Assignment Submission Form

1. Group Code/Name

eldad_ron_bar_yacobi

---

2. Team Member 1

- **ID (Required):** 207021916
- **Name (Optional):** Eldad Ron

---

3. Team Member 2

- **ID (Required):** 315471367
- **Name (Optional):** Bar Yacobi

---

4. GitHub Repository Link

Repository URL:
https://github.com/er1009/LLMs-And-Multi-Agent-Orchestration-Course/tree/main/ex1

Note: Please ensure the repository is set to **Public** for grading purposes.

---

5. Self-Recommended Grade

Recommended Grade: 100

Justification:

We recommend a grade of 100 based on the following:

Technical Implementation (30 points):

- Complete desktop chat application with GUI (Tkinter)
- Proper API integration with Ollama REST API (not web interface)

- Clean, modular code structure following best practices
- Conversation history maintenance working correctly
- Robust error handling and user feedback
- Model selection functionality implemented

Documentation (25 points):

- Comprehensive README.md with detailed installation instructions
- Screenshots of running application included
- Complete project structure documented
- All requirements clearly addressed
- Professional documentation in Documentation folder (PRD_PROMPT.md, AI_PROMPTS.md, DEVELOPMENT.md)

Testing (20 points):

- Unit tests for API client (5 tests)
- Integration tests for end-to-end flow (3 tests)
- All tests include expected results in docstrings
- Tests handle edge cases (Ollama not running, no models, etc.)
- Total of 8 tests covering all major functionality
- All tests pass successfully

Code Quality (15 points):

- Single responsibility principle followed
- Clean, readable code with proper documentation
- Proper error handling throughout
- No unnecessary complexity
- Well-structured project organization

Requirements Compliance:

- Uses virtual environment (conda) and requirements.txt
- Connects via API (not web interface)
- All assignment requirements met
- Professional documentation structure
- Complete PRD document included
- All screenshots provided
- All unit tests with expected results documented

Additional Strengths:

- Complete implementation of all required features
- Professional code quality and structure
- Comprehensive documentation covering all aspects
- Proper use of API endpoints (not web interface)
- Clean, maintainable codebase
- All deliverables completed to specification

We believe this submission demonstrates complete understanding of the requirements, excellent implementation quality, and professional documentation standards that fully meet all assignment criteria.

Exercise 1: LLMs and Multi-Agent Orchestration

Grade Report

Student ID:	38962
Team:	eldad_ron_bar_yacobi
Repository:	https://github.com/er1009/LLMs-And-Multi-Agent-Orchestration-Course/tree/mai
Assessment Date:	December 01, 2025

FINAL GRADE	**47%**
STATUS	**NEEDS IMPROVEMENT**

Assessment Feedback

This submission requires significant improvement. You need to focus on fundamental software engineering practices including planning, documentation, testing, and version control.

Strengths

• Perfect Python docstring coverage (9/9 files, 100%)

• Comprehensive README documentation

• Strong version control (prompt documentation present)

• Good code modularity (64 lines/file average)

• Clean, type-annotated code

Required Improvements

• CRITICAL: Add research analysis (parameter investigation, notebooks, or markdown documentation)

• CRITICAL: Add .gitignore file for security best practices

• Add comprehensive testing with mocks and coverage configuration

• Create complete PRD with all required sections

• Add architecture documentation

• Implement cost tracking and analysis

• Add quality tools (linting, CI/CD, style guides)

Please review the requirements and resubmit. Please review course materials and requirements carefully before your next submission.

Assessed: December 01, 2025

This grade reflects your overall software engineering practices

Exercise 1: LLMs and Multi-Agent Orchestration

Grade Report

Student ID:	38962
Team:	eldad_ron_bar_yacobi
Repository:	https://github.com/er1009/LLMs-And-Multi-Agent-Orchestration-Course/tree/mai
Assessment Date:	December 01, 2025

FINAL GRADE	**47%**
STATUS	**NEEDS IMPROVEMENT**

Assessment Feedback

This submission requires significant improvement. You need to focus on fundamental software engineering practices including planning, documentation, testing, and version control.

Strengths

• Perfect Python docstring coverage (9/9 files, 100%)

• Comprehensive README documentation

• Strong version control (prompt documentation present)

• Good code modularity (64 lines/file average)

• Clean, type-annotated code

Required Improvements

• CRITICAL: Add research analysis (parameter investigation, notebooks, or markdown documentation)

• CRITICAL: Add .gitignore file for security best practices

• Add comprehensive testing with mocks and coverage configuration

• Create complete PRD with all required sections

• Add architecture documentation

• Implement cost tracking and analysis

• Add quality tools (linting, CI/CD, style guides)

Please review the requirements and resubmit. Please review course materials and requirements carefully before your next submission.

Assessed: December 01, 2025

This grade reflects your overall software engineering practices