Year:-	Academic Year- 2025-26	Semester:-

Experiment # 3

Roll No:D084	Name: Somish Jain
Class:Btech ce	Batch: 2
Date of Experiment:4/8/25	Date of Submission:4/8/25

Code

```
import matplotlib.pyplot as plt
import requests

class Task:
    def __init__(self, id, arrival_time, duration):
        self.id = id
        self.arrival_time = arrival_time
        self.duration = duration
        self.start = 0
        self.finish = 0
        self.waiting = 0
        self.turnaround = 0
        self.response = 0

print("Retrieving job data from external source...")
url = "https://jsonplaceholder.typicode.com/todos"

try:
    api_response = requests.get(url, timeout=5)
    api_response.raise_for_status()
    job_data = api_response.json()
    print("Job data successfully loaded.\n")
except requests.exceptions.RequestException as err:
    print("Failed to load data:", err)
    exit(1)

jobs = []
for i, entry in enumerate(job_data[:10]):
    task_id = f"T{i + 1}"
    arrival_time = i * 2
    execution_time = (len(entry['title']) % 15) + 3
    jobs.append(Task(task_id, arrival_time, execution_time))
```

```
jobs.sort(key=lambda x: x.arrival_time)

current_time = 0
sum_wait = sum_turnaround = sum_response = 0

for job in jobs:
    job.start = max(current_time, job.arrival_time)
    job.finish = job.start + job.duration
    job.turnaround = job.finish - job.arrival_time
    job.waiting = job.turnaround - job.duration
    job.response = job.start - job.arrival_time

    current_time = job.finish
    sum_wait += job.waiting
    sum_turnaround += job.turnaround
    sum_response += job.response

print("\n==== FCFS Scheduling Summary (Fetched from API) ====")
print(f"{'ID':<5}{'Arrival':<8}{'Burst':<6}{'Start':<8}{'End':<6}"
      f"{'TAT':<6}{'Wait':<6}{'Response':<9}")

for job in jobs:
    print(f"{job.id:<5}{job.arrival_time:<8}{job.duration:<6}"
          f"{job.start:<8}{job.finish:<6}{job.turnaround:<6}"
          f"{job.waiting:<6}{job.response:<9}")

num_jobs = len(jobs)
print(f"\nAverage Waiting Time   : {sum_wait / num_jobs:.2f}")
print(f"Average Turnaround Time: {sum_turnaround / num_jobs:.2f}")
print(f"Average Response Time  : {sum_response / num_jobs:.2f}")

# Gantt Chart
print("\nGenerating Gantt chart for job execution timeline...")
fig, ax = plt.subplots(figsize=(10, 5))
ax.set_title("Gantt Chart with API by Somish Jain")
ax.set_xlabel("Time")
ax.set_yticks([15 + 10 * i for i in range(num_jobs)])
ax.set_yticklabels([job.id for job in jobs])
ax.set_xlim(0, current_time + 5)
ax.set_ylim(0, 10 * (num_jobs + 1))

for idx, job in enumerate(jobs):
    ax.broken_barh([(job.start, job.duration)], (10 * idx + 10, 8), facecolors='Red')

plt.grid(True)
```

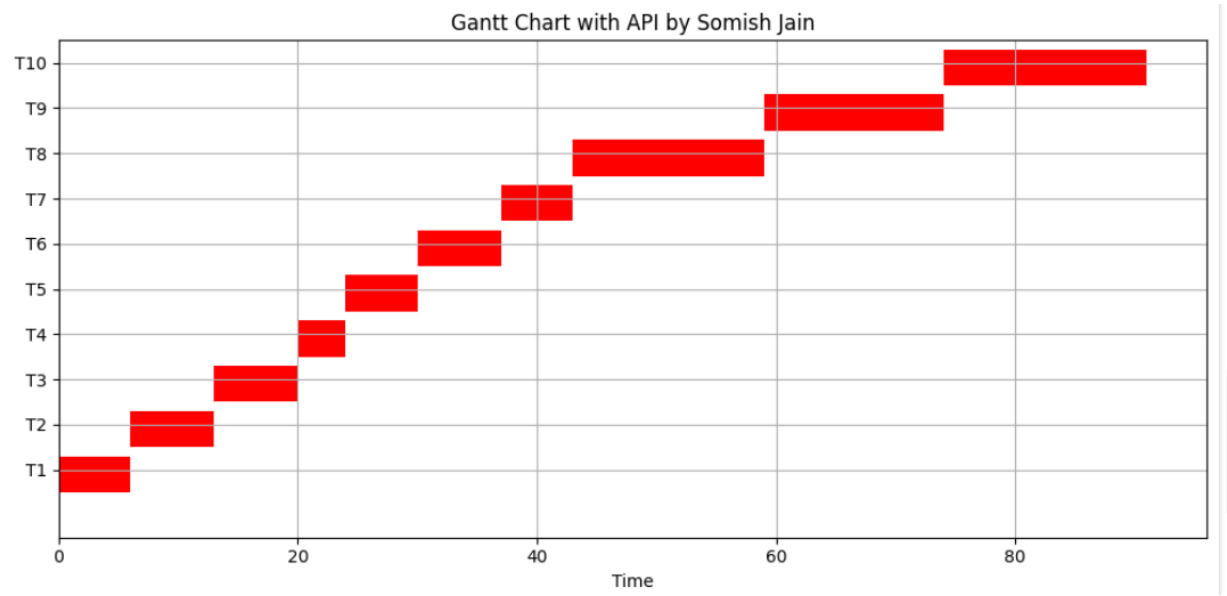
```
plt.tight_layout()
plt.show()
```

```
Retrieving job data from external source...
Job data successfully loaded.


==== FCFS Scheduling Summary (Fetched from API) ====
ID   Arrival Burst Start   End   TAT   Wait  Response
T1   0       6     0       6     6     0     0
T2   2       7     6       13    11    4     4
T3   4       7     13      20    16    9     9
T4   6       4     20      24    18    14    14
T5   8       6     24      30    22    16    16
T6   10      7     30      37    27    20    20
T7   12      6     37      43    31    25    25
T8   14      16    43      59    45    29    29
T9   16      15    59      74    58    43    43
T10  18      17    74      91    73    56    56

Average Waiting Time   : 21.60
Average Turnaround Time: 30.70
Average Response Time  : 21.60

Generating Gantt chart for job execution timeline...
```



Year:-	Academic Year- 2025-26	Semester:-

1. How does FCFS handle simultaneous arrival of processes?
In the FCFS (First-Come, First-Served) scheduling method, if multiple processes arrive at the same moment, they are scheduled based on the order in which they enter the queue. The one that is placed first in the queue is executed first, regardless of their burst times or priorities.

---

2. What is the difference between waiting time and response time Waiting Time is the total duration a process spends in the ready queue before it begins execution (excluding the execution time itself).

- Response Time is the time taken from the arrival of a process until it starts executing for the first time.
  In non-preemptive scheduling like FCFS, these two times are often the same since the process runs once it reaches the CPU.

---

3. Why is FCFS considered non-preemptive?
FCFS is non-preemptive because once a process starts using the CPU, it continues until it finishes its entire burst time. The CPU doesn't switch to another process, even if a new one with a shorter burst time arrives.

---

4. What impact does process burst time have on average waiting time in FCFS?
In FCFS scheduling, a process with a long burst time at the beginning can delay all the following processes, increasing their waiting time. This can significantly raise the overall average waiting time, making the scheduling less efficient.

---

5. How can you simulate scheduling behavior using real trace data?
To simulate scheduling, real or sample trace data containing process arrival and burst times can be used. Scheduling algorithms like FCFS can then be applied to this data to calculate performance metrics such as waiting time or turnaround time. Tools like Gantt charts can help visualize the scheduling sequence.

---

6. Why might response time be important in interactive systems?
In interactive applications, such as user interfaces or web services, quick responsiveness is key to user satisfaction.