The SAC

Scheduling **A**lgorithms **C**lub

CSL3030 Project Report

Indian Institute of Technology Jodhpur

Authors

Neha Soni (B19CSE058)
Nikhil Raj (B19CSE059)
Nandini Verma (B19CSE057)
Ekta Choudhary (B19CSE030)

Mentor

Jayesh Budhwani (M21CS007)

ABSTRACT

Operating system is an important software program that manages and operates various tasks and processes happening in computers. Scheduling these processes in an appropriate manner is a noteworthy issue. These schedules impact a computer's efficiency, computation time and throughput. There are various scheduling algorithms that a computer takes into account and having a clear and comprehensive understanding of the working of the same is crucial in achieving better implementation of schedulers in computers. Functioning of these scheduling algorithms could be quite complex and difficult to comprehend.

The SAC proposes a design and implementation of a web platform service, focused on visualising operating system's scheduling algorithms in a more easier and interactive way. This project demonstrates visualisation of various scheduling algorithms like FCFS, RR, SJF, on user defined scenarios and furthermore elaborates the design, implementation and working of the web App, including all the methodologies adopted, result achieved and working analysis.

Keywords : Operating system, Scheduling algorithms, Visualisation , FCFS, RR, SJF

INTRODUCTION

Operating system is an important software program that manages and operates various tasks and processes happening in computers. Scheduling these processes in an appropriate manner is a noteworthy issue as an inappropriate schedule might lead to computing less number of processes in a specific duration of time. These processes have been scheduled in a specific manner, taking into account various scenarios, to increase the throughput and efficiency of computers. These schedules impact a computer's efficiency, computation time and throughput values. There are various scheduling algorithms that a computer takes into consideration while performing the scheduling process to order various coming processes in a specific way.

Different algorithms work in a different peculiar way and Functioning of these scheduling algorithms could be quite complex and difficult to understand. But, having a clear and comprehensive understanding of the working of the same is crucial in achieving better implementation of process schedulers in computers. Any incorrect understanding might lead to a negative impact on a computer's functioning.

The SAC proposes a design and implementation of a web platform service, with prime focus on visualising these various operating system’s scheduling algorithms in a much easier and interactive way. This project demonstrates visualisation of various scheduling algorithms like FCFS (First Come First Serve), RR (Round Robin) , SJF (Shortest Job First), on certain user defined scenarios through the help of certain dynamic implementations, explained in methodology, and furthermore elaborates the proposed design, its implementation and working of the web App, including all the methodologies adopted, result achieved and working analysis of scheduling algorithms.

PROBLEM FORMULATION

1. Problem Statement

Unavailability of a common accessible free platform, in current time, where operating system scheduling algorithms could be visualised on user defined scenarios in real-time.

Having some user defined scenarios, a common platform is needed to elaborate and demonstrate various algorithms in a visually interactive way along with a comprehensive elaboration.

2. Problem Context

In the context of this project, three algorithms have been taken into consideration, which are First Come First Serve, Round Robin and Shortest Job First, allowing users to choose one algorithm and providing required input values accordingly. Problems are also elaborated in a comprehensive way with depicting algorithm description and algorithm working.

OBJECTIVE

The SAC aims at proposing a lightweight and snappy solution to the problem mentioned beforehand. An interactive and appealing React JS based web platform design has been proposed and implemented for all the OS enthusiasts, with prime focus on visualising the operating system's scheduling algorithms, through various dynamic tables and gantt charts, in a more easier, elaborative and comprehensive way, on user defined scenarios, in real time.

The main highlights of the web application are :

1. Responsive and Aesthetic platform hosted publicly for educational purpose.
2. Dynamic and interactive User Interface allowing users to explore the results, any number of times, by giving different inputs.
3. Algorithm working and elaboration along with necessary steps have been depicted

METHODOLOGY

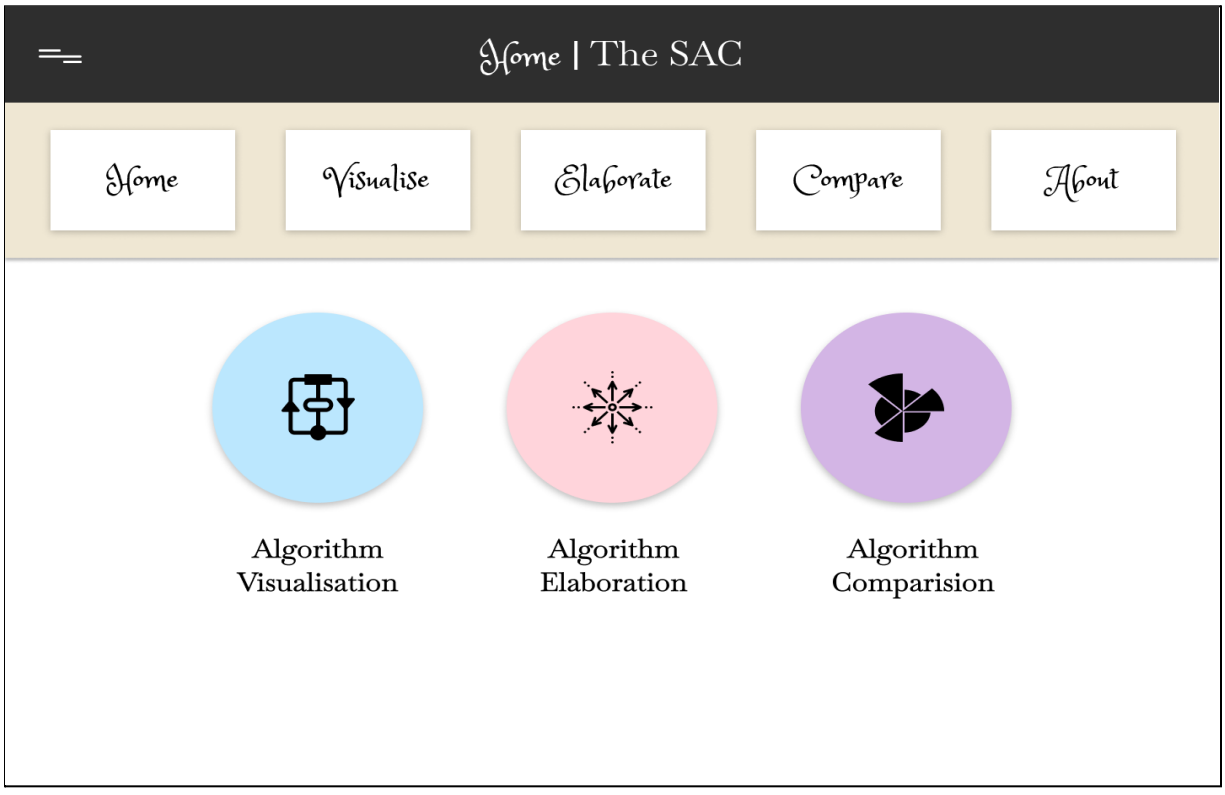
Till now, as we have come across the problem formulation where an interesting and accessible platform was needed to let all the Operating Systems enthusiasts explore the functioning of different CPU Scheduling Algorithms using different informative data tables and charts, these were the phases in which the entire work was divided :

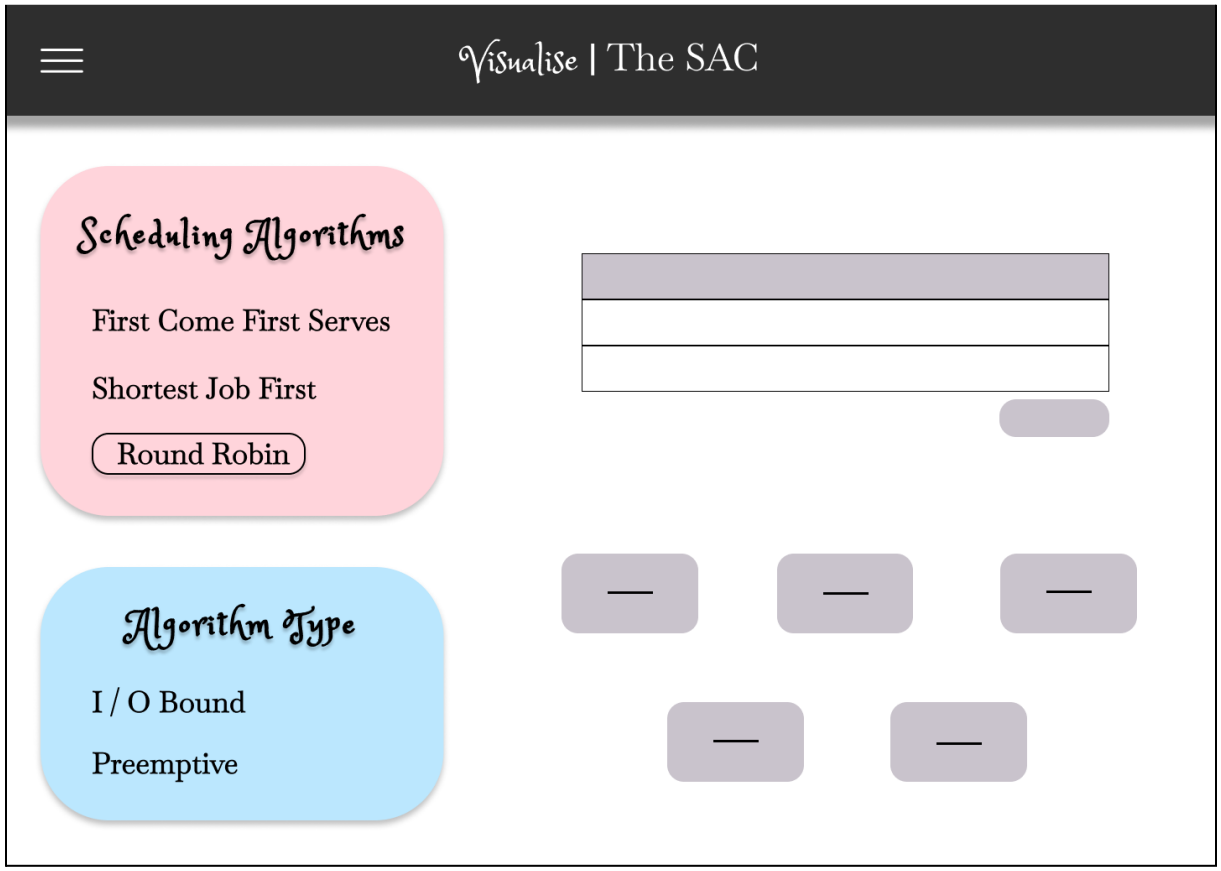
1. Designing a wireframe for the entire project
2. Implementation of different components
3. Testing and Debugging

Now one-by-one we'll go through all the phases and the methodology adopted to ensure the proper functioning of all the entities in a particular phase.

1. Wireframe Design

Now in the starting phase where it was needed to first create and analyse the entire design from scratch, we mainly used Figma as a platform to achieve the same. Attached are the following layout screenshots that were finalised after a large number of iterations to modify the entire design to accommodate all the features we thought of adding :





Here in this attached screenshot, to give user the freedom to explore the algorithms using different inputs, these were the different fields planned to achieve the motto as described :

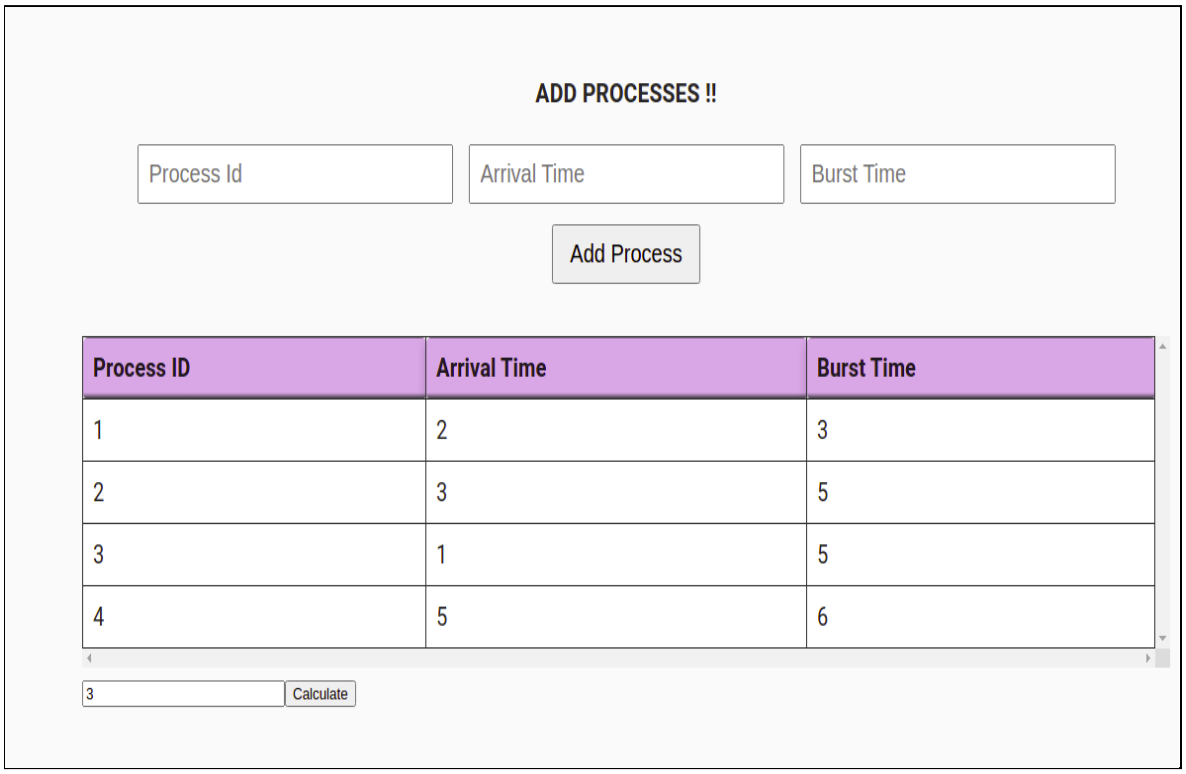
1. A Dynamic table to encompass all the values the user wants to feed (top-right).
2. A way to present the running of the algorithms in real time (bottom-right).
3. Different CPU scheduling algorithms to choose from (top-left).
4. An other field to represent the algorithm that is currently running (bottom-left)

Note : One thing to note is that, these were the planned features which were expected to be added. In the actual build, some features were added and removed in accordance with the team meetings.

2. Components Implementation

Listed below are the several components implemented to ensure that all the use-cases could be viable and the dynamic nature of the site can be maintained.

1. Dynamic Table for inputs



Asking the Process ID, Arrival Time and Burst time, this table can accept any number of processes and expands as the user desires and can provide the result accordingly. The table attached is taken from the UI of Round Robin algorithm, hence it has one extra field (bottom-left) which is accepting time-quantum for the processes.

All the edge cases and exception handling is taken care so that the site does not crash after accepting the invalid arguments from the user.

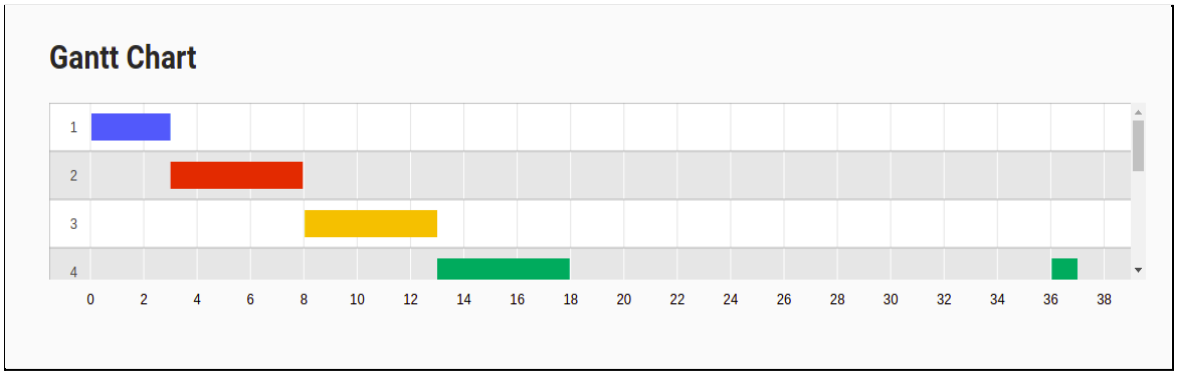
2. *Tabular representation of the output*

Output Table

Process ID	Arrival Time	Burst Time	Waiting Time	Turn Around Time
1	2	3	0	3
2	3	5	9	14
3	1	5	11	16
4	5	6	13	19

So as mentioned, after the algorithm completes its execution, all the required data is provided as an output which are highlighted in the table which are Process ID, Arrival Time, Burst Time along with the calculated results like Waiting Time and Turnaround time for every individual process.

3. *Interactive Gantt charts for Process Scheduling*

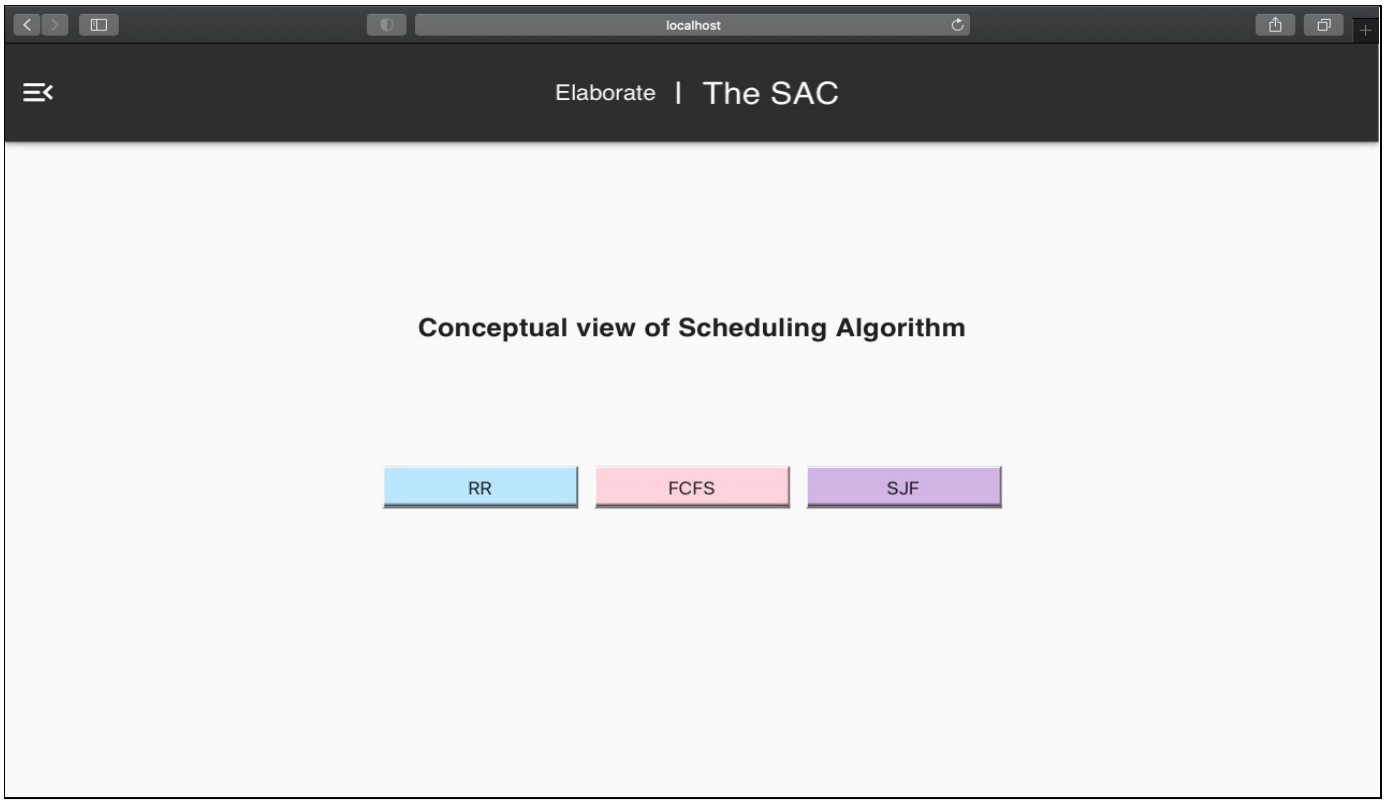


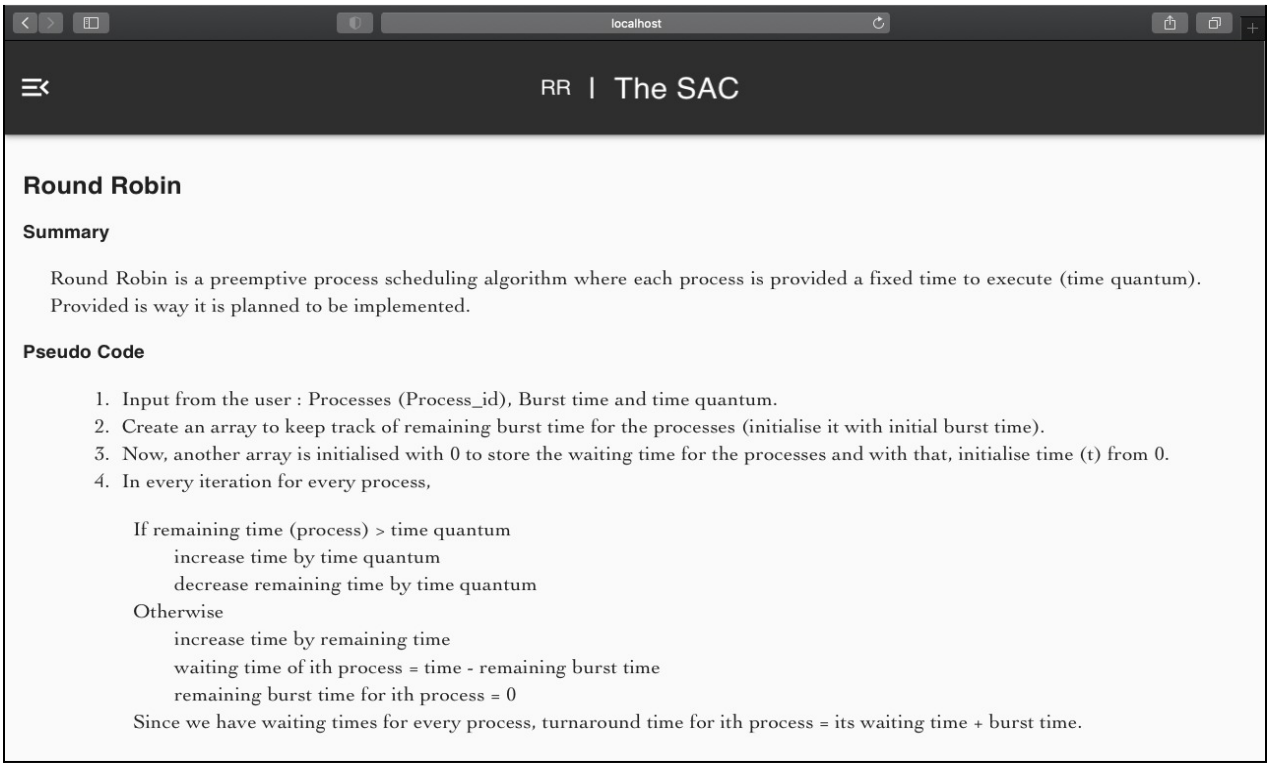
Implemented using Google's React-Google_Charts module, the presented gantt chart gets the input in the runtime for every process and when the algorithm completes its execution, it processes the information gathered to form a timeline as displayed in the

screenshot attached. Further, on hovering the mouse over it, all the process information is displayed to the user. It can also accommodate any number of processes and identifies the process with the process_id attached to it. So when a process comes back for execution after running for a specific period (like in Round Robin), it's execution information is added in that id only (as we can see for process_id 4 in the screenshot attached).

4. ***Informative content for the user***

Here in this component, all the information related to the algorithms implemented are presented. These components contain Summary and the Pseudo code for all the three algorithms for the simplified and better understanding of them.





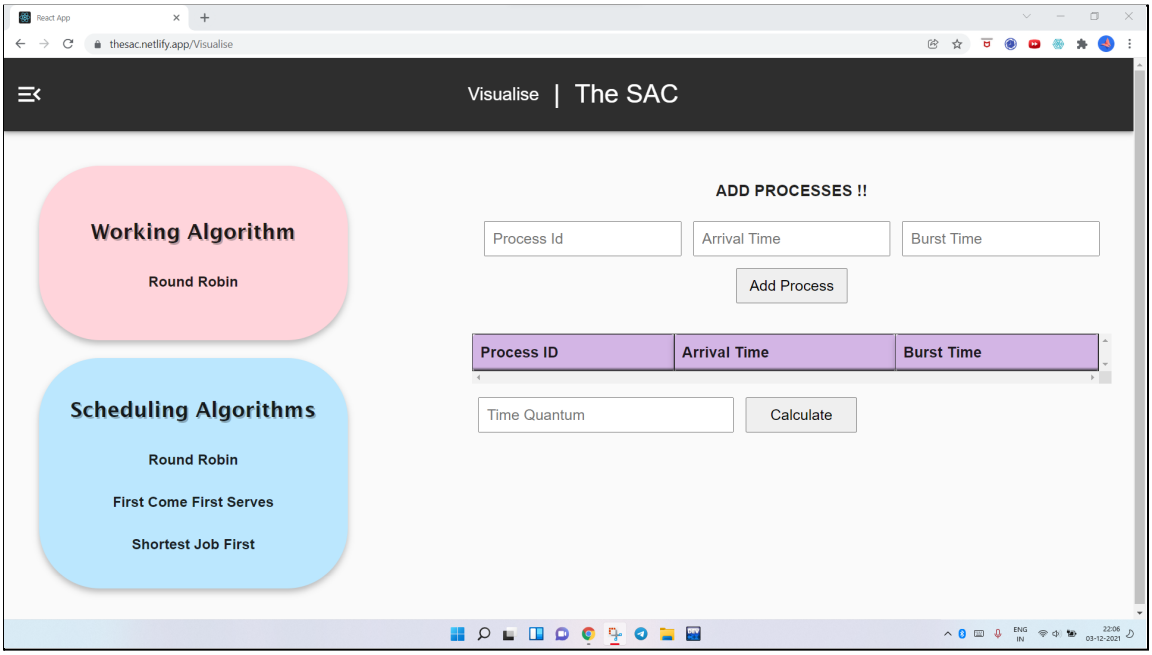
3. Testing and Debugging

1. In this phase, the website was tested with different inputs and outputs and result verification was done.
2. When this phase was completed, the site was hosted on the web for further reach of users.

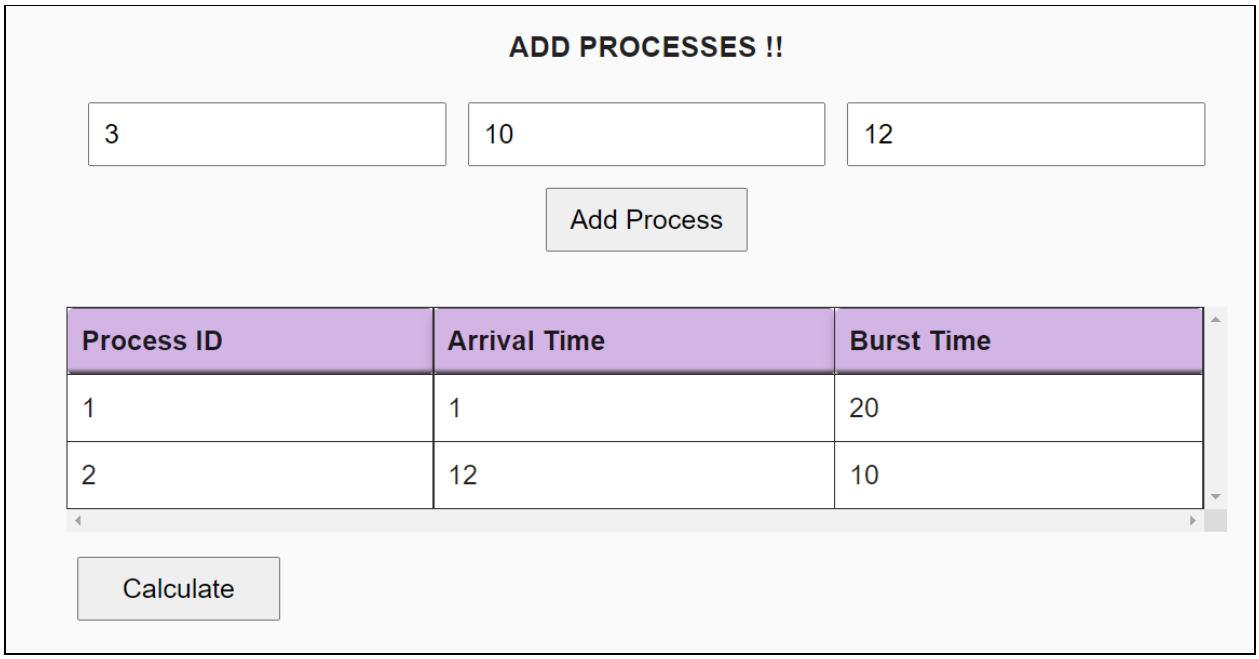
RESULT AND ANALYSIS

The final visualisation page looks like the following

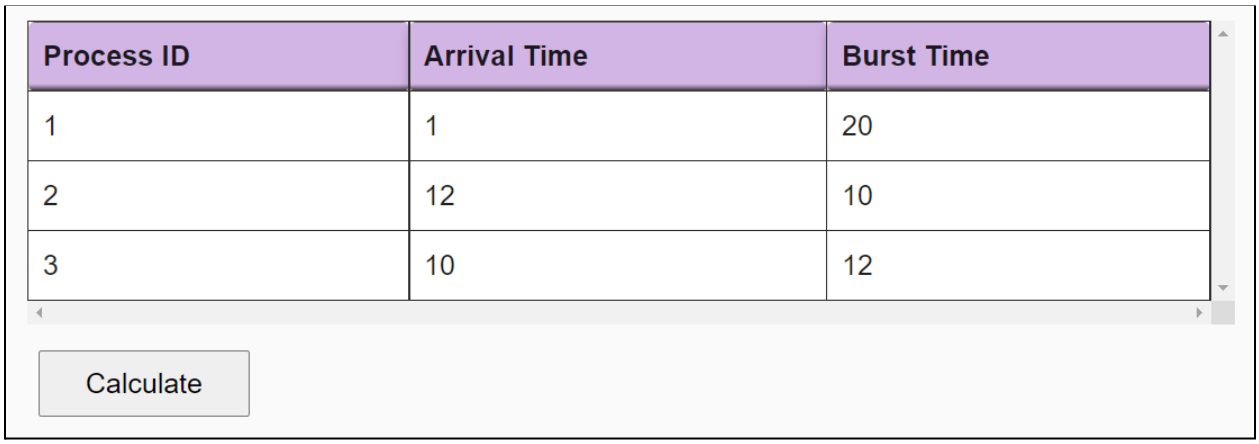
- Users can choose the algorithm provided in" Scheduling Algorithms" Blocks , here Round Robin is selected by default. Let us take an example where we Select FCFS (First come first serve) algorithm.



- After selecting the algorithm we can provide Process data (ID, Arrival time and Burst time)and click on Add Process button for example here we added 3 processes P1(1,1,20),P2(2,12,10) and P3(3,10,12)



- After adding processes click on calculate to run the algorithm. Given below is the final table formed after adding all processes.



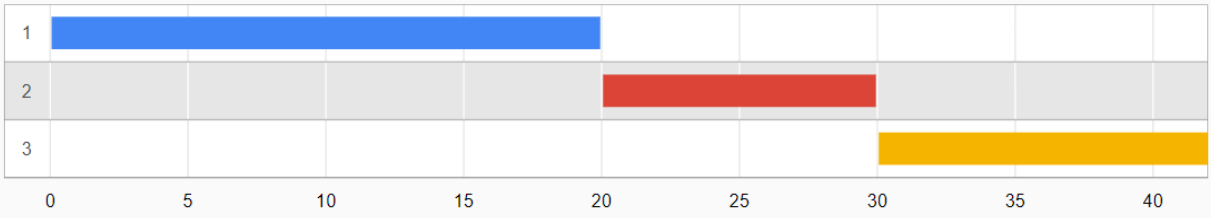
- After clicking on Calculate the Output table and Gantt chart appears. For the above inputs Conder FCFS That is first process executes first, Given below is the Output table that appears for the given example. This table includes input details as well as waiting time and turn-around time for the processes.

Output Table

Process ID	Arrival Time	Burst Time	Waiting Time	Turn Around Time
1	1	20	0	20
2	12	10	20	30
3	10	12	30	42

- Conder FCFS That is the first process executes first, So theFirst process P1 executes first, followed by P2 and then finally P3. Given below is the gantt chart that gives a visual representation of scheduled processes. Here first process run for first 20 seconds then form 20th to 30th second P2 was being processed and final 12 seconds process P3 was executed.

Gantt Chart



CONCLUSION

A lightweight and aesthetically appealing React Js based web application has been developed for all the OS enthusiasts, providing functionality of visualising three operating system's scheduling algorithms - First Come First Serve, Round Robin and Shortest Job First, through various dynamic tables (depicting Precisely calculated output values of CPU scheduling algorithms) and gantt charts (depicting output in an aesthetic and interactive way) , in a more easier, elaborative and comprehensive way, on user defined scenarios, in real time. Afterwards, Successfully executed and tested the React js web Application for various platforms (Desktop, Laptop) have been deployed on netlify.

IMPORTANT LINKS

1. The SAC
2. Github Repository

Possible future works on next page

FUTURE WORK

- ***Adding more algorithms***

  There are many CPU Scheduling algorithms apart from the ones added in this project, so as a part of future work we can include more algorithms..

- ***Comparison between different process scheduling algorithms may be included***

  One can compare two or more algorithms for better understanding.

- ***Animation and GUI***

  Visualisation can include animations to provide a better user experience and interactive way of learning algorithms. Sequential animation of processes being processed in Gantt chart or animation of working algorithms.

- ***Discussion Box/Page***

  Live responses or queries can be accepted and posted from users/ students to make this environment more interactive.