A Course Based Project Report on

**Resource Allocation Simulator**

Submitted to the

**Department of Computer Science and Engineering**

in partial fulfilment of the requirements for the completion of course

Operating Systems (22PC1IT202)

BACHELOR OF TECHNOLOGY

IN

**Computer Science and Engineering**

Submitted by

**V.NAGARJUNA 23071A05K8**

**V.SRI HASNIKA**  **23071A05K9**

**V.DHRUVAN 23071A05M1**

**Y.DEVENDRA**   **23071A05M2**

**Y.HARSHA VARDHAN**  **23071A05M3**

Under the guidance of

**Dr. D N VASUNDHARA**

**(Course Instructor)**

Assistant Professor, Department of CSE, VNRVJIET



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade, NBA

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as “College with Potential for Excellence” by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project report entitled “**Resource Allocation Simulator**” is a bonafide work done under our supervision and is being submitted by **V. NAGARJUNA (23071A05K8), V. SRI HASNIKA (23071A05K9), V. DHRUVAN (23071A05M1), Y. DEVENDRA (23071A05M2), Y. HARSHA VARDHAN (23071A05M3)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Computer Science and Engineering, of the VNRVJIET, Hyderabad during the academic year 2024-2025.

**Course Instructor Name DR. D N VASUNDHARA**

Assistant Professor

Department of CSE

**Course based Projects Reviewer**

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade,

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DECLARATION**



We declare that the course-based project work entitled “**File Compression and Decompression**” submitted in the Department of Computer Science and Engineering, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in** **Computer Science and Engineering** is a bonafide record of our own work carried out under the supervision of **DR. D N VASUNDHARA, Assistant Professor, Department of CSE, VNRVJIET.** Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

|  |  |  |  |
| --- | --- | --- | --- |
| **V.NAGARJUNA**  (23071A05K8) | **V.SRI HASNIKA**  **(**23071A05K9) | **V.DHRUVAN**  (23071A05M1) | **Y.DEVENDRA**  (23071A05M2) |

**Y.HARSHA VARDHAN**

(23071A05M3)

**ACKNOWLEDGEMENT**

We express our deep sense of gratitude to our beloved President, Sri. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr. C.D Naidu, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor Dr D N Vasundhara, Assistant Professor, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, **DR. D N VAUNDHARA**, Assistant Professor in Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

V. NAGARJUNA 23071A05K8

V. SRI HASNIKA 23071A05K9

V. DHRUVAN 23071A05M1

Y. DEVENDRA 23071A05M2

Y. HARSHAVARDHAN 23071A05M3

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.NO | TOPIC | PAGE |
| 1 | ABSTRACT | 6 |
| 2 | INTRODUCTION | 7 |
| 3 | METHODOLOGY | 8 |
| 4 | SOURCE CODE | 9-10 |
| 5 | RESULTS | 11-15 |
| 6 | OUTCOMES | 16 |
| 7 | APPLICATIONS | 17 |
| 8 | REFERENCES | 18 |

**ABSTRACT**

This project introduces a game-based learning prototype that simulates fundamental operating system concepts—primarily focusing on CPU scheduling algorithms and resource allocation mechanisms, including the Banker's Algorithm. The simulation is structured into an engaging and interactive game environment where players actively manage system resources and schedule processes, providing a hands-on learning experience. The system incorporates the most commonly used CPU scheduling algorithms: First-Come, First-Served (FCFS), Shortest Job First (SJF), Priority Scheduling, and Round Robin (RR). These are core components in operating systems and crucial for understanding how processes are managed and executed in real-world systems.

The game is divided into two modes—**Hack OS** and **System Commander**—each with a unique gameplay dynamic that challenges the player to optimize process execution and resource allocation to avoid system deadlocks. In **Hack OS**, the player is tasked with executing processes in a particular sequence using resources provided by the game. The challenge is to prevent deadlocks through strategic planning and correct application of the Banker's Algorithm. **System Commander**, on the other hand, puts the player in a role akin to a system administrator, requiring them to manage process priorities, allocate CPU time using scheduling algorithms, and ensure maximum throughput without compromising system stability.

This educational game prototype offers both theoretical and practical insights into process management, enabling users—particularly students and learners in the field of computer science and operating systems—to better understand complex scheduling and resource allocation concepts through interactive and experiential learning. The game also incorporates real-time scoring, where players earn points based on successful task execution, optimal scheduling, and avoidance of deadlock scenarios.

**INTRODUCTION**

Operating Systems (OS) serve as the backbone of computer systems, handling the coordination and management of hardware and software resources. Two of the most critical aspects in the study of operating systems are CPU scheduling and resource allocation. Traditional methods of teaching these concepts often involve textbook theory, abstract diagrams, and algorithmic analysis that can be challenging for students to visualize and internalize. To bridge this gap between theory and practical understanding, this project presents a game-based learning prototype that immerses the user in a simulated environment where they can experiment with real-time decision-making regarding CPU scheduling and resource allocation.

This prototype transforms theoretical knowledge into an interactive simulation, allowing users to input custom process parameters such as arrival time, burst time, and priority. These inputs are then used in conjunction with the selected CPU scheduling algorithm to determine the execution order of processes. The algorithms implemented—FCFS, SJF, Priority Scheduling, and Round Robin—are chosen for their relevance and widespread use in modern operating systems. Each algorithm provides unique challenges and outcomes, offering players opportunities to understand the strengths and trade-offs involved in each scheduling technique.

Furthermore, the inclusion of the Banker's Algorithm adds depth to the prototype, introducing resource allocation and deadlock avoidance—a complex topic often difficult to grasp in traditional teaching environments. The algorithm is presented in a game format where players must decide whether granting a process request will keep the system in a safe state or lead to potential deadlock. Through repeated gameplay and trial-and-error, users gain practical experience and intuition for recognizing safe and unsafe states in system resource management.

The game features two distinct gameplay modes:

* **Hack OS**, where players are given predefined resources and processes and must execute them in a deadlock-free sequence.
* **System Commander**, which allows players to design, schedule, and execute processes from scratch, using scheduling algorithms to manage CPU time and system resources.

This dual-mode gameplay encourages players to apply theoretical knowledge in practical scenarios, reinforcing their understanding through active problem-solving. The scoring system incentivizes efficient resource usage and successful task completion, making learning competitive, rewarding, and enjoyable. By integrating these core operating system principles into a gamified environment, the prototype serves as both an educational tool and an engaging simulation for students, educators, and enthusiasts alike.

**METHODOLOGY**

* **Requirement Gathering**

Defined the educational goals: teach CPU scheduling and resource allocation through interactive gameplay.

* **GameDesign**  
  Designed two modes—*Hack OS* and *System Commander*—to cover theoretical and practical aspects of OS concepts.
* **AlgorithmIntegration**  
  Implemented FCFS, SJF, Priority, and Round Robin scheduling, along with the Banker's Algorithm for deadlock avoidance.
* **Development**Built a user-friendly interface where players input process details and interact with scheduling mechanics and resource requests.
* **Testing&Refinement**  
  Collected user feedback through trial runs and improved gameplay, logic accuracy, and educational clarity.

**How It Works?**

1. **ProcessInput**  
   Players enter process data—arrival time, burst time, and priority.
2. **AlgorithmSelection**  
   Players choose a scheduling algorithm to see how processes are scheduled and executed.
3. **Game Modes:** In **Hack OS**, the player executes a fixed set of processes and resources to avoid deadlock, while in **System Commander**, the player creates and manages their own process set using scheduling algorithms.
4. **Bankers Algorithm Use**  
   Resource requests are evaluated using the Banker's Algorithm to determine if the system remains in a safe state.
5. **ScoringandFeedback**  
   Players earn points for correct scheduling, efficient execution, and deadlock avoidance, with real-time feedback and visual Gantt charts.

**SOURCE CODE**

**Simulation page**

import Link from "next/link"

import { motion } from "framer-motion"

export default function Home() {

return (

<main className="min-h-screen bg-os-darker flex flex-col">

<header className="bg-os-dark border-b border-os-light p-4">

<div className="container mx-auto flex justify-between items-center">

<h1 className="text-2xl font-bold text-white">

<span className="text-os-blue">OS</span> Resource Allocation Simulator

</h1>

<nav className="flex space-x-4">

<Link href="/simulation" className="text-white hover:text-os-blue transition-colors">

Simulation

</Link>

<Link href="/leaderboard" className="text-white hover:text-os-blue transition-colors">

Leaderboard

</Link>

<Link href="/login" className="text-white hover:text-os-blue transition-colors">

Login</Link></nav></div></header>

<section className="flex-1 flex flex-col items-center justify-center p-8 text-center">

<motion.div

initial={{ opacity: 0, y: 20 }}

animate={{ opacity: 1, y: 0 }}

transition={{ duration: 0.5 }}

className="max-w-3xl"

>

<h2 className="text-4xl md:text-5xl font-bold mb-4 text-white">

Learn <span className="text-os-blue">Operating Systems</span> Through Interactive Simulation

</h2>

<p className="text-xl text-gray-300 mb-8">

Master core OS concepts like Process Management, Deadlocks, and Synchronization through hands-on gameplay.</p>

<div className="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">

<div className="bg-os-dark p-6 rounded-lg border border-os-light">

<h3 className="text-xl font-bold text-os-green mb-2">Process Lifecycle</h3>

<p className="text-gray-400">

Visualize and control the complete process lifecycle from creation to termination.</ p>

</div>

<div className="bg-os-dark p-6 rounded-lg border border-os-light">

<h3 className="text-xl font-bold text-os-purple mb-2">CPU Scheduling</h3>

<p className="text-gray-400">

Experiment with different scheduling algorithms and observe their effects.

</p></div>

<div className="bg-os-dark p-6 rounded-lg border border-os-light">

<h3 className="text-xl font-bold text-os-red mb-2">Deadlock Management</h3>

<p className="text-gray-400">Create, detect, and resolve deadlocks in resource allocation scenarios.</p></div></div>

<Link

href="/simulation"

className="inline-block bg-os-blue hover:bg-blue-600 text-white font-bold py-3 px-8 rounded-md transition-colors text-lg"

>

Start Simulation</Link></motion.div></section>

<footer className="bg-os-dark border-t border-os-light p-4 text-center text-gray-400">

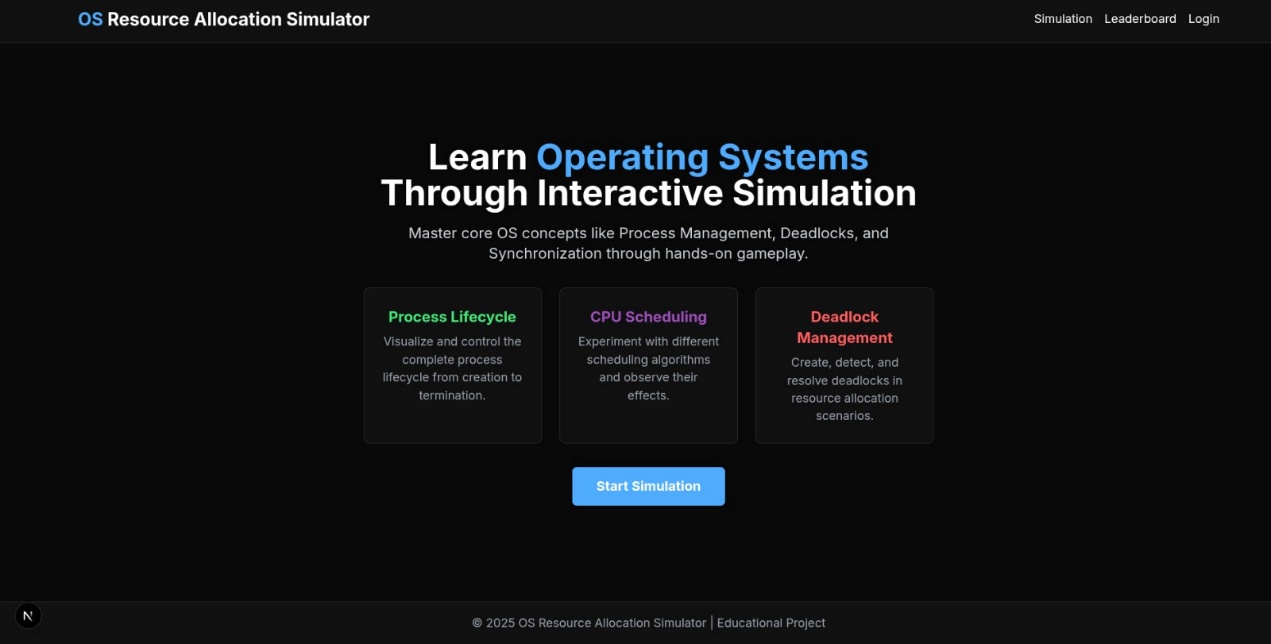
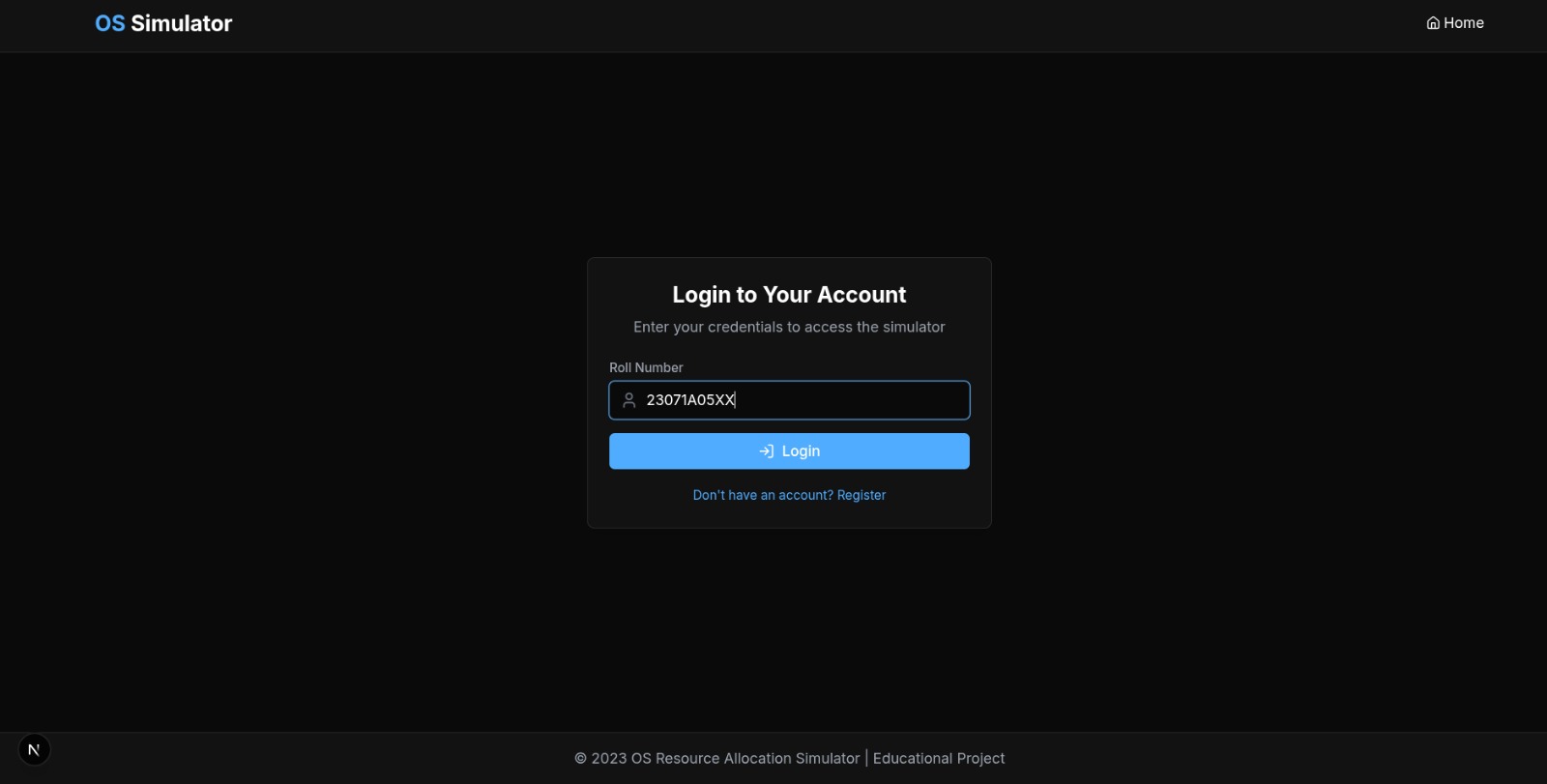
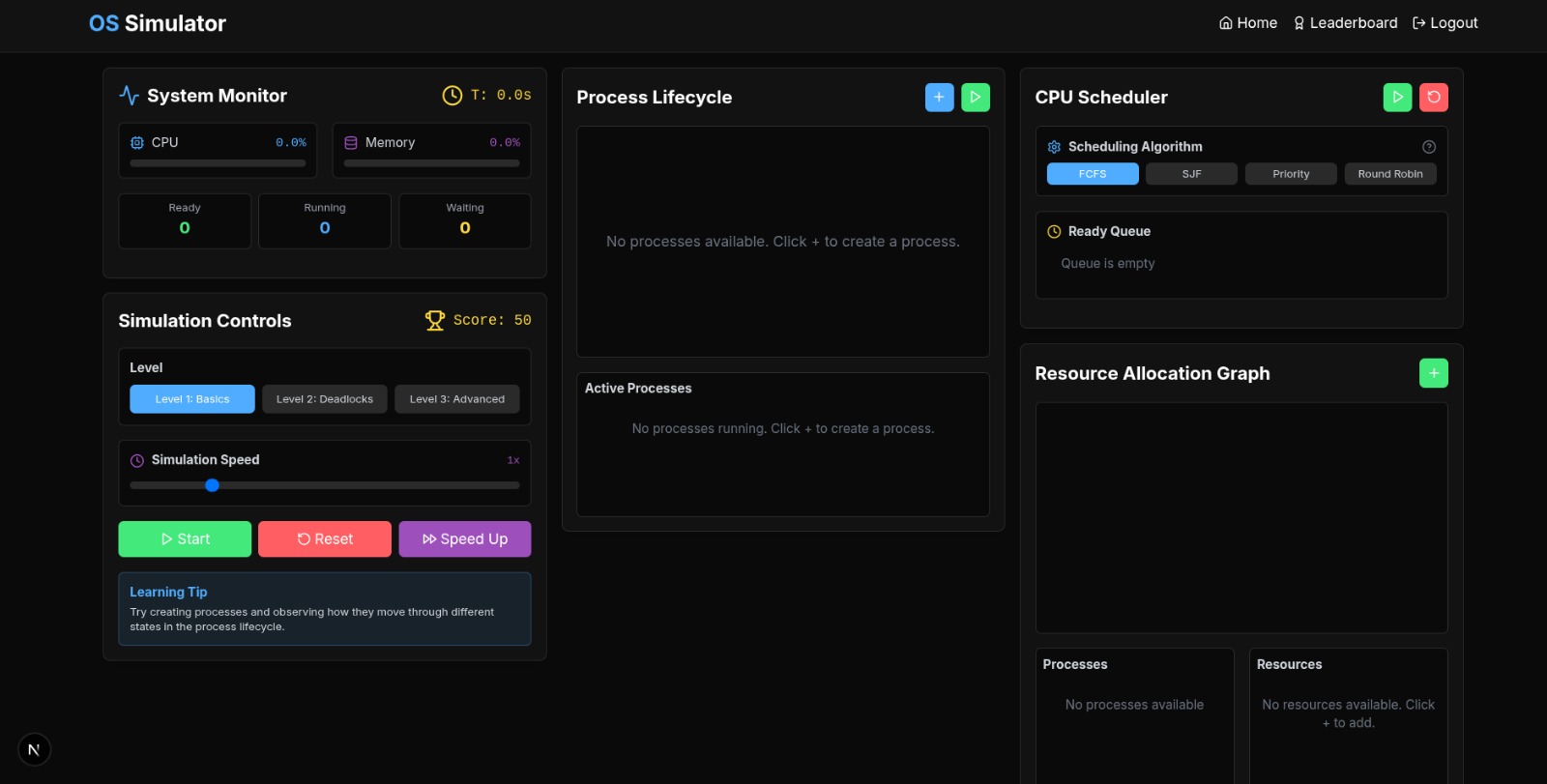
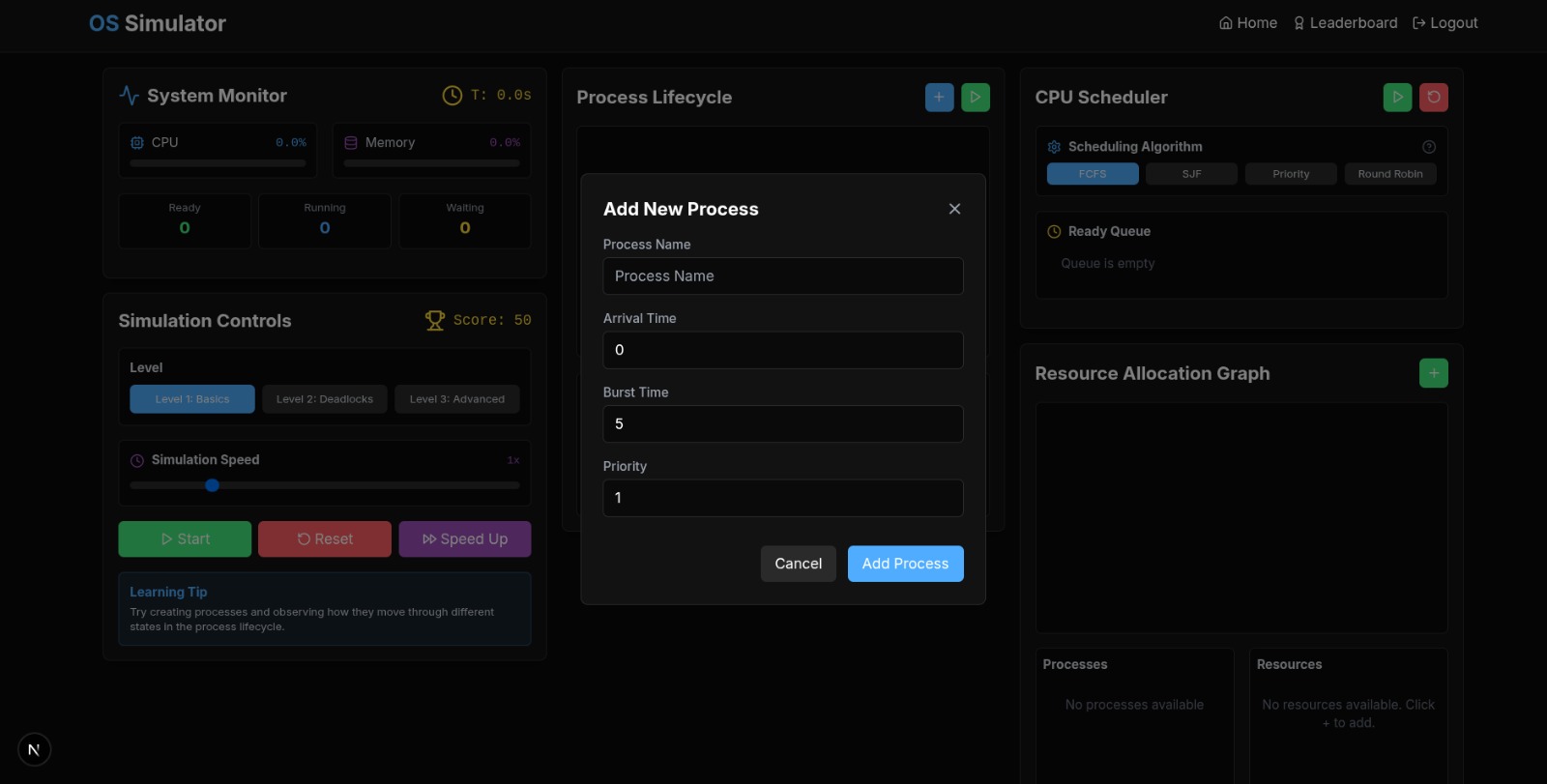
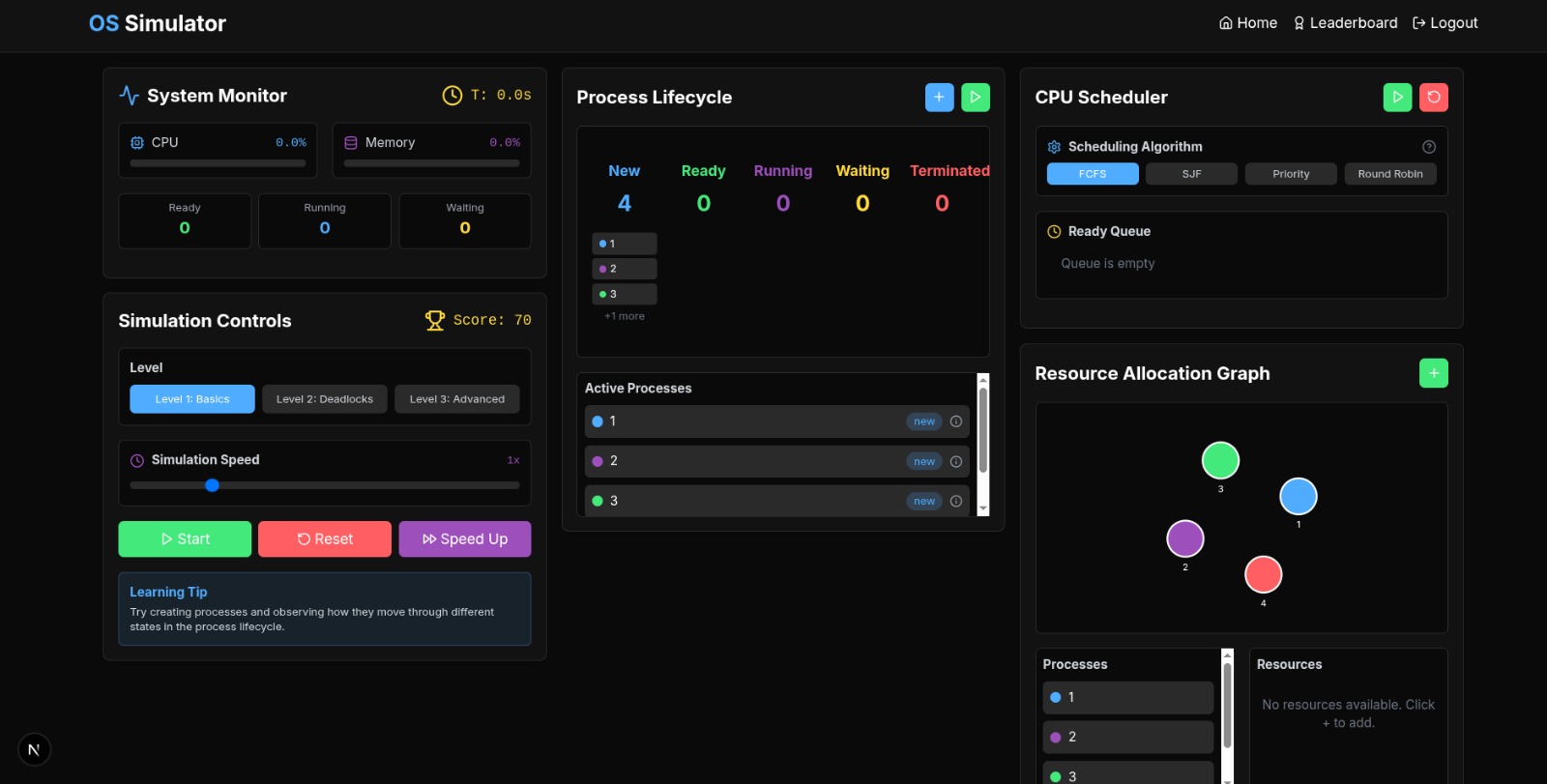
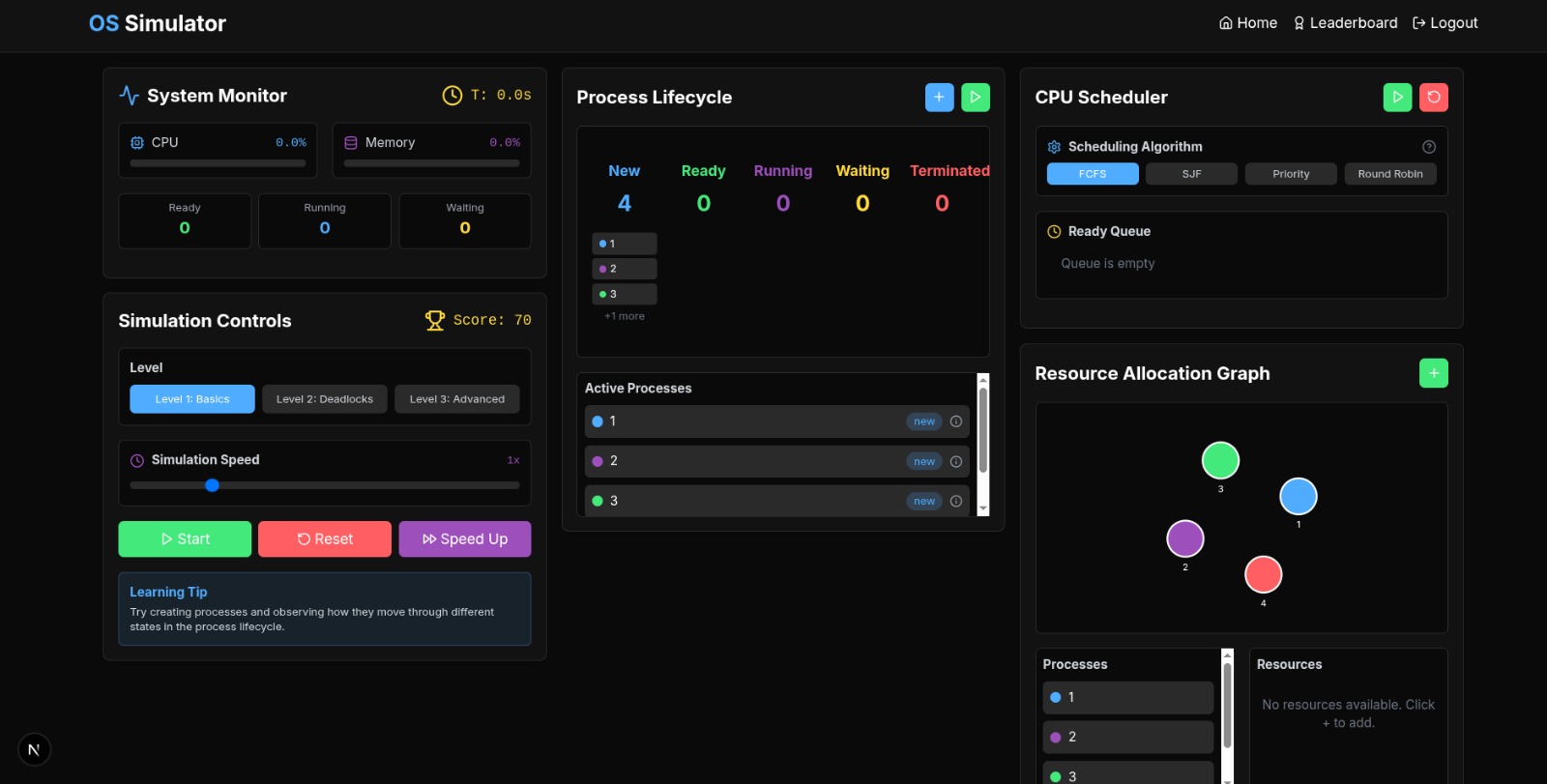
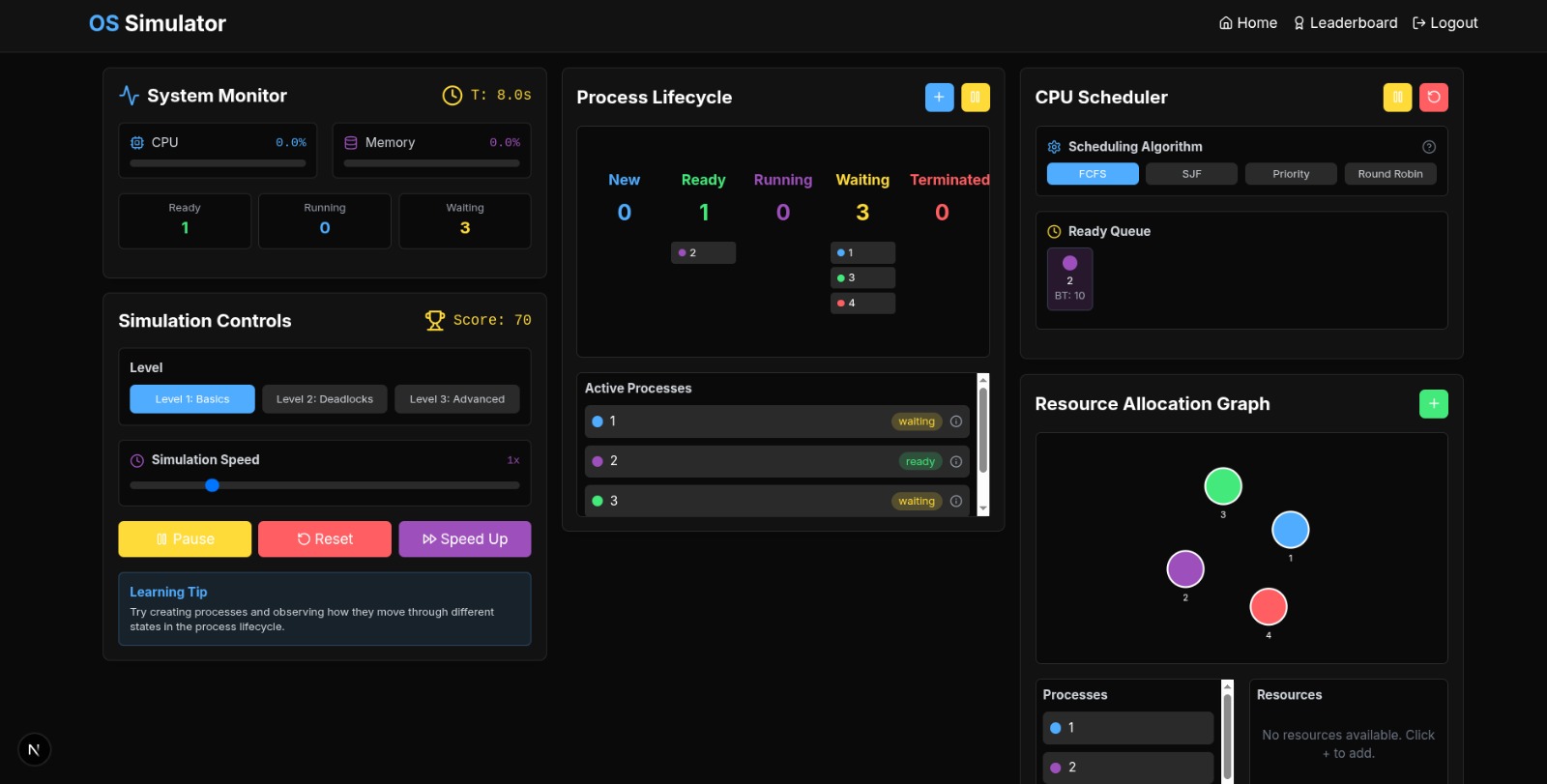
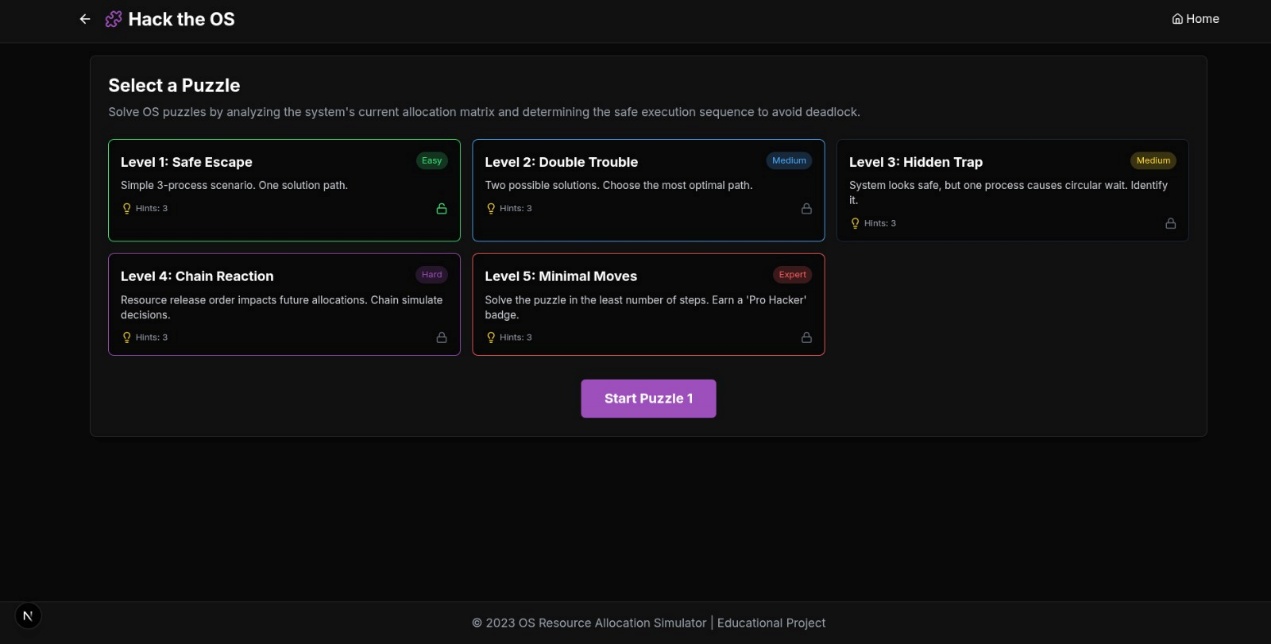
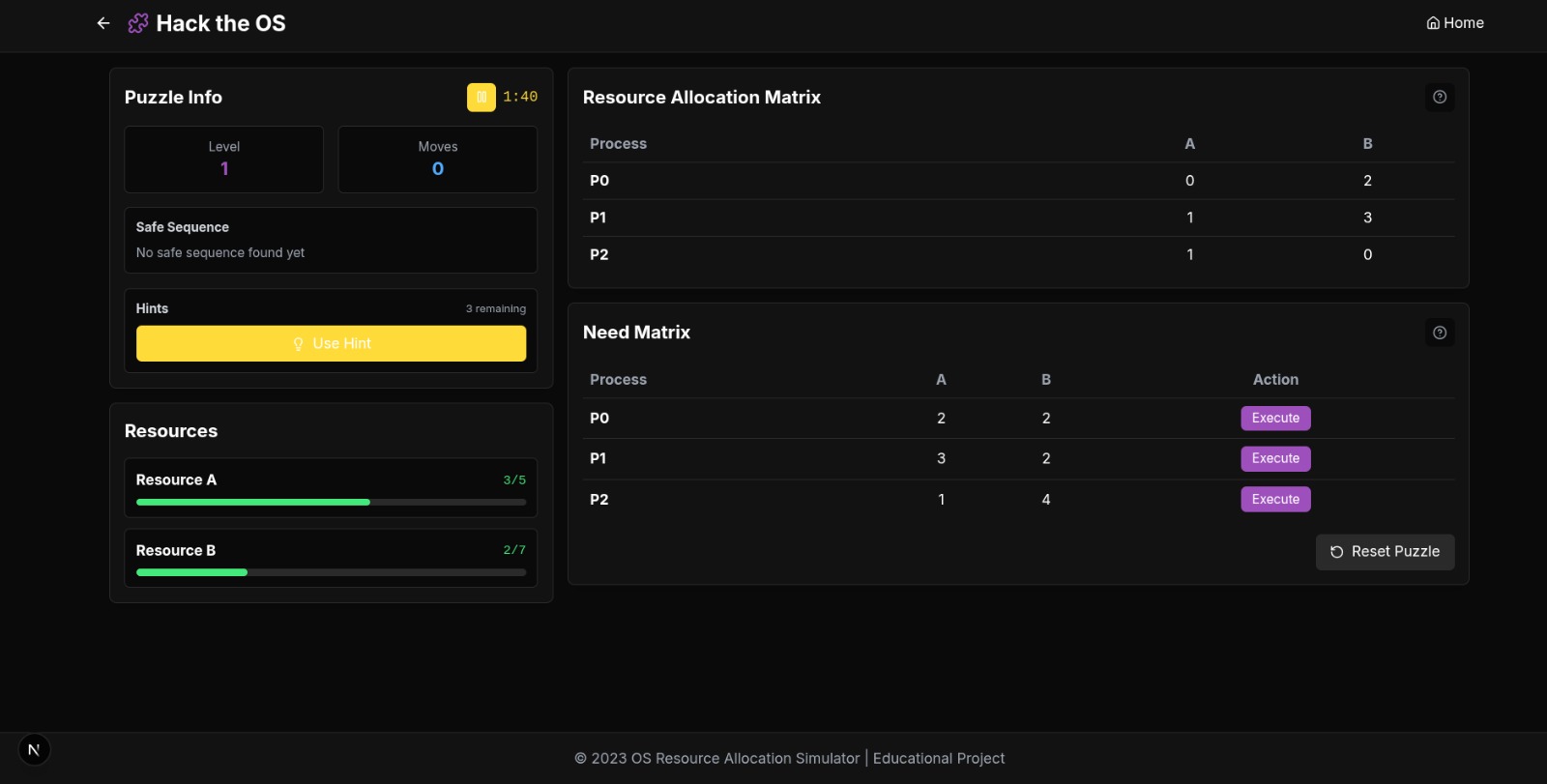
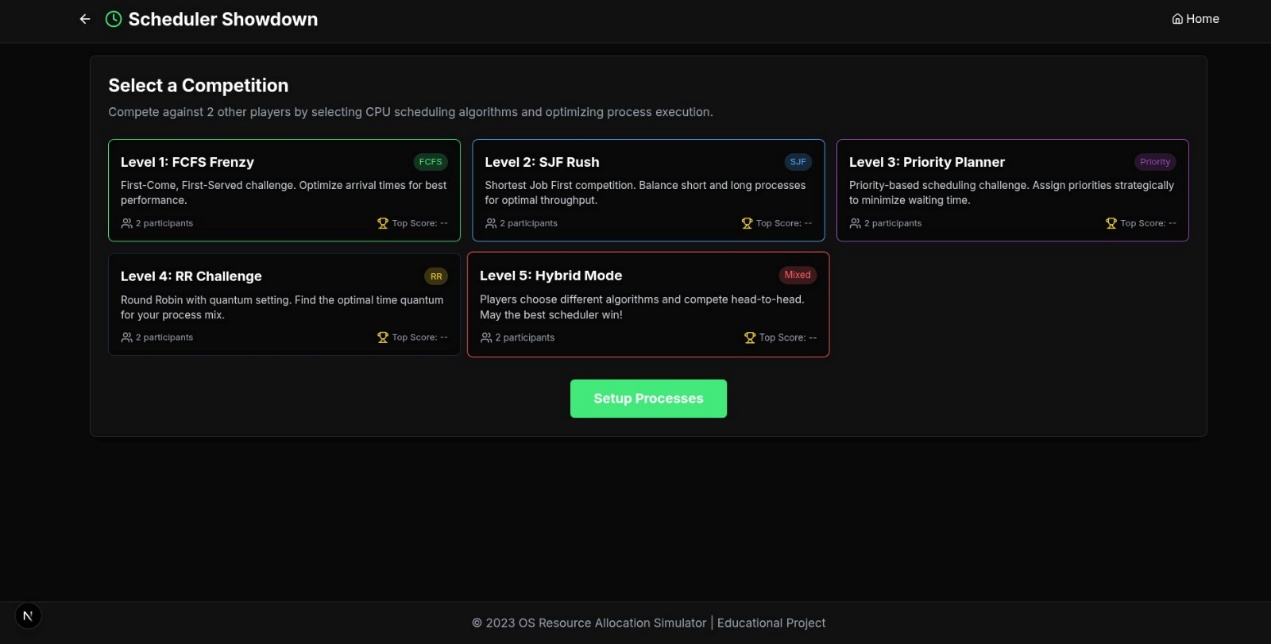
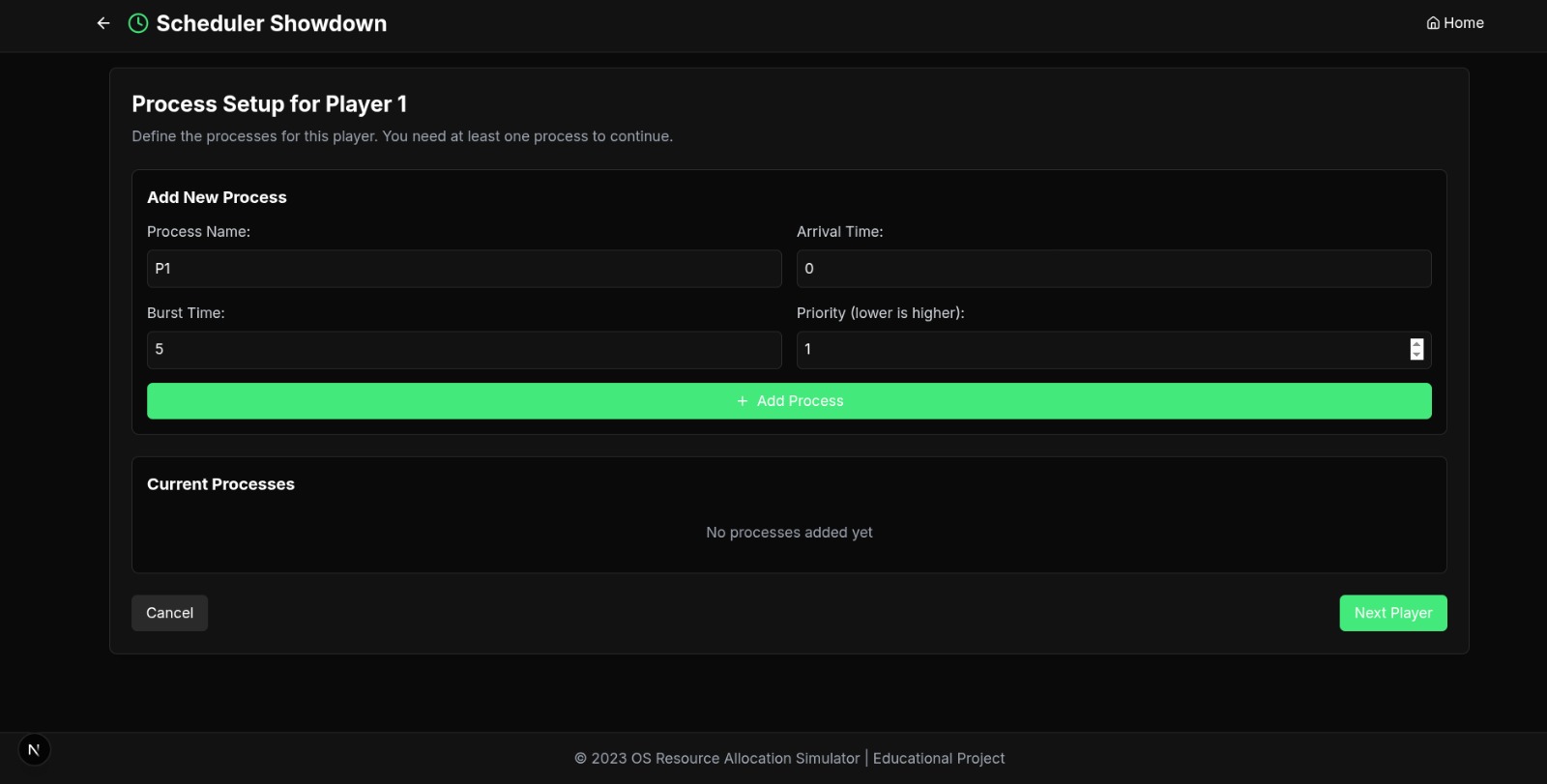
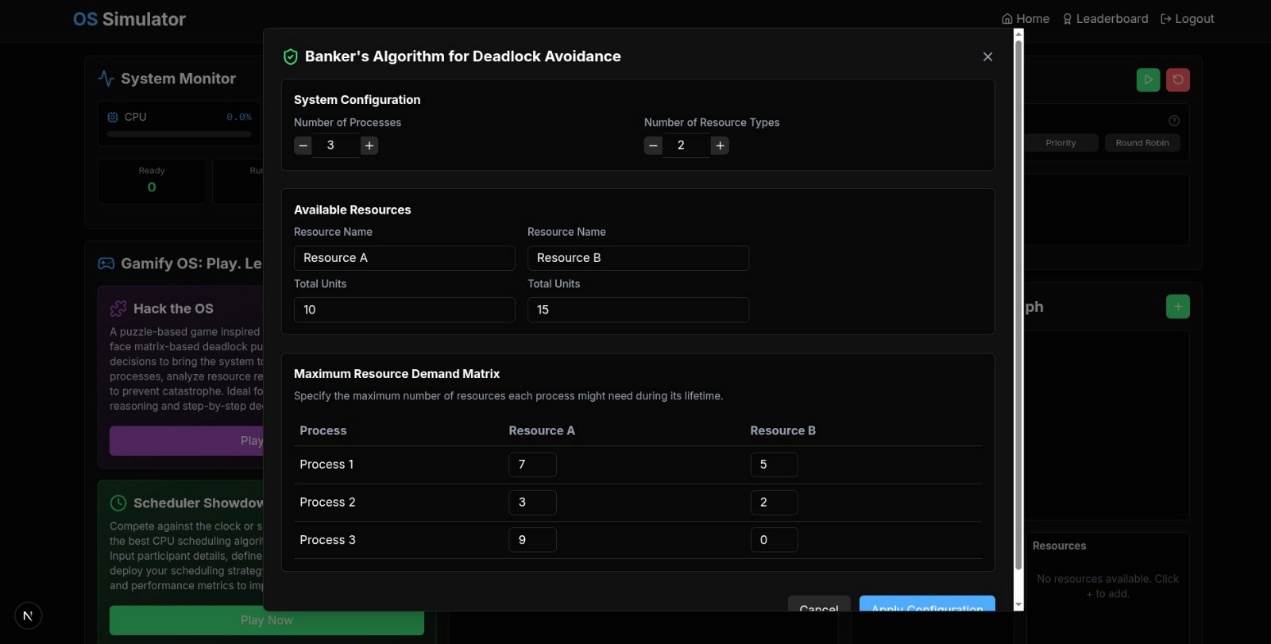
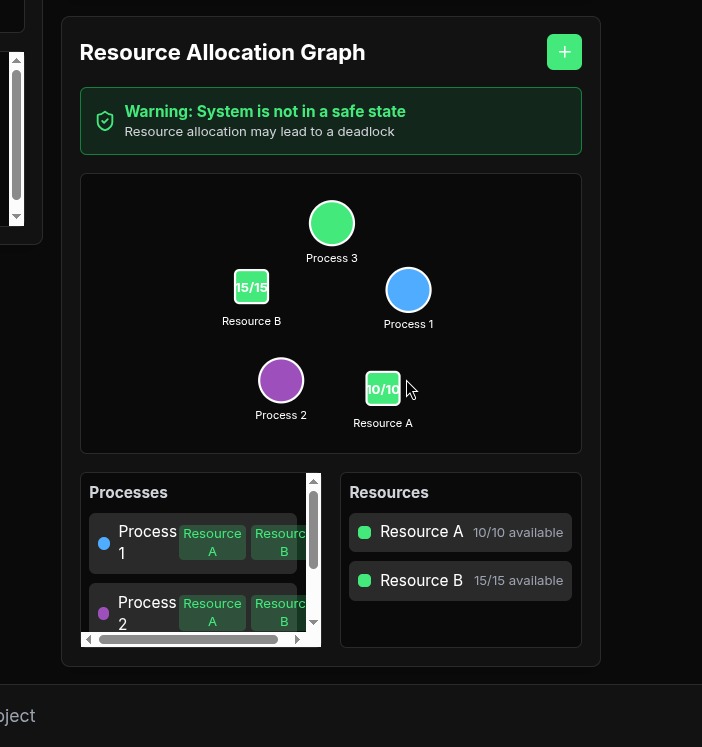
<p>© 2025 OS Resource Allocation Simulator | Educational Project</p>

</footer></main>

)

}

**RESULTS**

**OUTCOMES**

|  |  |  |
| --- | --- | --- |
| S NO. | REAL TIME SCENARIO | EXECUTION THROUGH PROGAME |
| 1. | Players experience how operating systems handle process scheduling and resource allocation similar to real computing environments. | By inputting arrival, burst, and priority values, players see immediate, visual results of scheduling logic in action |
| 2. | Players learn to recognize unsafe states and make real-time decisions based on the Banker's Algorithm—just as system administrators must do. | The program checks each resource request against system safety conditions, teaching players how deadlock prevention works in code. |

**APPLICATIONS**

The Resource Allocation Simulator, developed using concepts of operating systems, has a wide range of educational and practical applications for both students and professionals.

1. **Educational Tool**: The simulator acts as an interactive platform to help students understand complex CPU scheduling algorithms and deadlock avoidance strategies such as the Banker’s Algorithm. It simplifies the visualization of process scheduling and resource allocation concepts for academic use.
2. **Skill Development**: By engaging players in process scheduling tasks and resource management scenarios, it enhances their strategic thinking, analytical reasoning, and decision-making skills, which are essential in real-world operating system management.
3. **Operating Systems Laboratory Aid**: The game can be used as a laboratory tool to demonstrate and practically apply theoretical concepts of operating systems courses in a more engaging and understandable format.
4. **Professional Training**: IT professionals and system administrators can use the simulator to reinforce their understanding of process management, resource allocation, and deadlock avoidance mechanisms applicable in large-scale distributed or multi-user systems.
5. **Gamified Learning**: Through its interactive and competitive gameplay, the simulator encourages self-learning, experimentation, and problem-solving, making it suitable for hackathons, workshops, and online education platforms.
6. **Research and Development**: Researchers in operating systems can utilize the simulator as a testbed to experiment with custom variations of scheduling algorithms and deadlock prevention strategies in a controlled environment.

**REFERENCES**

Silberschatz, A., Galvin, P. B., and Gagne, G. (2018). *Operating System Concepts* (10thed.).Wiley.Covers CPU scheduling, deadlocks, and resource allocation concepts.

Tanenbaum, A. S., and Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.  
Explains process management, scheduling, and deadlock handling.

Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (8th ed.). Pearson.  
Focuses on OS architecture, scheduling, and deadlock prevention.

GeeksforGeeks. (n.d.). CPU Scheduling Algorithms. Retrieved from https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/  
Practical tutorials on FCFS, SJF, Priority, and Round Robin.

TutorialsPoint. (n.d.). Banker’s Algorithm. Retrieved from https://www.tutorialspoint.com/bankers-algorithm-in-operating-system  
Explanation and examples of Banker’s Algorithm for deadlock avoidance.

IEEE Xplore Digital Library. Studies on Process Scheduling Algorithms. Retrieved from<https://ieeexplore.ieee.org>  
Research articles on resource allocation and scheduling strategies.