**CPU Scheduling Algorithms**

**TEAM – THE 3AM CLUB**

**RA2211030010069 – HARNOOR CHAUHAN**

**RA2211030010083 – SHREYAN MUKHERJEE**

**RA2211030010130 – DIYA KALRA**

Aim:

The aim of this project is to implement and compare different CPU scheduling algorithms for process management in operating systems. This web-based application provides a visual and interactive platform to understand and analyse the behaviour of various CPU scheduling algorithms. It allows users to input a set of processes with different CPU Burst Times and I/O Burst Times and observe the scheduling and execution of these processes using different algorithms. The project aims to help students, system administrators, and enthusiasts gain insights into how different scheduling algorithms affect system performance.

Objectives:

Implement 8 different CPU scheduling algorithms.

Allow users to define processes with varying CPU Burst Times and I/O Burst Times.

Generate Gantt Charts and Timeline Charts to visualize the scheduling.

Provide an animation of the time log to show the execution of processes over time. Compare the performance of scheduling algorithms in terms of Average Completion Time, Turn Around Time, Waiting Time, and Response Time.

Enhance user understanding of CPU scheduling algorithms through an interactive web application.

Technologies Used:

HTML: HTML is used for creating the structure and layout of the web application.

CSS: CSS is used for styling and formatting the user interface, making it visually appealing and user-friendly.

Vanilla JS: JavaScript is used to implement the scheduling algorithms, handle user interactions, and create dynamic content.

Google Charts: Google Charts library is employed to generate Gantt Charts and Timeline Charts, allowing users to visualize the scheduling of processes.

By combining these technologies, this project aims to provide a comprehensive and educational tool for exploring and understanding CPU scheduling algorithms, making complex operating system concepts more accessible and engaging.

