Index

EXPERIMENT-1

Aim:
To elucidate the activity schedule of the project
Theory:
In software project management, an activity refers to a distinct task or piece of work that must be accomplished as part of the overall project plan. Activities are individual units of work that contribute to the successful completion of the project. They are usually defined, scheduled, assigned to team members, and tracked to ensure they are completed on time and within the project's constraints. When constructing a PERT chart, the following time spans are considered:

- Duration of the activity: The time required to complete a specific task or activity.
- Earliest start: The soonest a task can begin, based on dependencies and the project schedule.
- Earliest finish: The earliest possible time a task can be completed, considering its earliest start and duration.
- Latest start: The latest time a task can begin without affecting the project's overall timeline.
- Latest finish: The latest time a task can be completed without causing delays in the project.
- Float (Slack time): The amount of time a task can be delayed without impacting the project's overall duration.

Result:

ACTIVITY LABEL	**DURATION**	**EARLIEST START**	**EARLIEST FINISH**	**LATEST START**	**LATEST FINISH**	**FLOAT**
Requirement and Feasibility Analysis	4	1	4	1	4	0
SRS Design	3	4	7	4	7	0
Literature Review	2	7	9	7	9	0
Frontend code	1	9	10	10	11	0
Backend code	2	9	11	9	11	0
Testing	2	11	13	11	13	0
Deployment	2	13	15	13	15	0
Report Writing	3	15	18	15	18	0

EXPERIMENT-2

Aim:

To draw network graph and find the critical path using draw.io (CPM)

Theory:

A Network Diagram is a graphical representation of a project's tasks and activities along with their dependencies. It helps project managers visualize the sequence in which tasks must be completed and the relationships between them. In software project management, the network diagram serves as a roadmap, showing the order of activities, their duration, and their dependency on other activities.

Key Elements of a Network Diagram:

- Nodes (Events/Activities): Represent the milestones or activities in the project. Each node shows a specific task, like coding, testing, or deployment.
- Arrows (Dependencies): Arrows represent the dependencies or precedence relationships between tasks. If Task A must be completed before Task B can begin, an arrow connects A to B.
- Critical Path: The sequence of tasks that determines the shortest time in which the project can be completed (explained in CPM below).
- Start and End Nodes: Indicate the beginning and the completion of the entire project.

Types of Network Diagrams:

- Activity-on-Node (AON): Tasks are represented by nodes, and arrows depict the dependencies.
- Activity-on-Arrow (AOA): Arrows represent tasks, and nodes show the starting or ending points of these tasks.

Benefits of Network Diagrams in Software Management:

- Visual Clarity: It provides a clear picture of the project’s activities, sequence, and dependencies.
- Dependency Tracking: Ensures that tasks are done in the correct order, reducing bottlenecks.
- Resource Allocation: Helps in scheduling and allocating resources based on task sequencing.
- Time Management: Helps identify the critical path, which is essential for timely project completion.

The Critical Path Method (CPM) is a project scheduling technique used to identify the most important sequence of activities that determine the minimum project completion time. In software development, CPM is used to plan, schedule, and control the project’s activities and ensure timely delivery.

Key Concepts of CPM:

- Critical Path: The longest sequence of dependent activities that must be completed for the project to finish on time. Any delay in tasks on the critical path will delay the entire project.
- Activities on the critical path have zero float (slack time), meaning they cannot be delayed without affecting the overall project timeline.
- Float (Slack): The amount of time an activity can be delayed without affecting the project's completion time. Non-critical activities often have float, giving project managers flexibility.
- Earliest Start Time (ES): The earliest possible time an activity can begin, considering the completion of preceding tasks.
- Earliest Finish Time (EF): The earliest an activity can finish, calculated as the ES plus the activity duration.
- Latest Start Time (LS): The latest time an activity can start without delaying the project’s overall completion.

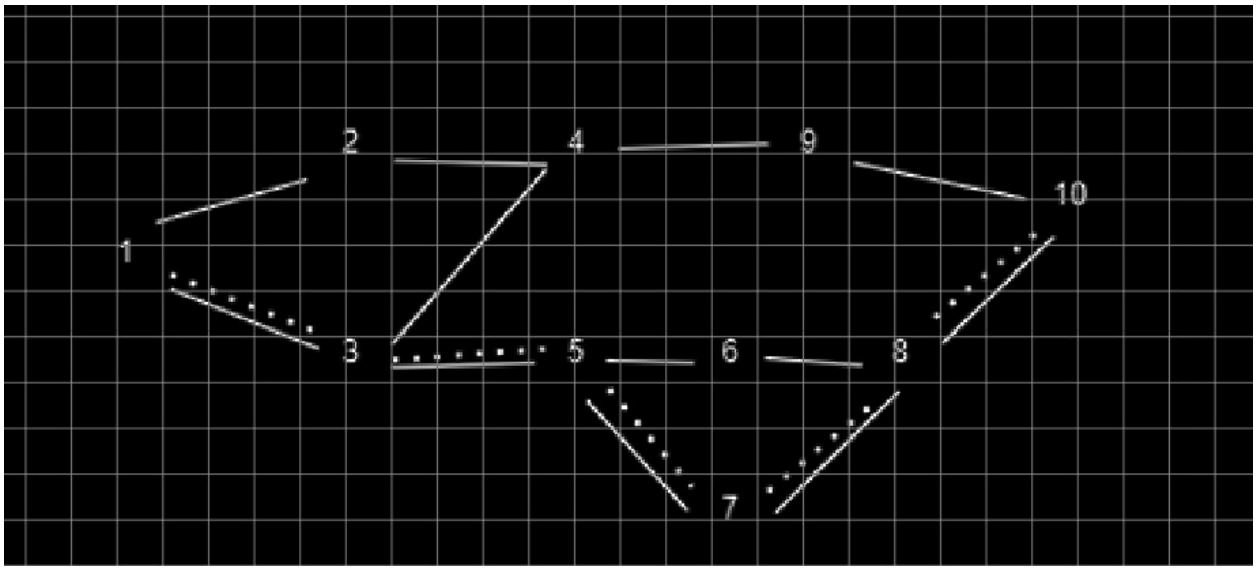
- Latest Finish Time (LF): The latest an activity can finish without affecting the project end date.

Given the activity schedule of a project:

Activity	Time(days)
1-2	4
1-3	1
2-4	1
3-4	1
3-5	6
4-9	5
5-6	4
5-7	8
6-8	1
7-8	2
8-10	5
9-10	7

Result:

The figure below shows the network diagram of the above activity schedule and the dotted lines depict the critical path.



Critical path: 1->3->5->7->8->10

EXPERIMENT-3

Aim:
Design the GANTT chart for the project

Theory:

A Gantt chart is a visual project management tool used to illustrate a project's schedule, showing tasks, their durations, dependencies, and progress. In software project management, Gantt charts are invaluable for planning, coordinating, and tracking various phases of software development.

Key Features of a Gantt Chart:

- **Horizontal Bars**: Represent tasks or activities and their durations over time.
- **Time Axis**: Usually at the top of the chart, showing the timeline (days, weeks, or months) across which the project tasks are distributed.
- **Tasks/Activities**: Listed vertically, representing different phases or components of the project (e.g., requirements gathering, design, coding, testing).
- **Dependencies**: Arrows or lines showing relationships between tasks (e.g., Task B cannot start until Task A is complete).
- **Milestones**: Key events or deadlines, represented as symbols (usually diamonds), marking significant points in the project timeline.
- **Progress Tracking**: Filled portions of bars represent how much of a task has been completed, providing real-time insight into project progress.

Importance of Gantt Charts in Software Project Management:

- **Visual Overview**: Offers a clear and detailed view of the project timeline, helping team members understand the flow of tasks and deadlines.
- **Task Dependencies**: Easily highlights which tasks depend on others, helping to manage task sequences and avoid bottlenecks.
- **Resource Allocation**: Helps in assigning resources efficiently, as you can visually see which team members or departments are responsible for each task.
- **Time Management**: Allows project managers to track the progress of tasks and adjust schedules as needed, ensuring on-time delivery of the project.

Important Components of a Gantt Chart**:**

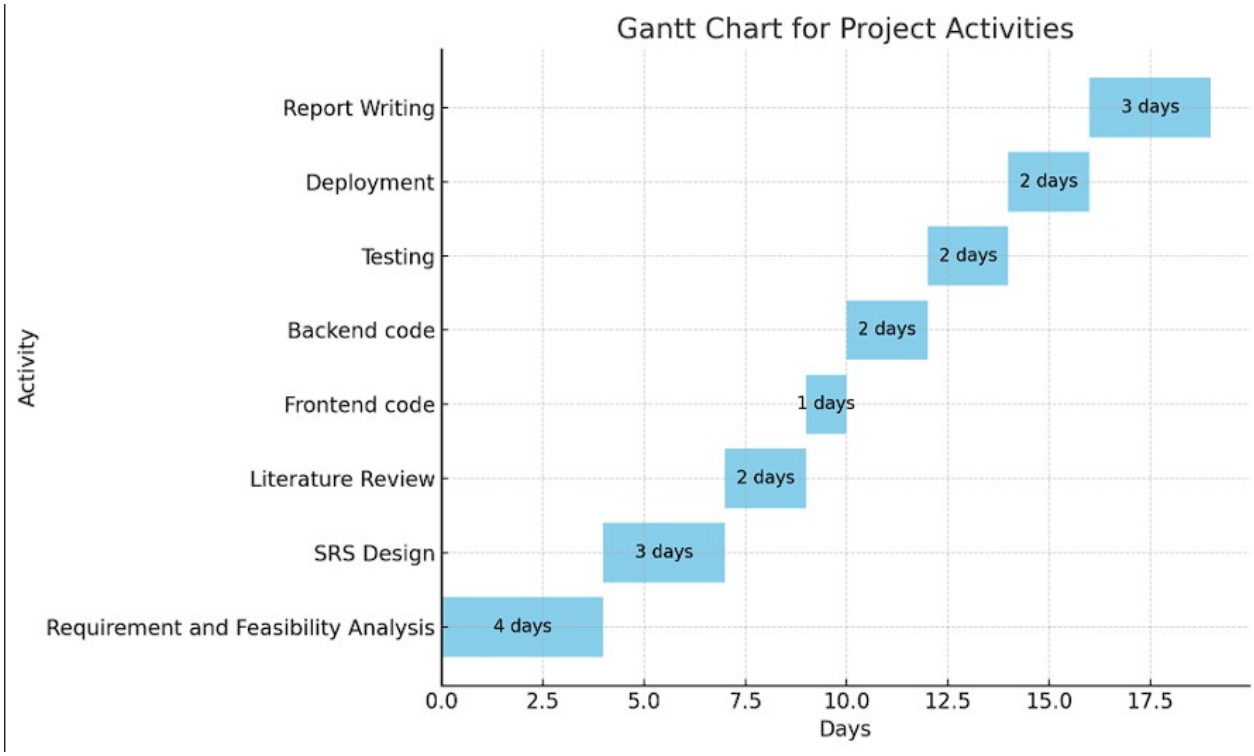
1. **Tasks/Activities**:
    - These are the individual pieces of work that need to be completed for the project. Each task is represented by a horizontal bar, and the length of the bar corresponds to the duration of the task.
    - In software management, tasks could include phases like **requirement analysis**, **design**, **coding**, **testing**, and **deployment**.
2. **Time Axis**:
    - The time axis, usually horizontal across the top of the Gantt chart, represents the project timeline. It can be divided into days, weeks, or months depending on the project's duration.
    - This axis helps track when each task starts and finishes, providing a clear view of deadlines.
3. **Duration**:

    - The duration of each task is represented by the length of the bar. Longer bars indicate tasks that take more time.
    - For example, **coding** might take 3 weeks, while **testing** could take 2 weeks.
4. **Dependencies**:
    - Dependencies show the relationships between tasks. Some tasks can't begin until others are finished (e.g., **frontend development** might depend on **backend setup**).
    - These dependencies are represented by arrows or connectors between tasks, making it clear which activities need to be completed before others can start.
5. **Milestones**:
    - **Milestones** are significant points in the project, such as the completion of a major phase or delivery of a key feature.
    - In software projects, milestones might include **beta testing complete**, **product launch**, or **client approval**.
    - Milestones are often shown as diamond shapes and don't have duration—they just mark important events or deadlines.
6. **Task Progress**:
    - A Gantt chart often shows how much progress has been made on each task. This is indicated by partially filled bars, where the filled portion reflects the percentage of the task that has been completed.
    - This feature is critical for tracking the actual progress of the project against the planned timeline.
7. **Resources**:
    - Some Gantt charts also include a column for resources, which shows which team members, tools, or departments are assigned to each task.
    - This helps in managing workload and ensuring the right resources are allocated to critical tasks at the right time.
8. **Critical Path**:
    - While not always explicit in basic Gantt charts, many modern Gantt chart tools can highlight the **critical path**, which is the sequence of tasks that directly affects the project's completion date.
    - Tasks on the critical path have zero float, meaning any delay will directly delay the project completion.
9. **Slack/Float**:
    - Some tasks may have **float** or **slack**, which means they can be delayed without affecting the overall project timeline. This is shown visually, often by a lighter section of the task bar extending beyond the planned finish date.
    - Float helps in understanding where there is flexibility in the project and which tasks need strict management.

Given the following project phases and their respective duration:

Activty	Time(in days)
Requirement and Feasibility Analysis	4
SRS Design	3
Literature Review	2
Frontend code	1
Backend code	2
Testing	2
Deployment	2
Report Writing	3

Result:


Gantt Chart for Project Activities
Activity
Report Writing
Deployment
Testing
Backend code
Frontend code
Literature Review
SRS Design
Requirement and Feasibility Analysis
3 days
2 days
2 days
2 days
1 days
2 days
3 days
4 days
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Days

EXPERIMENT-4

Aim:
To draw the PERT chart for a project

Theory:
The PERT chart is a flowchart-like diagram that visually represents the sequence of activities (tasks) required to complete a project, along with the time estimates for each activity. It helps in determining the minimum time needed to complete the project and identifies the critical path—tasks that must be done on time to avoid delaying the overall project.

Key Concepts of PERT Chart:

- Events: Milestones that signify the start or end of one or more activities. Events don't consume time but represent key points in the project timeline.
- Activities: Tasks that need to be completed. Each activity has an associated duration estimate and represents the actual work that takes time to complete.
- Dependencies: The relationships between activities that indicate which activities must be completed before others can begin.
- Critical Path: The longest path of dependent activities from start to finish that defines the shortest time possible to complete the project.

Time Estimates:
Each activity in PERT is given three time estimates:

- Optimistic Time (O): The shortest time an activity will take if everything goes perfectly.
- Pessimistic Time (P): The longest time an activity will take if everything goes wrong.
- Most Likely Time (M): The best guess at how long the activity will take based on normal circumstances.
- These estimates are used to calculate the Expected Time (TE) using the formula:

$$TE = \frac{O + 4M + P}{6}$$

Given the following activity schedule with the predecessors and durations:

Activity	Predecessor	t0	tm	tp
A	-	2	4	9
B	A	5	8	14
C	B	4	10	13
D	B	4	7	10
E	C	11	14	20
F	D	9	13	16
G	E, F	2	4	6

Result:

Activity	Predecessor	t0	tm	tp	Exp time	variance	S.D.
A	-	2	4	9	4.5	1.361	1.166
B	A	5	8	14	8.5	2.25	1.5
C	B	4	10	13	9.5	2.25	1.5
D	B	4	7	10	7	1	1
E	C	11	14	20	14.5	2.25	1.5
F	D	9	13	16	12.83	1.361	1.166
G	E, F	2	4	6	4	0.445	0.667

EXPERIMENT-5

Aim:
To make SLIP chart, Timeline Chart and Ball Chart

Theory:

A **SLIP Chart** (Schedule, Load, Investment, and Performance) is a visual representation used in project management to track and analyze the progress and performance of a project. It focuses on four main aspects of a project: Schedule, Load, Investment, and Performance.

Components:

Schedule (S): Represents the planned timeline for project activities, showing when tasks are scheduled to start and finish.
Load (L): Indicates the resources (human and material) allocated to each task, helping to visualize resource distribution.
Investment (I): Represents the financial aspects of the project, including budgeted versus actual costs. It helps track how financial resources are being utilized.
Performance (P): Shows key performance indicators (KPIs) and metrics related to project outputs, such as quality and efficiency.

A **Timeline Chart** is a graphical representation of a project's schedule, illustrating the sequence of tasks and their durations over a specified time period. It presents a clear view of project milestones and deadlines.

Components:

Tasks/Activities: Each task or activity is represented along the timeline, often displayed as horizontal bars.
Time Axis: The horizontal axis represents time, divided into suitable intervals (days, weeks, months).
Milestones: Significant events or goals in the project are marked along the timeline, indicating critical points of progress.

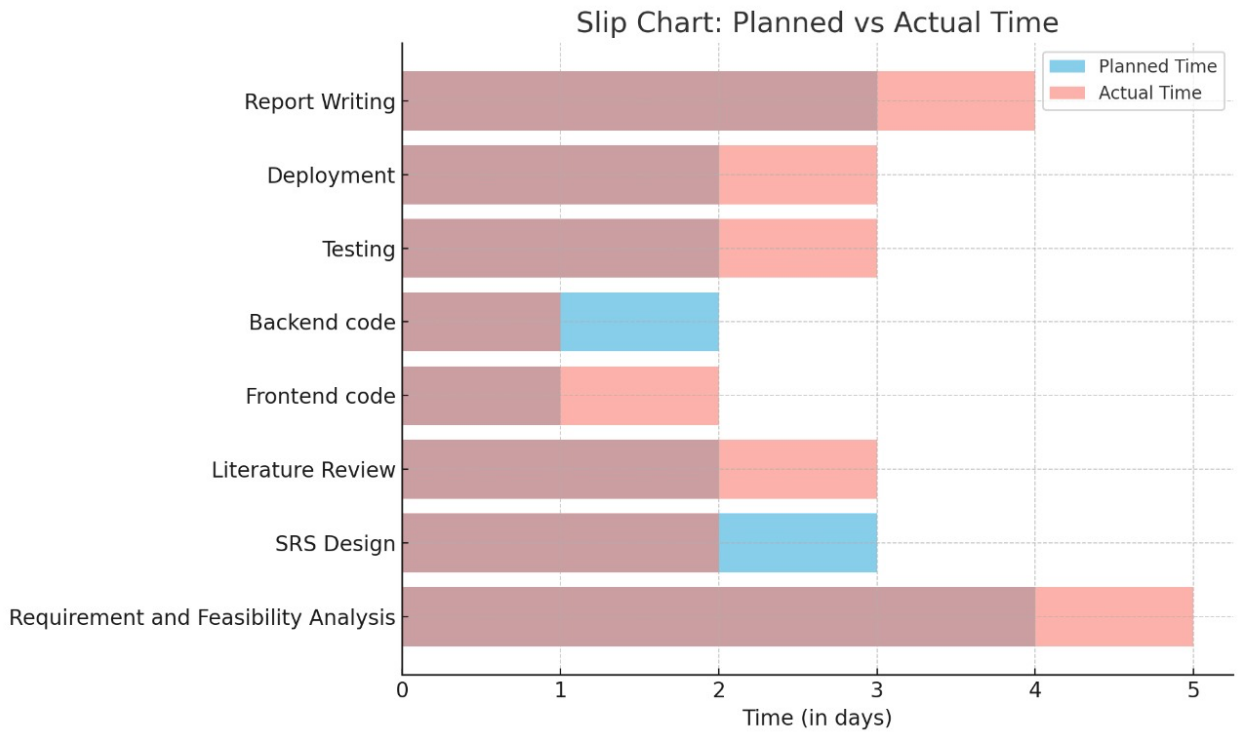
A **Ball Chart** is a visual tool used in project management to illustrate the status of project tasks and resources in a simplified manner. It displays tasks as colored balls on a grid, representing their progress and resource allocation.

Components:

Balls/Circles: Each ball represents a task or activity. The color or size of the ball may indicate its status (e.g., completed, in progress, delayed).
Axes: The horizontal axis often represents time, while the vertical axis represents the level of effort, resources, or priority.
Legends/Keys: Used to decode the meaning of different colors or sizes of the balls.

Result:

SLIP chart:



Ball Chart: