**ROUND ROBIN CPU SCHEDULAR**

**A PROJECT REPORT**

**Submitted by**

**Shivanand Reddy 1RVU23CSE431  
Vikranth Subramanyam 1RVU23CSE538  
Kushal Rajpurohit 1RVU23CSE231**

*in partial fulfillment for the award of the degree*

*of*

**B Tech(Hons)**

*in*

**Computer Science Engineering**

****

**School of Computer Science and Engineering**

**RV University**

**RV Vidyaniketan,8th Mile, Mysuru Road, Bengaluru, Karnataka, India - 562112**

**NOVEMBER & 2024**

**A black and gold logo

Description automatically generated**

**School of Computer Science and Engineering**

RV University

RV Vidyaniketan,8th Mile, Mysuru Road, Bengaluru, Karnataka, India - 562112

**CERTIFICATE**

This is to certify that the project work titled "Round Robin Scheduling Simulator" is performed by **Shivanand Reddy (1RVU23CSE431),** **Vikranth Subramanyam (1RVU23CSE538)** and **Kushal Rajpurohit (1RVU23CSE231)** bonafide students of Bachelor of Technology at the School of Computer Science and Engineering, RV University, Bengaluru, in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science & Engineering, during the Academic year 2024-2025.

**Faculty Name Dr.Shobha G.**

Designation Dean

SOCSE SOCSE

RV University RV University

Date: 11 November 2024

**TABLE OF CONTENTS**

**Introduction/ Background**

**Problem statement**

**Methodology**

**Implementation**

**Results**

**Conclusion**

**Introduction**

The Round Robin (RR) scheduling algorithm is one of the simplest and most widely used CPU scheduling algorithms in time-sharing systems. Its primary goal is to allocate CPU time fairly among all processes, giving each an equal opportunity to execute by using a fixed time quantum, or time slice. In Round Robin scheduling, each process is assigned a specific time slice, after which it is either completed or preempted and placed at the back of the queue if further execution is required.

This cyclical approach ensures that no process is starved for CPU time, making Round Robin an effective method for managing interactive tasks in multi-user or multi-tasking environments, like operating systems and web servers. With its straightforward design, Round Robin helps maintain system responsiveness and provides predictable execution patterns, making it especially suitable for environments where fairness and user experience are important.

**Problem Statement**

In modern operating systems, efficient CPU scheduling algorithms are crucial for optimal resource utilization and ensuring fair process management. The Round Robin (RR) algorithm, a widely adopted preemptive scheduling technique, is designed to handle time-sharing systems by assigning fixed time quanta to each process in a cyclic order. The challenge lies in balancing system performance, minimizing response time, and ensuring fairness among multiple running processes. The problem involves creating a simulator that demonstrates the Round Robin algorithm's effectiveness in managing multiple tasks with varied burst times and arrival times. The simulator will visualize the Gantt chart, calculate metrics like Turnaround Time (TAT) and Waiting Time (WT), and assess the impact of different quantum values on system performance.

# Methodology

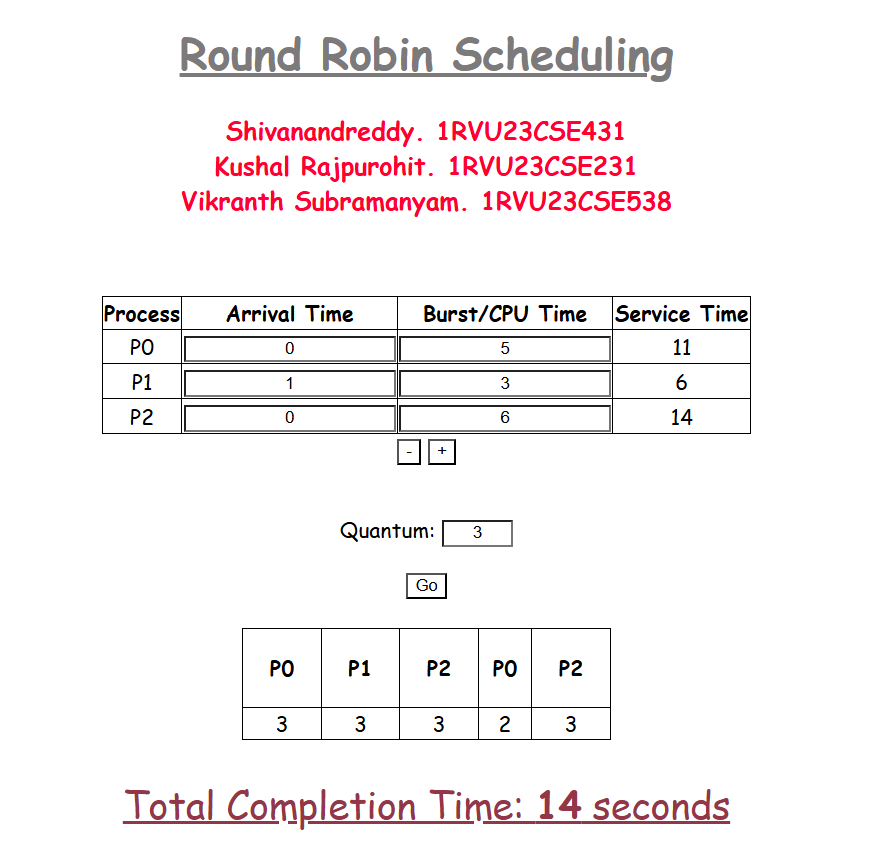
1. Requirements Analysis: Gather requirements for the Round Robin Scheduling Simulator, including user inputs for process details, quantum time, and real-time Gantt chart visualization.  
2. Design: Outline the UI/UX design, specifying input fields for process names, burst times, and arrival times, along with a section for the Gantt chart and result metrics. Use HTML, CSS, and JavaScript for front-end implementation.  
3. Development: Implement the Round Robin algorithm using JavaScript, creating a cyclic execution model. Use animations to represent process execution in the Gantt chart.  
4. Testing: Validate the simulator with various scenarios to ensure accuracy in process scheduling and time calculations. Adjust the quantum values to analyze system performance under different configurations.  
5. Analysis: Evaluate the simulator’s performance by comparing TAT and WT across different quantum settings.

# Implementation

1. Frontend Development: Utilize HTML for structuring input forms and the Gantt chart. Apply CSS to enhance the visual appeal of the simulator, ensuring clarity and accessibility. Use JavaScript to handle user inputs, execute the Round Robin algorithm, and animate the Gantt chart.  
2. Algorithm Implementation: Accept process data and quantum time as inputs. Maintain a queue of processes and manage execution based on the provided quantum. Dynamically update the Gantt chart and calculate TAT and WT.  
3. Visualization: Use JavaScript animations to display process transitions and execution periods. Update the total completion time and other metrics in real-time.

# Results

The simulator demonstrates how varying quantum times influence process management and overall system performance. Key observations include:  
- Fairness: All processes get equal opportunities, reducing the risk of starvation.  
- Efficiency: Quantum time significantly impacts the response time and context-switch overhead.  
- Visualization: The Gantt chart provides an intuitive understanding of process execution order.  
  
Testing revealed that smaller quantum values lead to higher context-switch frequency, while larger values increase the waiting time. The simulator successfully models the trade-offs inherent in the Round Robin algorithm.



# Conclusion

The Round Robin Scheduling Simulator effectively illustrates the principles of preemptive scheduling and showcases the impact of quantum time on system performance. It serves as an educational tool, highlighting the balance between fairness and efficiency in CPU scheduling. Future enhancements could include more advanced algorithms for comparative analysis and improved visualization techniques.