Project Title: Web-Based CPU Scheduling Algorithms Simulation

Course: Operating Systems (First Semester 2025/2026)

Instructor: Dr. Abdullah Al zaqebah

Team Members:

1. **Joud Kayyali – 3230601030**
2. **Ibraheem Abdullah – 3230605046**
3. **alnader ahmad – 32306010**

1. Introduction

CPU scheduling is one of the most important responsibilities of an operating system. In multiprogramming environments, many processes compete for CPU time. A CPU scheduling algorithm determines which process will run next, affecting key performance criteria such as CPU utilization, throughput, waiting time, turnaround time, and response time.

2. Project Objectives

The main objectives of the project are:

A. Implement the following scheduling algorithms: FCFS, SJF (non-preemptive), Round Robin (preemptive), and Preemptive Priority Scheduling.
B. Display a Gantt chart for each simulation run.
C. Calculate individual and average performance metrics, including:
    a. Waiting Time (WT)
    b. Turnaround Time (TAT)
    c. Response Time (RT)
D. Present a clean and usable interface (GUI) to support input and visualization**.**

3. Implemented Algorithms

A. First Come First Served:

FCFS is the simplest scheduling algorithm. Processes are executed in the exact order they arrive in the ready queue. Once a process starts executing, it runs until completion.

B. Shortest Job First:

SJF selects the process with the smallest burst time from the available processes in the ready queue. Like FCFS, once a process starts it runs until completion.

C. Round Robin:

RR is designed for time-sharing systems. Each process is executed for a fixed time quantum. If it does not finish within the quantum, it is placed back at the end of the ready queue.

D. Priority:

Each process has a priority value. The CPU always executes the highest priority process currently available. If a process arrives with higher priority than the running one, the CPU preempts and switches.

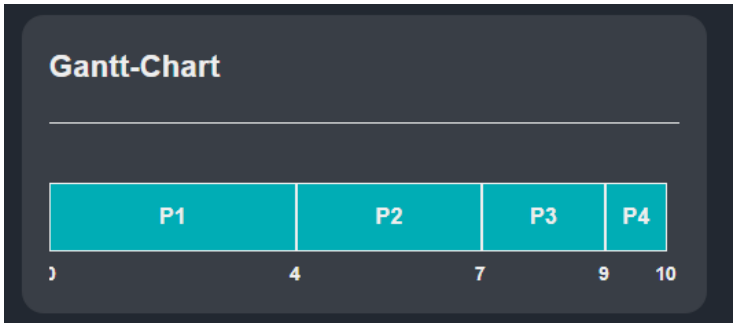
4. Test Cases, Results, and Gantt Charts

Test Case #1: Same Arrival Time

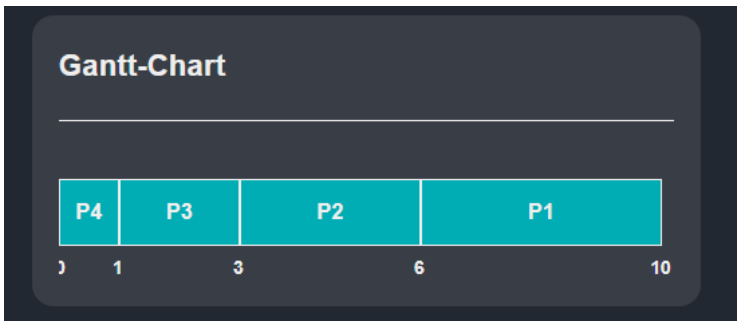
Process	Arrival Time	Burst Time	Priority
P1	0	4	2
P2	0	3	1
P3	0	2	3
P4	0	1	2

Gantt Charts:

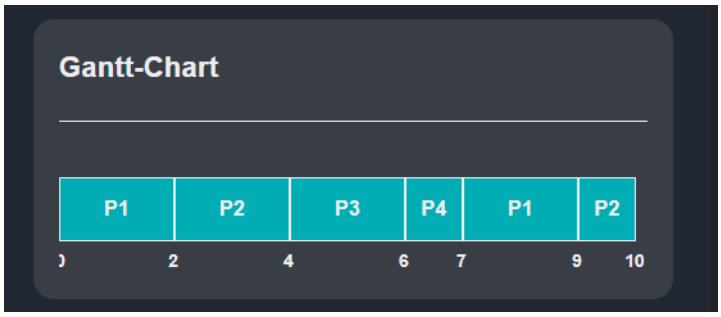
FCFS:



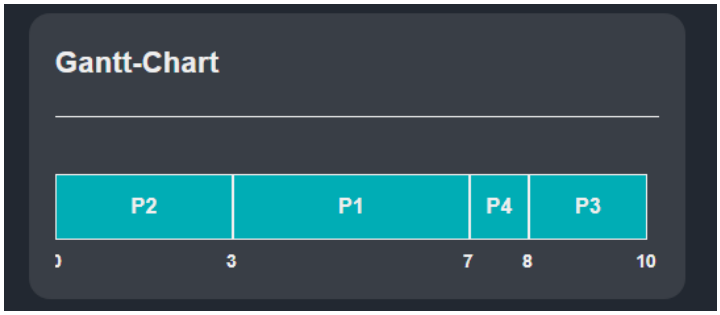
SJF:



RR(Q = 2):

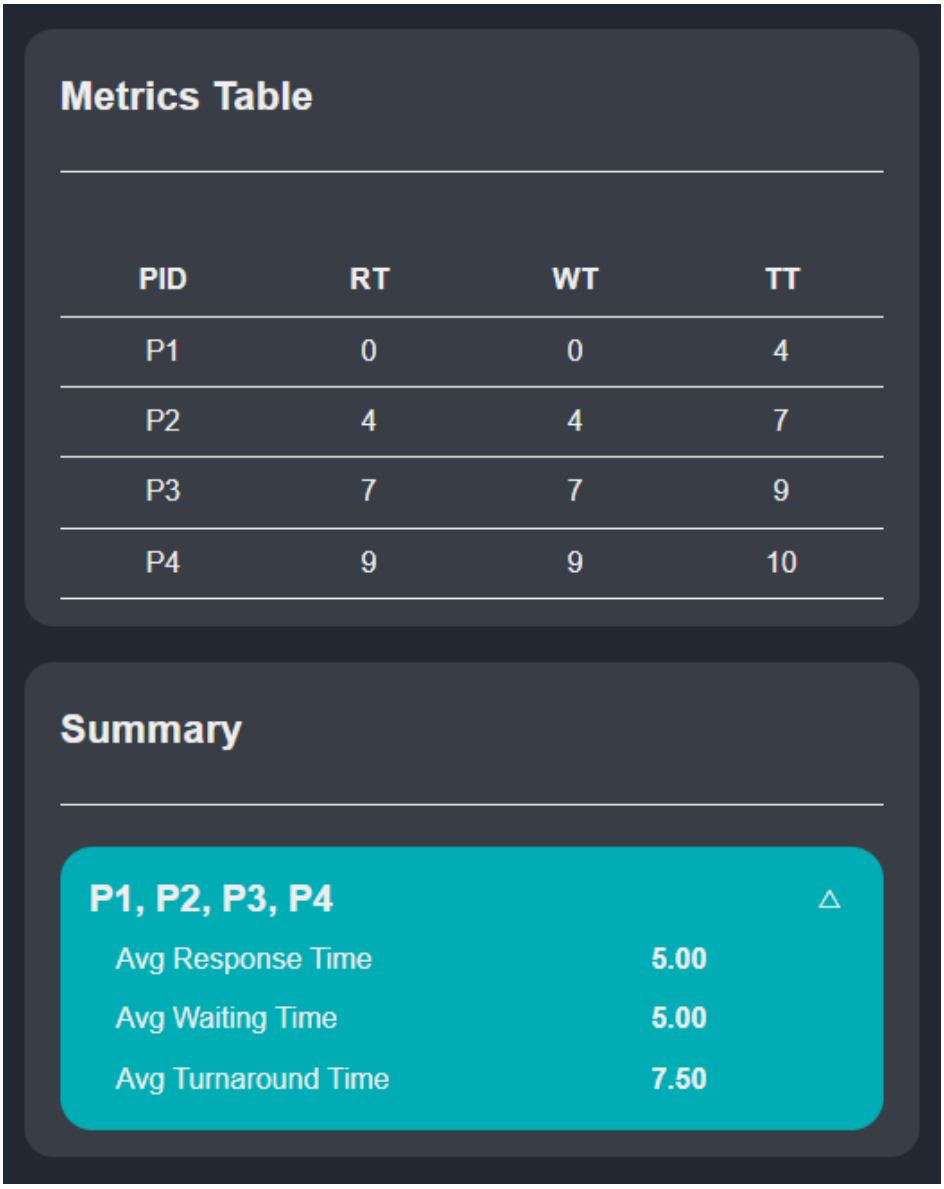


Priority:

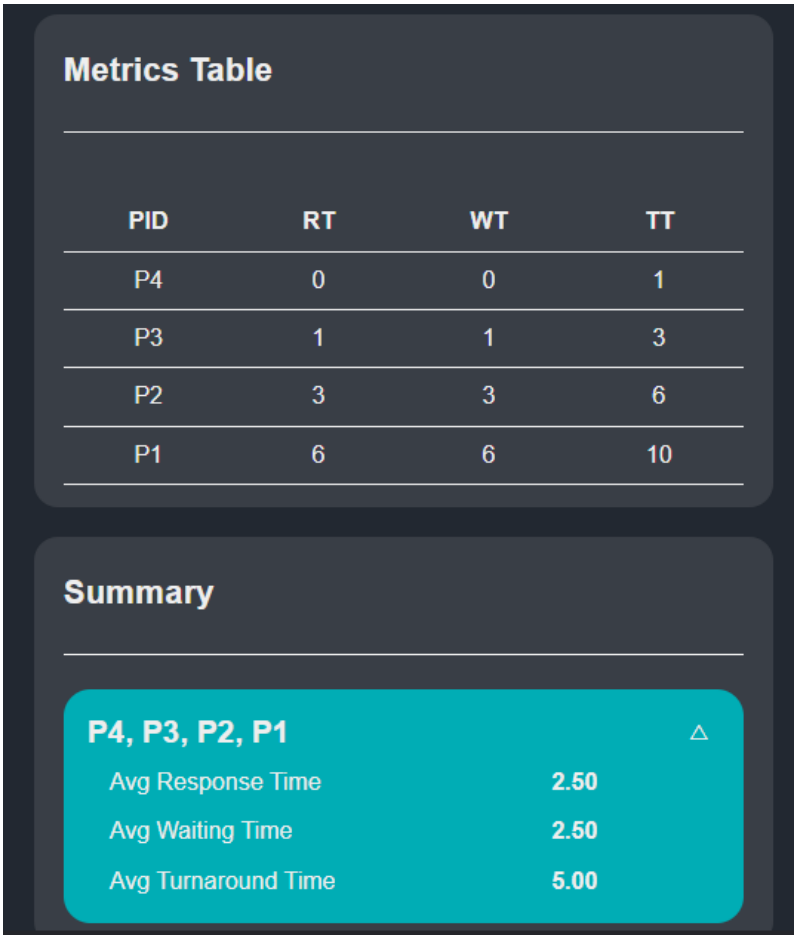


Metrics:

FCFS:



SJF:



Metrics Table

PID	RT	WT	TT
P4	0	0	1
P3	1	1	3
P2	3	3	6
P1	6	6	10

Summary

P4, P3, P2, P1 △

Avg Response Time	2.50
Avg Waiting Time	2.50
Avg Turnaround Time	5.00

RR(Q = 2):

Metrics Table

QT = 2

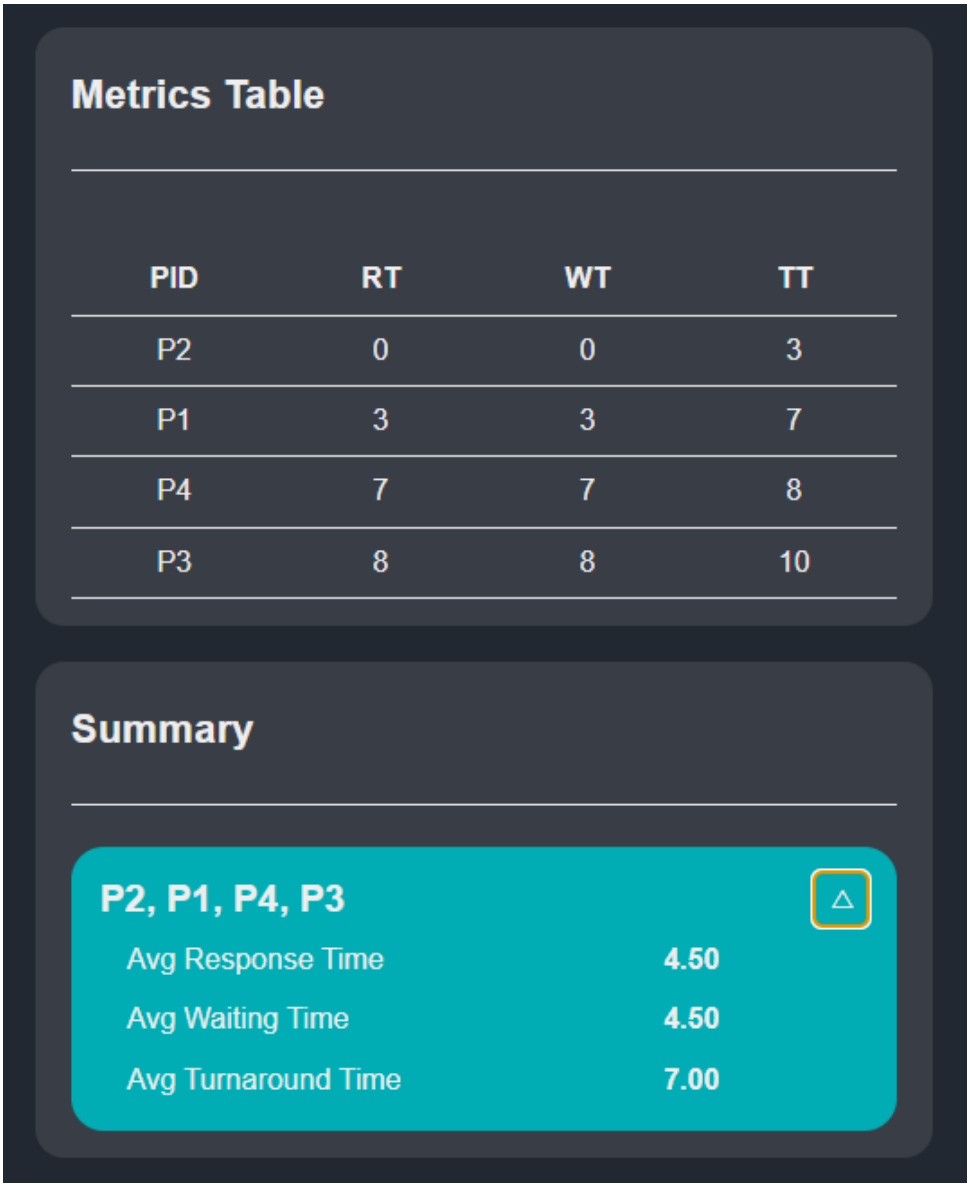
PID	RT	WT	TT
P1	0	5	9
P2	2	7	10
P3	4	4	6
P4	6	6	7

Summary

P1, P2, P3, P4

Avg Response Time	**3.00**
Avg Waiting Time	**5.50**
Avg Turnaround Time	**8.00**

Priority:

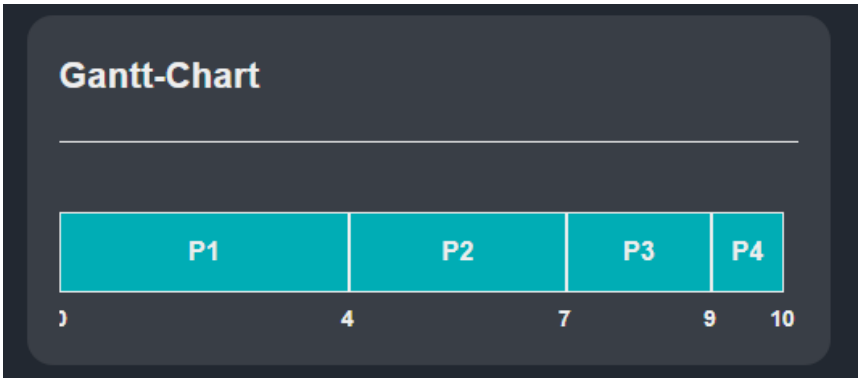


Test Case #2: Different Arrival Time

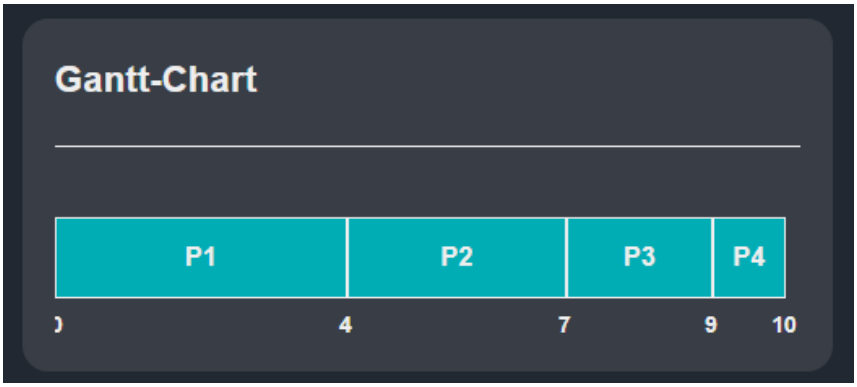
Process	Arrival Time	Burst Time	Priority
P1	0	4	2
P2	3	3	1
P3	5	2	3
P4	9	1	2

Gantt Charts:

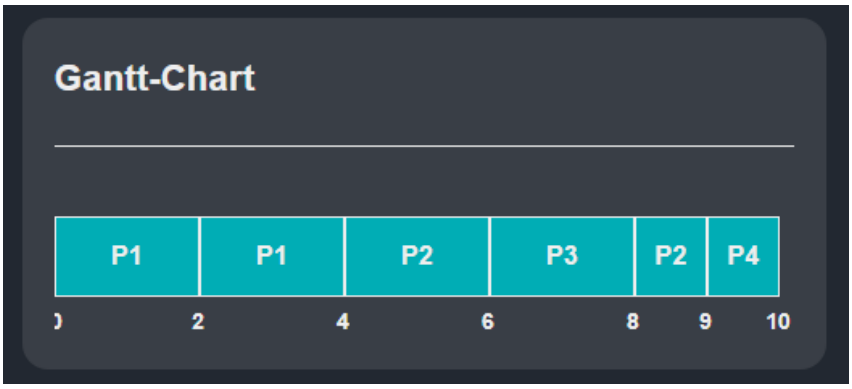
FCFS:



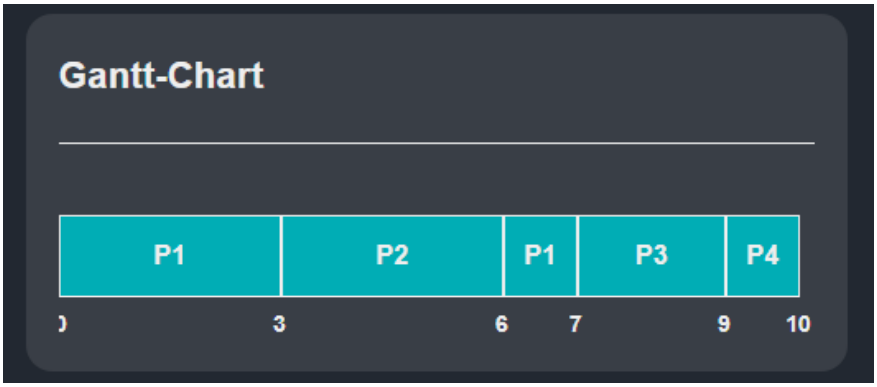
SJF:



RR(Q = 2):

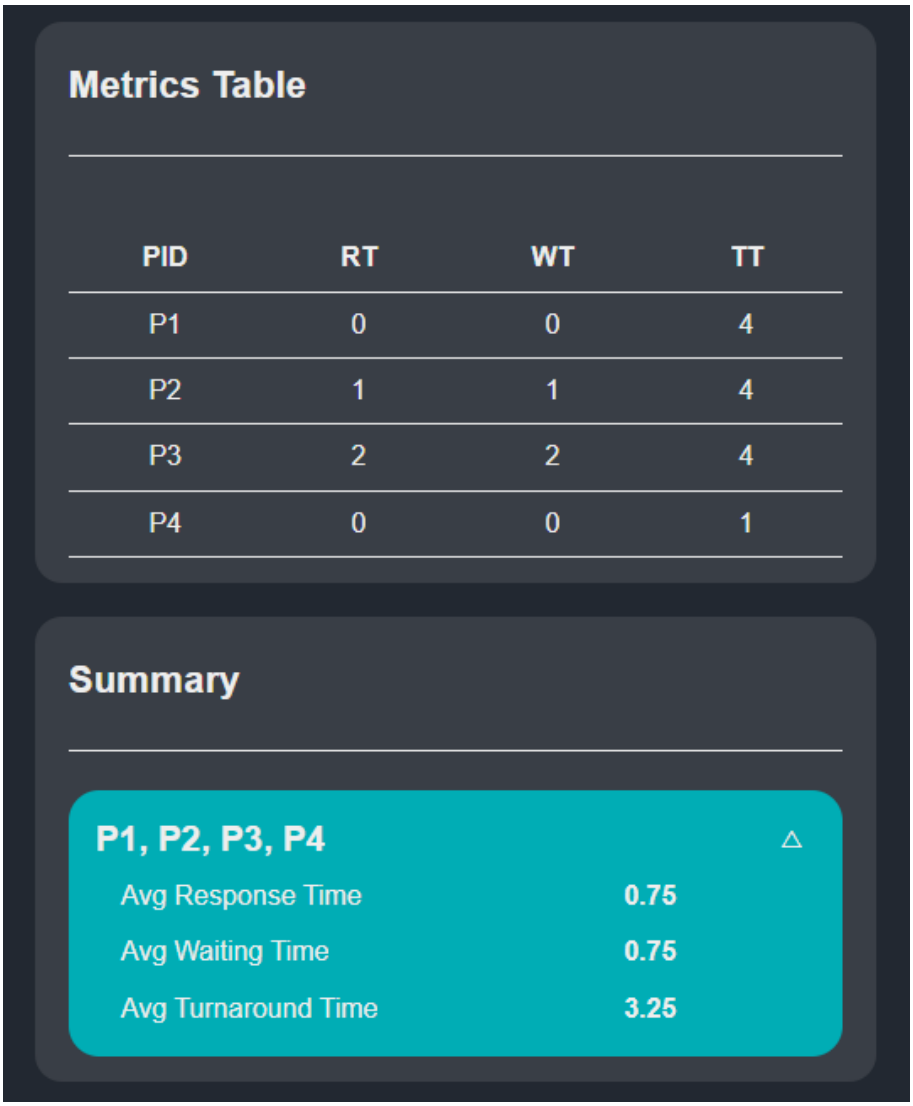


Priority:



Metrics:

FCFS:



SJF:

Metrics Table

PID	RT	WT	TT
P1	0	0	4
P2	1	1	4
P3	2	2	4
P4	0	0	1

Summary

P1, P2, P3, P4 △

Avg Response Time	0.75
Avg Waiting Time	0.75
Avg Turnaround Time	3.25

RR(Q = 2):

Metrics Table

QT = 2

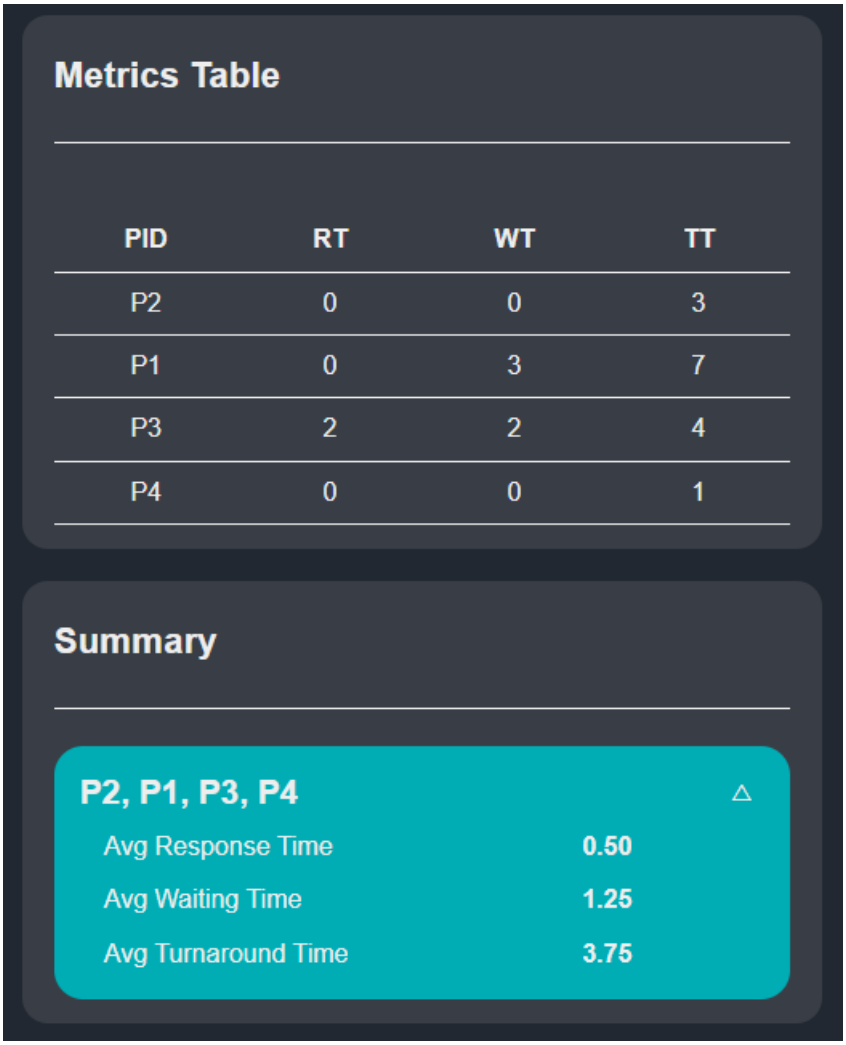
PID	RT	WT	TT
P1	0	0	4
P2	1	3	6
P3	1	1	3
P4	0	0	1

Summary

P1, P2, P3, P4 △

Avg Response Time	0.50
Avg Waiting Time	1.00
Avg Turnaround Time	3.50

Priority:



5. Conclusion

This project successfully implemented and tested multiple CPU scheduling algorithms in a web-based interactive simulator. The system allows users to input processes, visualize execution using a Gantt chart, and compute relevant scheduling metrics. Testing confirmed that different scheduling policies lead to significantly different performance outcomes depending on process arrival patterns, burst lengths, priorities, and time quantum.