OPERATING SYSTEM

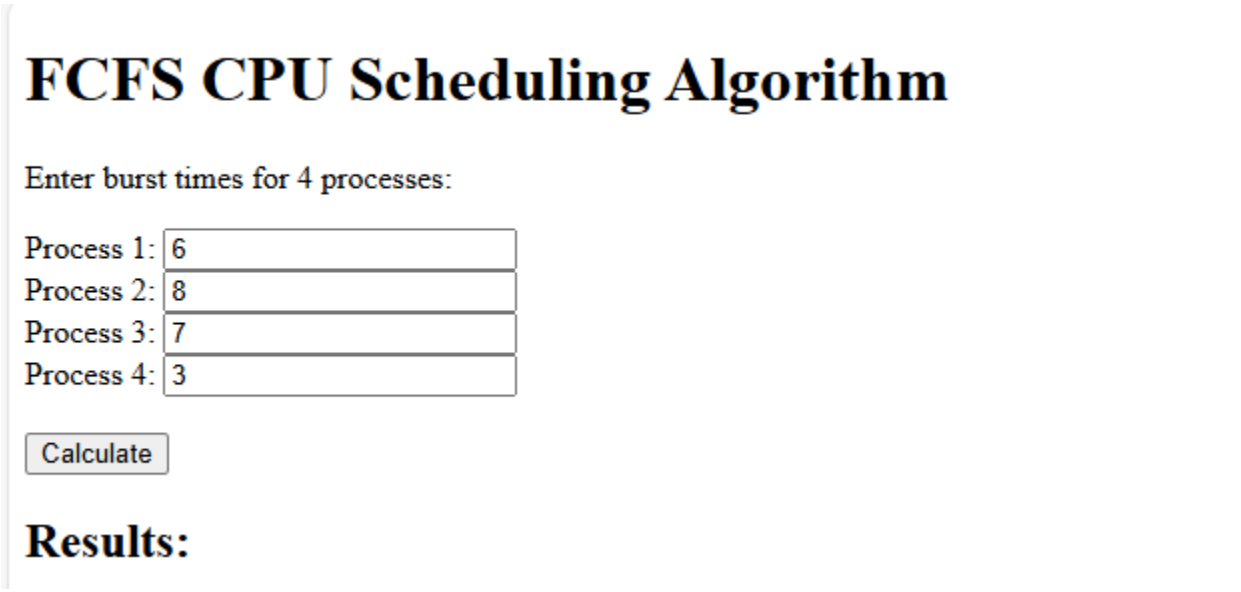
HANAN FATHIMA NS
S3 BCA

Question Task: Illustrate any CPU Scheduling Algorithm (FCFS, SJF, or Round Robin) using any tool of your choice (Python, C, Excel, Google Sheets, Java, JavaScript, or even a diagram tool). In addition, demonstrate your understanding of Git Collaboration Workflow as follows:

Using HTML and JAVASCRIPT
CODE

```
<html>
<head>
  <title>Simple FCFS Scheduling</title>
</head>
<body>
  <h1>FCFS CPU Scheduling Algorithm (Simple)</h1>

  <p>Enter burst times for 4 processes:</p>

  <label>Process 1: <input type="number" id="bt1" value="6" min="1"></label><br>
  <label>Process 2: <input type="number" id="bt2" value="8" min="1"></label><br>
  <label>Process 3: <input type="number" id="bt3" value="7" min="1"></label><br>
  <label>Process 4: <input type="number" id="bt4" value="3" min="1"></label><br><br>

  <button onclick="calculateFCFS()">Calculate</button>

  <h2>Results:</h2>
  <pre id="output"></pre>

  <script>
    function calculateFCFS() {
      let burstTimes = [];
      burstTimes.push(parseInt(document.getElementById('bt1').value));
      burstTimes.push(parseInt(document.getElementById('bt2').value));
      burstTimes.push(parseInt(document.getElementById('bt3').value));
      burstTimes.push(parseInt(document.getElementById('bt4').value));

      let waitingTime = [0, 0, 0, 0];
      let turnaroundTime = [0, 0, 0, 0];
      for (let i = 1; i < 4; i++) {
        waitingTime[i] = waitingTime[i - 1] + burstTimes[i - 1];
      }
      for (let i = 0; i < 4; i++) {
        turnaroundTime[i] = waitingTime[i] + burstTimes[i];
```

```
    }
    let result = "Process\tBurst Time\tWaiting Time\tTurnaround Time\n";
    for (let i = 0; i < 4; i++) {
      result += `P${i+1}\t\t${burstTimes[i]}\t\t${waitingTime[i]}\t\t${turnaroundTime[i]}\n`;
    }
    document.getElementById('output').textContent = result;
  }
 </script>
</body>
</html>
```



FCFS CPU Scheduling Algorithm

Enter burst times for 4 processes:

Process 1: 6
Process 2: 8
Process 3: 7
Process 4: 3

Calculate

Results:

Process	Burst Time	Waiting Time	Turnaround Time
P1	6	0	6
P2	8	6	14
P3	7	14	21
P4	3	21	24

FCFS CPU Scheduling Algorithm

Enter burst times for 4 processes:

Process 1: 8
Process 2: 9
Process 3: 10
Process 4: 1

Calculate

Results:

Process	Burst Time	Waiting Time	Turnaround Time
P1	8	0	8
P2	9	8	17
P3	10	17	27
P4	1	27	28