




PROJECT AND TEAM INFORMATION

Project Title:

Deadlock Simulator using Python

Student / Team Information:

Team Name	MEGARUSHERS
Team Member 1 (Team Lead) Samarth Agarwal Student ID: - 22011896 samarth2404agarwal@gmail.com	
Team Member 2 Kunwardeep Singh Student ID: - 22011787 kunwar2104@gmail.com	
Team Member 3 Lakshaydeep Chaudhary Student ID: - 2219016 lakshay71003@gmail.com	

PROJECT PROGRESS DESCRIPTION

Project Abstract:

The Deadlock Simulator is a Python-based tool designed to simulate, visualize, and analyse deadlock scenarios in operating systems. It aims to educate users about deadlock conditions (mutual exclusion, hold and wait, no pre-emption, circular wait) and demonstrate prevention/avoidance strategies like the Banker's Algorithm. The simulator includes interactive scenes (e.g., Exam Hall, Library) to illustrate real-world deadlock situations and their resolutions.

Updated Project Approach and Architecture:

The project follows a modular design:

1. **Core Simulation:** Implements processes, resources, and deadlock logic (threading, locks).
2. **Deadlock Detection:** Uses resource allocation graphs (e.g., Exam Hall scene).
3. **Prevention/Avoidance:** Priority scheduling in Library scenes (FIFO for teachers/students).
4. **Visualization:** Console-based output for simplicity (future: matplotlib/networkx).
5. **User Interface:** CLI for scene selection and input.

Technologies: Python, threading, sys, time, collections.

Tasks Completed:

Task Completed	Team Member
Exam Hall deadlock simulation	Samarth Agarwal
Library priority scheduling	Kunwardeep Singh
Optimized Library scene	Lakshaydeep Chaudhary

Challenges / Roadblocks:

1. **Thread Synchronization:** Deadlocks occurred during testing due to improper lock handling. *Solution:* Refined threading.Lock() usage.
2. **Input Validation:** Invalid inputs crashed scenes. *Solution:* Added try-except blocks.
3. **Visualization Limitations:** Current output is text-based. *Plan:* Integrate matplotlib in Phase 3.

Tasks Pending:

Task Pending	Team Member
Banker's Algorithm integration	Kunwardeep Singh
GUI development (tkinter/PyQt)	Lakshaydeep Chaudhary
Comprehensive testing	Samarth Agarwal

Project Outcomes / Deliverables:

1. Functional CLI simulator with 3 scenes.
2. Deadlock detection in Exam Hall scene.
3. Priority scheduling in Library scenes.
4. Documentation (code comments, user guide).

Project Overview:

- **Ahead of Schedule:** Core simulation and scenes (Exam Hall, Library).
- **On Track:** Input validation, thread synchronization.
- **Behind Schedule:** Visualization (pending matplotlib integration).

Codebase Information:

- **Repository:** GitHub (link to be added).
- **Branch:** main.
- **Key Commits:** Exam Hall logic, Library priority queues.

Testing and Validation Status:

Test Type	Status	Notes
Exam Hall deadlock	Pass	Validates mutual exclusion.
Library FIFO	Pass	Teacher/student priority works.

Deliverables Progress:

- **Completed:** Scenes 1–3, basic deadlock detection.
- **In Progress:** Banker's Algorithm, GUI.
- **Pending:** Final report, advanced visualization.