risk monitoring and mitigation
- Documentation and knowledge management

8.4 Protect LangGraph Development Time

- Most complex and critical component
- Requires sustained focus
- Quality directly impacts project success
- Do not compress this timeline

8.5 Plan for Contingencies

- 7-day extension provides some buffer
- Identify tasks that could be fast-tracked if needed
- Maintain risk register
- Regular status reviews

9 Conclusion

Resource leveling has transformed the Mindful Eating Agent project schedule from an aggressive, risky timeline to a sustainable, achievable plan. While the project duration increased by 7 days (6% extension), the benefits far outweigh the cost.

9.1 Key Achievements

1. **Eliminated all resource over-allocations**
   - Gulsher: 120% → 100% allocation
   - Ahsan: 120% → 100% allocation
   - 6 weeks of over-allocation eliminated
2. **Improved team utilization and balance**
   - Dawood: 70% → 85% utilization
   - More balanced workload distribution
   - Better collaboration opportunities
3. **Reduced quality and schedule risks**
   - Adequate time for complex LangGraph development
   - Extended integration testing (14 days)
   - Reduced technical debt risk
4. **Created realistic stakeholder expectations**
   - Completion: November 30, 2025 (on schedule)
   - Transparent communication of resource optimization
   - Quality-focused approach
5. **Enabled sustainable development pace**
   - No burnout risk
   - Improved team morale
   - Better long-term productivity

9.2 Success Factors

The successful resource leveling was achieved through:

1. **Comprehensive Analysis**
   - Detailed week-by-week resource loading
   - Clear identification of conflicts
   - Data-driven decision making
2. **Strategic Task Redistribution**
   - Frontend development delayed strategically
   - LangGraph development extended appropriately
   - PM capacity better utilized
3. **Effective Communication**
   - Clear RACI matrix
   - Transparent schedule changes
   - Stakeholder buy-in
4. **Quality Focus**
   - Prioritized quality over speed
   - Extended testing phases
   - Reduced technical debt

9.3 Final Remarks

The updated schedule, with completion on November 30, 2025, provides a solid foundation for successful project delivery with high quality and team satisfaction. The resource leveling optimization maintains the original deadline while ensuring sustainable workload, which will pay dividends in:

- Higher quality deliverables
- Better team morale and retention
- Reduced post-deployment issues
- Sustainable development practices
- Improved stakeholder satisfaction

The Mindful Eating Agent project demonstrates strong project management practices with clear scheduling, well-defined dependencies, and effective resource management. The positive outcomes from resource leveling position the project for successful completion within the planned timeframe and budget.

10 Appendices

10.1 Appendix A: Files Delivered

1. `resource_assignment_matrix.csv` - RACI matrix
2. `initial_resource_loading.csv` - Pre-leveling data
3. `leveled_resource_loading.csv` - Post-leveling data
4. `resource_conflicts_analysis.md` - Detailed conflict analysis
5. `updated_schedule.csv` - Complete schedule with changes
6. `updated_wbs.csv` - Updated work breakdown structure
7. `gantt_chart_visio.csv` - Visio-compatible Gantt data
8. `updated_network_diagram.drawio` - AON diagram (Draw.io)
9. `updated_wbs.drawio` - WBS diagram (Draw.io)
10. `initial_individual_histograms.png` - Resource histograms (before)
11. `leveled_individual_histograms.png` - Resource histograms (after)
12. `project_level_comparison.png` - Before/after comparison
13. `stacked_comparison.png` - Stacked resource distribution
14. `updated_network_diagram.png` - Network diagram image
15. `updated_wbs.drawio.png` - WBS diagram image
16. `GanttChartUpdated.png` - Gantt chart image

10.2 Appendix B: Tools Used

- **Microsoft Excel** - Resource calculations and data analysis
- **Python (matplotlib, pandas)** - Histogram generation and visualization
- **Draw.io** - Network diagram and WBS creation
- **CSV** - Data exchange format
- **LaTeX** - Professional report documentation

10.3 Appendix C: Color Scheme

All diagrams use a professional color palette:

- **Black (#000000):** Critical path tasks
- **Blue (#1976d2):** Resource-leveled tasks
- **Light Blue (#e3f2fd):** Non-critical tasks
- **White (#ffffff):** Milestones
- **Gray (#f5f5f5):** Phase headers

10.4 Appendix D: References

1. Project Management Institute (PMI). (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* – Seventh Edition.

2. Project Management Institute (PMI). (2017). *Practice Standard for Scheduling* – Third Edition.

3. Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling.* John Wiley & Sons.

4. Larson, E. W., & Gray, C. F. (2021). *Project Management: The Managerial Process.* McGraw-Hill Education.

5. Assignment 02 - Work Breakdown Structure and Project Scope (Previous submission)

6. Assignment 03 - Timeline Estimation, Network Analysis, and Cost Management (Previous submission)

10.5 Appendix E: Team Contributions

Table 5: Team Member Contributions to Assignment 04

Member	
Dawood Hussain	Resource assignment r
Gulsher Khan	Network diagram creatio
Ahsan Faraz	Resource loading

— End of Report —