# Assignment 3 – Project Work Plan (Group 3: MemoriAI)

## 🎯 1. Objective

This project phase requires the development of three MVPs (Minimum Viable Products) based on the previously defined use cases. Only one use case will be fully implemented, while the other two will be designed at the architectural and planning level.

Group: Group 3 – MemoriAI

Use Cases:

🧠 Cognitive & Identity Assist – memory recall, word-finding, identity recognition (Fully Implemented MVP)

⏰ Daily Reminders & Safety Support – adaptive, intelligent reminders (Architectural & API Design only)

📊 Caregiver Dashboard & Awareness Hub – caregiver insights and analytics (Architectural & API Design only)

## ✅ 2. Deliverables

• Updated HLD, Architecture Overview, and README reflecting all 3 MVPs.

• Updated Azure DevOps Board with backlog grooming, Internal Review & UAT tasks, and PR links.

• Detailed API contract tables for each MVP with endpoints, parameters, and latency targets.

• Fully implemented Cognitive MVP (FastAPI + Docker + SQLite/VectorDB) following INFO8665\_API\_Architectures\_v2.

• Architectural diagrams and API schemas for Reminder and Dashboard MVPs (not implemented).

• A complete README, CI/CD workflow documentation, and an updated Scrum Plan.

• 5-minute presentation covering architecture, live demo (Cognitive MVP), and planned extensions.

## 🧩 3. MVP Overview and Scope

|  |  |  |  |
| --- | --- | --- | --- |
| MVP | Scope | Key Deliverables | Responsible |
| MVP 1 – Cognitive & Identity Assist | Fully Implemented | FastAPI microservice, Cognitive API, testing & deployment | Tessa |
| MVP 2 – Daily Reminders & Safety | Architectural & API Design Only | Service contracts, endpoint mapping, backlog structure | Krishna |
| MVP 3 – Caregiver Dashboard Hub | Architectural & API Design Only | Dashboard wireframes, endpoint schema, README visuals | Bhupender |

## 👥 4. Team Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Member | Role | Responsibilities |
| Tessa (AI Lead) | AI & Documentation Lead | Develop and test Cognitive MVP (Use Case 1), finalize documentation, architecture diagrams, and deliver presentation. |
| Krishna (Backend & QA) | Backend Developer & QA Lead | Design Reminder Service architecture (Use Case 2), perform API testing and validation across all MVPs, review PRs. |
| Adhitya (DevOps) | DevOps Engineer | Handle Dockerization, CI/CD pipelines, Azure deployment, and monitoring integration. |
| Bhupender (UI/UX) | Frontend Developer | Design Dashboard (Use Case 3) architecture and wireframes, link API routes, and prepare README visuals. |
| Nikitha (QA & Docs) | QA & Documentation | Maintain Azure Board, map PRs to backlog, assist with UAT documentation and presentation content. |

## 🧠 5. Technical Checklist

• Architecture Diagram showing all services (Cognitive, Reminder, Dashboard) and ports (8001–8003).

• SOA + MLOps compliance validated across architecture documents.

• FastAPI Cognitive microservice implemented and tested (main deliverable).

• Reminder and Dashboard services documented with full API design tables (no implementation).

• Docker Compose and CI/CD setup configured (Azure environment).

• Azure DevOps backlog updated with Internal Review & UAT tasks for each feature.

• Sprint 1 branch active in GitHub; code, documentation, and PR links pushed.

• 5-minute team presentation including demo, architecture overview, and team Q&A readiness.

## 📅 6. Project Timeline and Strategy

Sprint 0: Environment setup, repository linking, backlog initialization.

Sprint 1: Cognitive MVP implementation and testing (Tessa).

Sprint 2: Reminder and Dashboard API designs, documentation (Krishna & Bhupender).

Sprint 3: Integration, Docker, CI/CD testing (Adhitya).

Sprint 4: Review, presentation prep, and submission (Full Team).

## 🏆 7. Rubric Alignment Plan

• Backlog & Sprint Planning: Defined sprints, balanced workload, and PR mapping.

• Pull Requests: Internal Review & UAT tasks linked for each feature.

• Architecture Implementation: SOA architecture fully mapped with one working MVP.

• Repository Structure: Organized repo with all branches, code, and docs.

• Presentation & Collaboration: All members participate and present respective sections.

• Evaluation Readiness: Documentation and evidence prepared for all rubric categories.