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 - 3230601029

---

1. Project Overview

We propose to design and implement a comprehensive simulation tool for CPU scheduling algorithms. The project aims to visualize how processes are scheduled in an Operating System, calculate efficiency metrics, and provide a comparative analysis of different scheduling strategies. By developing this tool, we intend to deepen our understanding of process management, CPU utilization, and system performance metrics such as Turnaround Time and Waiting Time.

2. Selected Algorithms

We have selected the following four algorithms to cover both preemptive and non-preemptive scheduling strategies, meeting the minimum requirement of three algorithms:

1. **First Come First Serve (FCFS):** A non-preemptive algorithm serving processes in the order of arrival.
2. **Shortest Job First (SJF) - Non-preemptive:** Prioritizes the process with the smallest burst time.
3. **Round Robin (RR):** A preemptive algorithm designed for time-sharing systems, using a configurable Time Quantum.
4. **Priority Scheduling (Preemptive):** Assigns the CPU to the process with the highest priority, switching if a more important process arrives.

3. Data Representation (Process Control Block)

To simulate the OS environment effectively, each process will be represented as an object (Process Control Block - PCB) within the software. The data structure will include the following attributes:

- **Process ID (PID):** Unique identifier (e.g., P1, P2).
- **Arrival Time (AT):** The time at which the process enters the ready queue.
- **Burst Time (BT):** The total time required by the CPU to execute the process.
- **Priority:** An integer value indicating the urgency of the process.
- **Time Quantum:** (Global parameter for Round Robin).
- **Computed Metrics:** Completion Time, Turnaround Time, Waiting Time, and Response Time.

4. Tools and Implementation Strategy

We have chosen a **Web-Based approach** to ensure cross-platform compatibility and to deliver a high-quality Graphical User Interface (GUI) as a project enhancement.

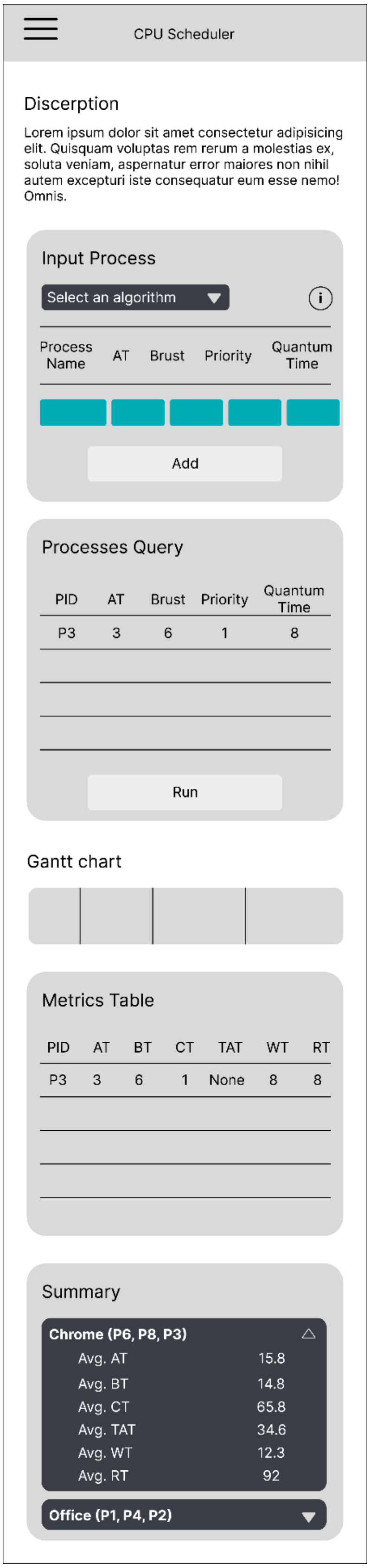
- **Core Logic:** JavaScript (ES6+) for implementing the scheduling algorithms and queue management.
- **User Interface:** HTML5 and CSS3 (Modern Flexbox/Grid layouts) to create a responsive dashboard.
- **Tools:** VS Code, Git/GitHub for version control.

5. Input and Output Design

The system is designed to be user-friendly and interactive:

- **Input Format:**
    - A dynamic form allows users to add processes manually (entering Arrival Time, Burst Time, and Priority).
    - A configuration panel to select the Algorithm and set the Time Quantum (if applicable).
    - *Reset* and *Run* controls to manage the simulation flow.
- **Output Format:**
    1. **Dynamic Gantt Chart:** A visual horizontal bar chart representing the execution timeline of processes.
    2. **Results Table:** A detailed breakdown of start times, finish times, waiting times, and turnaround times for each process.

3. **Comparative Analysis:** A summary section displaying the **Average Waiting Time** and **Average Turnaround Time** to analyze algorithm efficiency.



6. Project Timeline

We will follow the course schedule strictly:

- **Phase 1 (Current):** Submission of this proposal and initial design.
- **Phase 2 (Week 9):** Implementation of the FCFS algorithm and core UI structure.
- **Phase 3 (Week 12):** Completion of all algorithms (SJF, RR, Priority), final testing, comparative analysis, and oral presentation.

Project Resources & Repository

To ensure transparency and version control throughout the development of lifecycles, we have established the following resources:

- **Source Code Repository (GitHub):**

  https://github.com/JoudN2001/OS-Scheduler-Simulation-Project

- **UI/UX Design Prototype (Figma):**

  https://www.figma.com/design/vTsK3KUjGWgyldyTu8tXic/OS-Project?node-id=0-1&t=nBNpk7a6tS8u1rEd-1