ALLIANCE UNIVERSITY

Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

PROJECT TITLE :- AI AGENT FOR TASK SCHEDULING AND DEPENDENCY RESOLUTION

TEAM MEMBERS:

1. DEEPAK V SAJJAN
2. SUNIL DK
3. RAVI SAINI

AI AGENT FOR TASK SCHEDULING AND DEPENDENCY RESOLUTION

1. Introduction

In the modern project management landscape, the complexity of task interdependencies often leads to scheduling bottlenecks, manual rescheduling overhead, and unforeseen delays. Traditional tools require significant manual input to maintain an accurate timeline, frequently failing to adapt dynamically when task parameters change.

The **AI Agent for Task Scheduling and Dependency Resolution** is designed to bridge the gap between natural language planning and rigorous algorithmic execution. By leveraging Large Language Models (LLMs) alongside advanced graph theory, this system transforms unstructured task descriptions into an optimized, conflict-free schedule.

2. Problem Statement

Project managers and developers often struggle with:

Complex Dependency Mapping: Manually identifying which tasks must precede others in a large-scale project.

Static Scheduling: Timelines that become obsolete the moment a single task is delayed.

Resource Constraints: Balancing strict deadlines with limited daily availability and team bandwidth.

3. Objectives

Natural Language Processing: Seamlessly convert informal task lists into structured scheduling data.

Autonomous Conflict Resolution: Automatically recalculate downstream impacts when a task's duration or status changes.

Visual Clarity: Provide users with a clear, interactive dashboard that visualizes both the dependency tree and the resulting timeline.

Through the integration of **Python's NetworkX** for graph operations and **GPT-4/Llama** for cognitive parsing, this agent acts as a dynamic project coordinator, reducing the cognitive load on human operators and ensuring project milestones are met with mathematical precision.

4. System Overview

This project is a smart project management agent that replaces manual planning with automated logic. It uses a three-step process to turn messy human instructions into a perfect, "dependency-aware" schedule.

How it Works

Extraction (LLM): The user types their plan in plain English. An AI (GPT-4) extracts the "what," the "how long," and the "after what" (dependencies).

Structuring (DAG): These tasks are mapped into a Directed Acyclic Graph (DAG). This mathematical structure ensures that "Task B" never happens before "Task A" and prevents circular logic (loops).

Optimization (Scheduler): The system uses Topological Sorting to order tasks and the Critical Path Method to find the most efficient timeline, highlighting which tasks can run in parallel to save time.

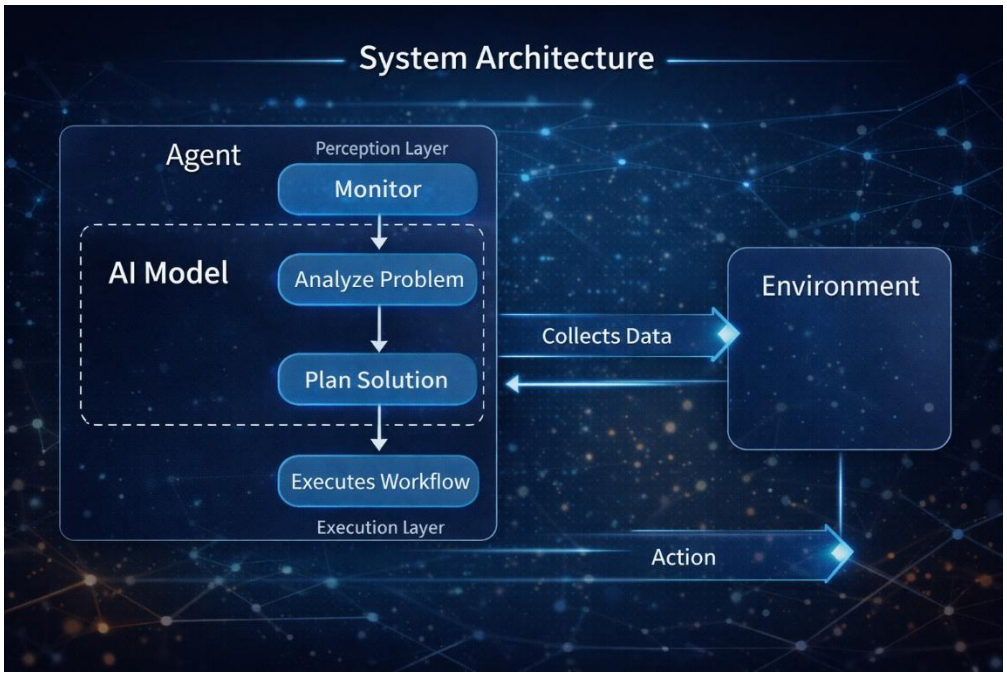
Why it's Different

Dynamic Rescheduling: If a user reports a delay on Day 1, the agent automatically "pushes" all dependent tasks and updates the entire timeline instantly.

Constraint-Aware: It respects human limits, such as "only 4 working hours per day" or "weekends off."

Visual Clarity: It provides both a big-picture phase view (e.g., "Development Phase") and a granular, hour-by-hour daily plan.

5. System Architecture



6. technologies used

Model

- GPT-4 / Llama 2 (LLM)

Backend

- Python
- FastAPI / Django
- NetworkX

Database

- PostgreSQL
- MongoDB

Frontend

- React
- Material UI (MUI)
- Tailwind CSS
- React Flow / D3.js

Integrations

- Google Calendar API
- Outlook API
- Slack Webhooks

7. Advantages

- **Real-time Adaptation:** Automatically shifts all dependent tasks when a delay occurs.

- **Optimal Parallelism:** Identifies which tasks can run at the same time to save days.

- **Natural Interface:** Users can plan using plain English instead of complex software menus.

- **Error Proofing:** Mathematical graph logic prevents impossible circular loops.

- **Critical Path Tracking:** Pinpoints exactly which tasks determine the final deadline.

8. Disadvantages

- **AI Interpretation Risks:** The LLM might misinterpret vague or complex dependencies.
- **Garbage In, Garbage Out:** Requires clear task descriptions to build an accurate graph.
- **Processing Cost:** Large projects with hundreds of tasks require more backend power to recalculate.
- **Lack of Nuance:** Algorithms don't account for human factors like team morale or "creative block."

9. Conclusion

This project successfully integrates **Large Language Models** with **Graph Theory** to bridge the gap between natural language planning and precise execution. By automating dependency resolution and real-time rescheduling, the agent eliminates manual overhead and ensures project timelines remain mathematically optimized even when delays occur. Ultimately, this system provides a scalable, "living" schedule that adapts to the complexities of modern project management, allowing teams to focus on execution rather than logistical maintenance.

10. References

https://github.com/ravi3404/Agentic-Ai-Project