AI-Powered Task Assignment Agent for Smart Team Resource Allocation

---

1. Project Overview

Project Title:

AI-Powered Task Assignment Agent for Smart Team Resource Allocation

Contributors:

- Zainab Awada – 6146
- Zeina Balhas – 5998
- Hassan Hijazi – 5922

Supervisor:

Dr. Mohammed Aoudi

---

2. Introduction

In today's fast-paced work environments, task assignment is often handled manually by team leads or managers. This process can be time-consuming, subjective, and inefficient. Biases, unbalanced workloads, and skill mismatches are common pitfalls in traditional task allocation methods.

This project proposes the development of an AI-powered intelligent agent designed to automate the task assignment process using both **rule-based logic** and **machine learning (LLM)** techniques. The system leverages explainable AI (XAI) principles to maintain transparency and trust.

---

3. Problem Statement

Task allocation in organizations is typically:

- Manual and slow
- Prone to human bias
- Unaware of employee workload and skill compatibility

There is a need for a **data-driven**, **autonomous**, and **explainable** decision-making agent that can optimize this process.

---

4. Objectives

- Build a dual-mode task assignment system (Rule-based + LLM-based)
- Create a user-friendly frontend for task input and assignment review
- Develop a backend system integrating AI models and a relational database
- Implement explainable decision-making mechanisms
- Ensure local deployment for privacy and security

---

5. System Architecture

Frontend

- Technologies: HTML5, CSS3, JavaScript (Bootstrap or Vanilla)
- Features: Task input form, assignment display, mode selection

Backend

- Framework: Python (FastAPI or Flask)
- Responsibilities: API handling, rule/AI logic execution, database operations

Database

- System: MySQL
- Entities: Employees, Tasks, Workloads, Logs

AI/ML Components

- **Rule-Based Engine**: Matches tasks using skill tags and workload thresholds
- **LLM Engine (Mistral via Ollama)**: Generates intelligent and explainable task suggestions
- **Comparison Module**: Compares rule-based vs AI recommendations

Deployment

- **Docker Compose**: Enables full offline setup including frontend, backend, DB, and AI engine

---

6. Core Features

- 🧠 **AI-Powered Task Recommendation** using Ollama & Mistral
- ⚖️ **Workload-Balancing** through fair task distribution
- 🔒 **Privacy-First Architecture** with fully local deployment
- 💬 **Explainable Assignments** from both rule-based and AI-based engines
- 🌐 **Simple Web Interface** for task entry and viewing recommendations

---

7. How It Works

Step-by-Step Flow:

1. **Input Task:**
   Manager enters task title, required skills, and priority.

2. **Mode Selection:**
   Choose between Rule-based or AI-powered assignment.

3. **Processing:**
   Backend determines best-fit employee using selected logic.

4. **Output:**
   Result is displayed in the web interface with explanations and stored in the database.

---

8. Technology Justification

Layer	Technology	Reason
AI/ML Engine	Mistral (via Ollama)	Lightweight, performant, explainable
Backend	Python (Flask/FastAPI)	Rapid development, rich AI ecosystem
Frontend	HTML/CSS/JS	Lightweight, quick deployment
Database	MySQL	Robust, scalable, relational structure
Deployment	Docker Compose	Modular, offline-capable, reproducible

---

9. Demo Highlights

- Secure local execution (data stays on device)
- Full-stack integration with intuitive user interface

- Assignment recommendation with reasoning
- No internet required (Ollama runs locally)

---

10. Expected Outcomes

- Faster task allocation process
- More balanced workload among employees
- Reduced human error and bias
- Transparent and explainable decision-making
- Privacy-preserving AI deployment

---

11. Challenges & Mitigation

Challenge	Solution
Explainability of AI decisions	Use XAI Layer with generated justifications
Offline AI inference	Use lightweight LLM (Mistral via Ollama)
User trust in AI	Provide side-by-side comparison with rules
Data privacy	Run everything locally with Docker

---

12. Future Enhancements

- Support for dynamic skill learning via feedback
- Integration with calendars to assess availability
- Enhanced UI/UX with real-time assignment updates

- Voice input and NLP-based task entry

---

13. Conclusion

This project introduces a new standard in workforce automation by blending classic rules with modern AI for decision-making. By ensuring transparency, fairness, and local data security, this AI-powered agent not only enhances operational efficiency but also builds trust in autonomous systems.

---

14. References

- Ollama LLM Documentation
- FastAPI and Flask Documentation
- MySQL Deployment with Docker
- Explainable AI Concepts (XAI)