Problem Statement: Agent of Justice

Overview:

Design an intelligent courtroom simulation system using Large Language Model (LLM) agents. Given a textual case description, your system must simulate a full courtroom trial involving multiple autonomous agents, each representing key legal roles.

Minimum Required Agents:

- **Defendant**
- **Defense Lawyer**
- **Plaintiff**
- **Prosecution Lawyer**
- **Judge**

Phases of the Simulation:

1. **Opening Statements**
    - Defense and Prosecution introduce their sides of the case.
2. **Witness Interrogation & Argumentation**
    - Defense and Prosecution can each interrogate witnesses or construct iterative arguments.
3. **Closing Statements**
    - Each side gives a final summary of their position.
4. **Judge's Ruling** *(Final Phase)*
    - The judge agent deliberates and delivers a verdict based on the arguments, evidence, and witness testimonies.

Bonus Objectives (Extra Credit):

- Support **freely evolving trial structure** —

  Even though you are provided with a basic conversation structure, you may implement flexible trial progression based on how the case unfolds, ensuring it is grounded in reality.

- Implement **dynamic agent creation** —

  You may extend this system with additional roles such as:

  - Witnesses (called during the trial)
  - Jury members
  - Expert consultants
  - Bailiff, stenographer, etc.

These roles should be dynamic — agents can be spawned or dismissed based on the needs of the case..

Submission Requirements:

- All submissions must be a **GitHub repo**.
- Repositories must include:
  - A **comprehensive README** explaining setup, usage, and the architecture.
  - Clear **instructions to run a demo** (CLI / Notebook / Web interface).
  - **Sample case inputs** from provided dataset and corresponding simulation outputs.

Judgment Criteria:
Creativity of solution
Agent Realism & Roleplay Depth
Dynamic Trial Flow & Flexibility
Judge's Reasoning & Verdict Coherence
Efficiency & Latency of implementation
Code Quality & Documentation