A blue and white sign

Description automatically generated with low confidence

Operating Systems (Lab)

Instructor: Asif Ali

**PROCESS SCHEDULING**

Michael Ghouri (56116)

Arham Shahid Anwar (29942)

Mariyum Raees (35400)

**Introduction:**

Process scheduling plays a crucial role in operating systems, allowing efficient allocation of system resources to executing processes. The main objective of process scheduling is to maximize system throughput, minimize response time, reduce waiting time, and ensure fair resource allocation. This project aims to design and implement a process scheduling system and evaluate its performance using various scheduling algorithms.

**Background and Literature Review:**

Process scheduling algorithms are responsible for determining the order in which processes are executed on the CPU. Various algorithms have been developed and studied extensively in the literature. The commonly used scheduling algorithms include First-Come, First-Served (FCFS), Round Robin (RR), Shortest Job First (SJF), Priority Scheduling, and Multilevel Queue Scheduling.

In FCFS scheduling, processes are executed in the order they arrive. RR scheduling assigns a fixed time quantum to each process, allowing each process to be executed for a specific time before moving to the next process. SJF scheduling selects the process with the shortest burst time first. Priority scheduling assigns priority levels to processes and executes them based on their priority. Multilevel Queue Scheduling categorizes processes into multiple priority queues and assigns different scheduling algorithms to each queue.

**Methodology:**

For this project, the system was implemented using the C programming language. The process scheduling system utilizes a combination of data structures and algorithms to manage the execution of processes. The key components of the methodology are as follows:

**Data Structures:**

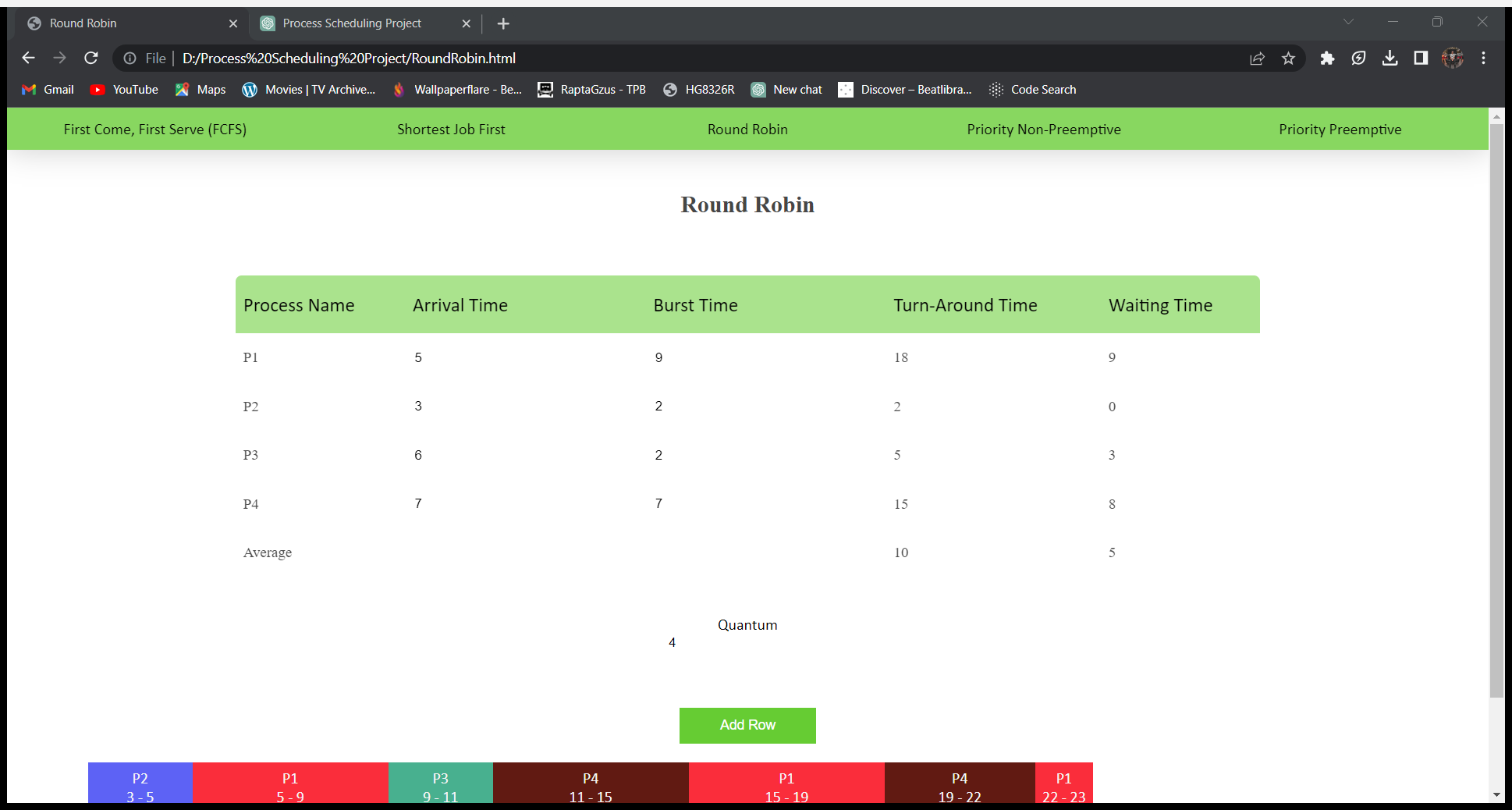
Process Control Block (PCB): Contains information about a process, such as its ID, arrival time, burst time, priority, and current state.

**Queue:** Used to organize the processes based on their scheduling parameters, such as arrival time, burst time, or priority.

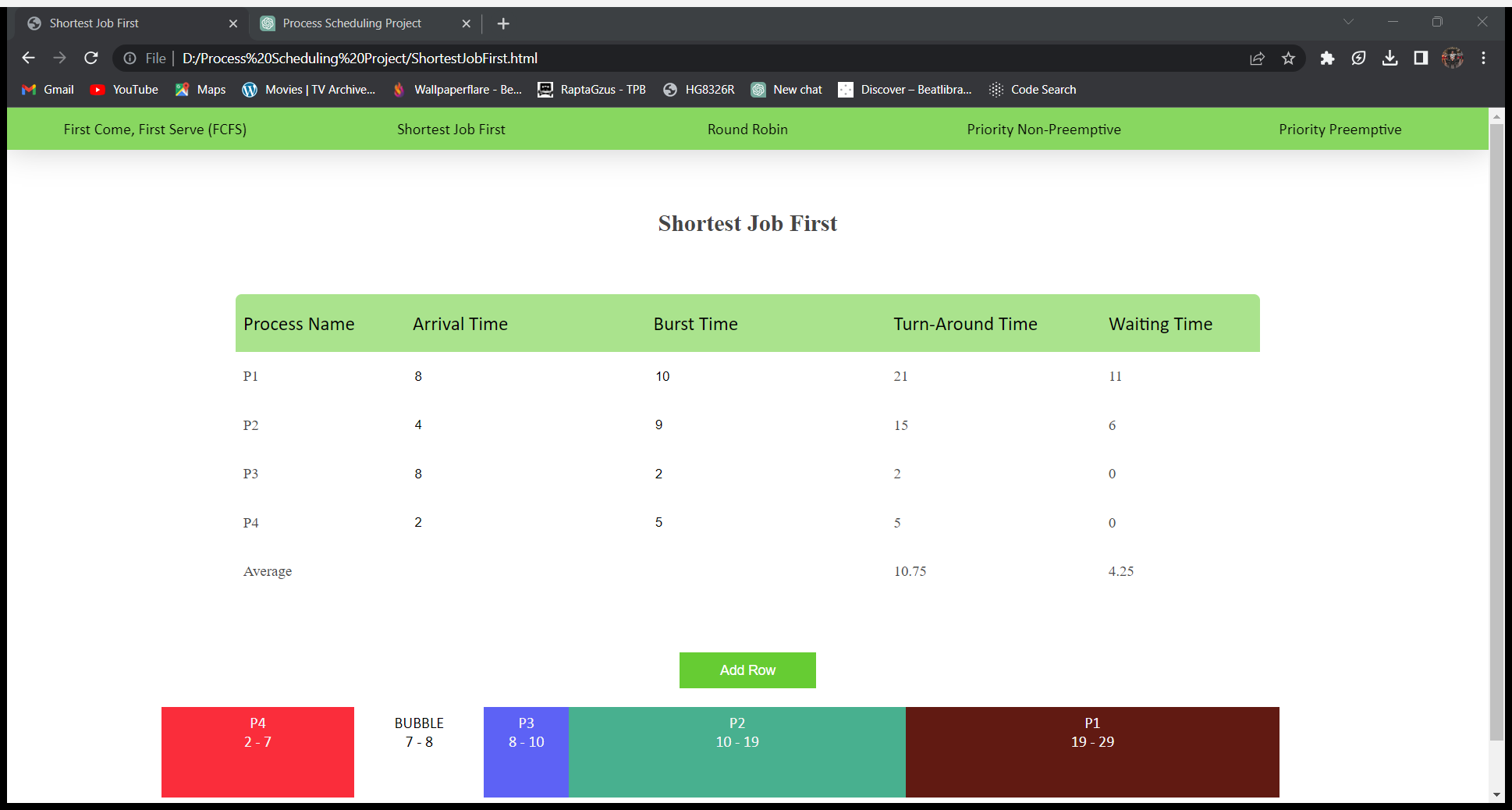
**Scheduling Algorithms:**

**FCFS:** Implements a simple queuing mechanism where processes are executed in the order they arrive.

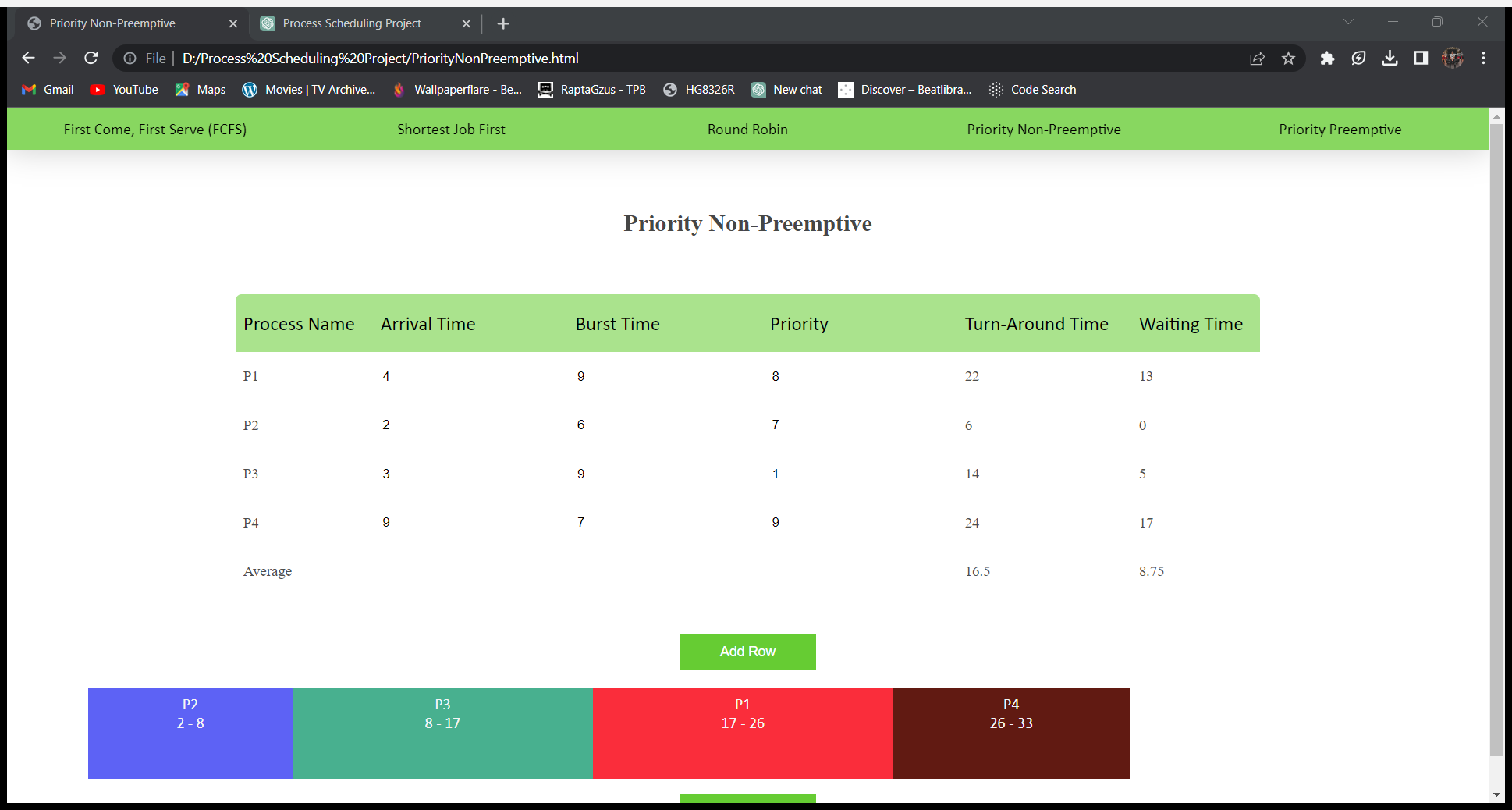
**Round Robin:** Assigns a fixed time quantum to each process, allowing them to execute for a specific time before moving to the next process.



**SJF:** Selects the process with the shortest burst time first.



**Priority Scheduling:** Assigns priority levels to processes and executes them based on their priority.



**Process Creation and Management:**

Process creation: Simulates the arrival of processes by generating random values for process attributes such as arrival time, burst time, and priority.

**Process termination:** Removes the completed processes from the execution queue.

System Design and Implementation

The system was designed following a modular approach to enhance maintainability and reusability. The main modules of the system are:

**Process Management Module:**

**Process Creation:** Randomly generates processes with unique process IDs and assigns them arrival time, burst time, and priority.

**Process Termination:** Removes completed processes from the execution queue.

**Scheduling Module:**

**FCFS Scheduling:** Implements a first-come, first-served queue where processes are executed in the order they arrive.

**Round Robin Scheduling:** Allocates a fixed time quantum to each process and cycles through the queue.

**SJF Scheduling:** Selects the process with the shortest burst time first.

**Priority Scheduling:** Executes processes based on their assigned priority level.

**Multilevel Queue Scheduling:** Assigns processes to priority queues and employs different scheduling algorithms for each queue.

**Experimental Evaluation**

To evaluate the performance of the implemented process scheduling system, several experiments were conducted. The following aspects were considered for evaluation:

**Test Environment and Setup:**

Specification of the hardware and software used in the experiments.

Determination of the system load by specifying the number of processes and their attributes.

**Performance Metrics:**

**Turnaround Time:** The time taken from process arrival to completion.

**Waiting Time:** The time a process spends waiting in the ready queue.

**Response Time:** The time taken from process arrival to the first response by the system.

**CPU Utilization:** The percentage of time the CPU is busy executing processes.

**Experimental Results and Analysis:**

Presentation of experimental data in the form of tables, graphs, or charts.

Comparison of the performance metrics for different scheduling algorithms.

Analysis of the results to identify strengths and weaknesses of each algorithm.

Results and Discussion

Based on the experimental evaluation, the following observations and conclusions were made:

The FCFS algorithm is simple to implement but may result in higher waiting times.

Round Robin provides fair CPU allocation but can cause high context switch overhead.

SJF scheduling minimizes waiting time for short burst time processes but may lead to starvation for long burst time processes.

Priority scheduling ensures that high-priority processes are executed first, but low-priority processes may suffer from starvation.

Multilevel queue scheduling provides a balance between different scheduling algorithms based on priority levels.

**Conclusion:**

In conclusion, this project successfully designed and implemented a process scheduling system with various scheduling algorithms. Through experimental evaluation, it was possible to compare the performance of different algorithms and gain insights into their strengths and limitations. The project contributes to a deeper understanding of process scheduling techniques and their impact on system performance.