

# Scheduling

2025-2026

## A. Objective

Creating, analyzing, and optimizing a project schedule.

## B.Scenario: Project "Wishlist"

You are the new project manager for a small e-commerce company. Your team is tasked with developing a new "Wishlist" feature for the website. You have three developers, each with a specialty:

- **Salah (Backend)**
- **Ahmed (Frontend)**
- **Zyed (DB/QA)**

### Project Brief: Task List (WBS)

Here is the high-level task list your team has compiled.

| Task ID | Task Name            | Duration (Days) | Resource(s)         | Predecessor(s) |
|---------|----------------------|-----------------|---------------------|----------------|
| A       | Kickoff Meeting      | 1               | Salah, Ahmed        | -              |
| B       | Define Feature Specs | 2               | Salah, Ahmed , Zyed | A              |
| C       | Build API            | 4               | Salah               | B              |
| D       | Implement DB Schema  | 3               | Zyed                | B              |
| E       | Design UI Mockups    | 3               | Ahmed               | B              |

| <b>Task ID</b> | <b>Task Name</b>                           | <b>Duration (Days)</b> | <b>Resource(s)</b> | <b>Predecessor(s)</b> |
|----------------|--------------------------------------------|------------------------|--------------------|-----------------------|
| <b>F</b>       | <i>Build UI Components</i>                 | 5                      | Ahmed              | E                     |
| <b>G</b>       | <i>Write Test Cases</i>                    | 3                      | Zyed               | B                     |
| <b>H</b>       | <i>Integrate API &amp; DB</i>              | 2                      | Salah, Zyed        | C, D                  |
| <b>I</b>       | <i>Final Integration &amp; System Test</i> | 3                      | Ahmed, Zyed        | F, G, H               |
| <b>J</b>       | <i>Deploy to Production</i>                | 1                      | Salah              | I                     |

## C. Activity Steps

### 1. Initial Network Diagram & Critical Path

First, let's map out the project as-is.

- Draw the Diagram:** Using an online whiteboard, create an **network diagram** based on the task list. For nodes use a six-compartment box.
- Calculate Schedule:** Perform a **Forward Pass** and **Backward Pass** to determine the Early Start (ES), Early Finish (EF), Late Start (LS), and Late Finish (LF) for every task.
- Find the Critical Path:**
  - What is the total project duration?
  - Identify the critical path (or paths) by finding the sequence of tasks with **zero slack** (Slack = LS - ES).

### 2. What-If Analysis - Leads & Lags

Modify your diagram from Part 1 to reflect these new scenarios.

- **Scenario 1 (Lag):** The team decides that **Task G (Write Test Cases)** can only begin after the UI mockups are finished and approved. This adds a **1-day lag** after **Task E**. The new dependency for G is E (FS+1d).
- **Scenario 2 (Lead/Fast-Tracking):** To speed things up, **Task F (Build UI Components)** can start when **Task E (Design UI Mockups)** is 2/3 complete. Task E takes 3 days, so Task F can start 1 day before E finishes. This is a **Finish-to-Start with a 1-day lead** (FS-1d).

**Task:** Re-calculate the ES, EF, LS, LF, and the critical path with these new dependencies.

- Has the total project duration changed?
- Has the critical path changed?

### 3. Resource Visualization

- a. **Create Gantt chart:** Draw a Gantt-style chart, but create three horizontal "swimlanes," one for each team member: **Salah, Ahmed** , and **Zyed**. (Tasks with multiple resources should be duplicated: Task T becomes Task Ta and Tb with a start to start constraint).



- b. **Plot the Schedule:** Using your (updated) schedule from Part 2, draw each task as a bar on its **Earliest Start (ES)** date in the correct resource's swimlane.
- c. **Identify Conflicts:** Examine your swimlane chart. Do you see any days where a single resource (e.g., Zyed) is assigned to two or more tasks simultaneously?

### 4. Resource Leveling

#### 1. Analyze the Conflict:

- Which resource is over-allocated?
- Which tasks are in conflict?
- Look at the **slack** for the conflicting tasks.

2. **Propose a Solution:** Apply **resource leveling**. The easiest way is to delay the start of one of the conflicting tasks until the other is finished.
  - Which task should you delay? **Hint:** Always try to delay the task with the **most slack** first, as it's less likely to affect the critical path.
3. **Create a New Schedule:** Implement your solution by adding a new precedence (e.g., Task D must finish before Task G can start).
4. **Re-compute:** Recalculate the ES, EF, LS, LF, and slack for all tasks based on this new, leveled schedule.
5. **Final Analysis & Report**

Conclude the activity by analyzing the results of your leveling.

1. **Final Critical Path:** What is the **new** critical path and the **new** final project duration?
2. **Impact:** Did leveling the resources extend the project? Why or why not?
3. **What If...:** What if your leveling did extend the project, but management insists on the original deadline? Based on your lecture, name one **schedule compression** technique (like crashing or fast-tracking) you could apply to the new critical path to shorten it.

## D. Submission

### Part 1: Initial Network Diagram & Critical Path

Schedule Calculation Table

| Task ID | Task Name            | Duration (Days) | Resource(s)        | Predecessor(s) | ES | EF | LS | LF | Slack | Critical Path? |
|---------|----------------------|-----------------|--------------------|----------------|----|----|----|----|-------|----------------|
| A       | Kickoff Meeting      | 1               | Salah, Ahmed       | -              | 0  | 1  | 0  | 1  | 0     | Yes            |
| B       | Define Feature Specs | 2               | Salah, Ahmed, Zyed | A              | 1  | 3  | 1  | 3  | 0     | Yes            |
| C       | Build API            | 4               | Salah              | B              | 3  | 7  | 5  | 9  | 2     | No             |
| D       | Implement DB Schema  | 3               | Zyed               | B              | 3  | 6  | 6  | 9  | 3     | No             |
| E       | Design UI Mockups    | 3               | Ahmed              | B              | 3  | 6  | 3  | 6  | 0     | Yes            |

|          |                                 |   |             |         |    |    |    |    |   |            |
|----------|---------------------------------|---|-------------|---------|----|----|----|----|---|------------|
| <b>F</b> | Build UI Components             | 5 | Ahmed       | E       | 6  | 11 | 6  | 11 | 0 | <b>Yes</b> |
| <b>G</b> | Write Test Cases                | 3 | Zyed        | B       | 3  | 6  | 8  | 11 | 5 | No         |
| <b>H</b> | Integrate API & DB              | 2 | Salah, Zyed | C, D    | 7  | 9  | 9  | 11 | 2 | No         |
| <b>I</b> | Final Integration & System Test | 3 | Ahmed, Zyed | F, G, H | 11 | 14 | 11 | 14 | 0 | <b>Yes</b> |
| <b>J</b> | Deploy to Production            | 1 | Salah       | I       | 14 | 15 | 14 | 15 | 0 | <b>Yes</b> |

### Results

- **Total Project Duration:** 15 Days
- **Critical Path:** A → B → E → F → I → J

### Part 2: What-If Analysis – Leads & Lags

#### Schedule Calculation Table (With Leads & Lags)

| <b>Task ID</b> | <b>Duration</b> | <b>Predecessor(s) / Constraints</b> | <b>ES</b> | <b>EF</b> | <b>LS</b> | <b>LF</b> | <b>Slack</b> | <b>Critical Path?</b> |
|----------------|-----------------|-------------------------------------|-----------|-----------|-----------|-----------|--------------|-----------------------|
| <b>A</b>       | 1               | -                                   | 0         | 1         | 0         | 1         | 0            | <b>Yes</b>            |
| <b>B</b>       | 2               | A                                   | 1         | 3         | 1         | 3         | 0            | <b>Yes</b>            |
| <b>C</b>       | 4               | B                                   | 3         | 7         | 4         | 8         | 1            | No                    |
| <b>D</b>       | 3               | B                                   | 3         | 6         | 5         | 8         | 2            | No                    |
| <b>E</b>       | 3               | B                                   | 3         | 6         | 3         | 6         | 0            | <b>Yes</b>            |
| <b>F</b>       | 5               | E (FS-1d)                           | 5         | 10        | 5         | 10        | 0            | <b>Yes</b>            |
| <b>G</b>       | 3               | E (FS+1d)                           | 7         | 10        | 7         | 10        | 0            | <b>Yes</b>            |
| <b>H</b>       | 2               | C, D                                | 7         | 9         | 8         | 10        | 1            | No                    |
| <b>I</b>       | 3               | F, G, H                             | 10        | 13        | 10        | 13        | 0            | <b>Yes</b>            |
| <b>J</b>       | 1               | I                                   | 13        | 14        | 13        | 14        | 0            | <b>Yes</b>            |

## Analysis

- **Has the total project duration changed?** Yes, the project duration decreased from 15 days to **14 days** due to fast-tracking Task F.
- **Has the critical path changed?** Yes, fast-tracking created **two parallel critical paths** (both lasting 14 days):

1. **A → B → E → F → I → J**

2. **A → B → E → G → I → J**

## Part 3: Resource Visualization

### Resource Swimlane Schedule (Based on Part 2 Earliest Start Dates)

| Resource     | Day 1  | Day 2  | Day 3  | Day 4  | Day 5  | Day 6      | Day 7  | Day 8      | Day 9      | Day 10 | Day 11 | Day 12 | Day 13 | Day 14 |
|--------------|--------|--------|--------|--------|--------|------------|--------|------------|------------|--------|--------|--------|--------|--------|
| <b>Salah</b> | Task A | Task B | Task B | Task C | Task C | Task C     | Task C | Task H     | Task H     | -      | -      | -      | -      | Task J |
| <b>Ahmed</b> | Task A | Task B | Task B | Task E | Task E | Task E / F | Task F | Task F     | Task F     | Task F | Task I | Task I | Task I | -      |
| <b>Zyed</b>  | -      | Task B | Task B | Task D | Task D | Task D     | -      | Task G / H | Task G / H | Task G | Task I | Task I | Task I | -      |

### Identified Conflicts (Over-allocations)

1. **Ahmed (Day 6):** Assigned to both **Task E** (Days 3–6) and **Task F** (Days 5–10) concurrently.
2. **Zyed (Days 8 & 9):** Assigned to both **Task G** (Days 7–10) and **Task H** (Days 7–9) concurrently.

## Part 4: Resource Leveling

### Conflict & Leveling Analysis

- **Over-allocated Resources:** Ahmed and Zyed.
- **Conflicting Tasks:**

- **Ahmed:** Task E vs. Task F. Fast-tracking must be removed so Task F starts after Task E finishes (ES = 6).
- **Zyed:** Task G (Slack = 0) vs. Task H (Slack = 1).
- **Proposed Leveling Solution:** Delay **Task H** because it has 1 day of slack (whereas Task G is critical with 0 slack). Add a new precedence constraint where **Task H must start after Task G completes** ( $G \rightarrow H$ ).

Re-computed Leveled Schedule Table

| Task ID  | Duration | Predecessor(s)  | ES | EF | LS | LF | Slack | Critical Path? |
|----------|----------|-----------------|----|----|----|----|-------|----------------|
| <b>A</b> | 1        | -               | 0  | 1  | 0  | 1  | 0     | <b>Yes</b>     |
| <b>B</b> | 2        | A               | 1  | 3  | 1  | 3  | 0     | <b>Yes</b>     |
| <b>C</b> | 4        | B               | 3  | 7  | 6  | 10 | 3     | No             |
| <b>D</b> | 3        | B               | 3  | 6  | 7  | 10 | 4     | No             |
| <b>E</b> | 3        | B               | 3  | 6  | 3  | 6  | 0     | <b>Yes</b>     |
| <b>F</b> | 5        | E (Standard FS) | 6  | 11 | 7  | 12 | 1     | No             |
| <b>G</b> | 3        | E (FS+1d)       | 7  | 10 | 7  | 10 | 0     | <b>Yes</b>     |
| <b>H</b> | 2        | C, D, <b>G</b>  | 10 | 12 | 10 | 12 | 0     | <b>Yes</b>     |
| <b>I</b> | 3        | F, G, H         | 12 | 15 | 12 | 15 | 0     | <b>Yes</b>     |
| <b>J</b> | 1        | I               | 15 | 16 | 15 | 16 | 0     | <b>Yes</b>     |

## Part 5: Final Analysis & Report

### 1. Final Critical Path & Project Duration:

- **New Critical Path:**  $A \rightarrow B \rightarrow E \rightarrow G \rightarrow H \rightarrow I \rightarrow J$
- **New Project Duration:** 16 Days

### 2. Impact of Leveling:

- Resource leveling **extended the project by 2 days** (from 14 days to 16 days).

- **Reason:** Removing Ahmed's lead delayed Task F, and resolving Zyed's overlap required pushing Task H to start after Task G finished. This pushed the start of Task I from Day 10 to Day 12, directly extending the completion date along the critical path.

### 3. Schedule Compression Technique:

- **Crashing:** If management requires keeping the 14-day deadline without over-allocating resources, the team should apply **Crashing**. This involves adding extra cost or resources to critical path tasks (such as Task E, G, H, or I)—for instance, having Salah support Zyed on Task G/H or bringing in temporary support to shorten task durations back to 14 days.