**PRACTICAL - 7**

**AIM:** Consider a software development project with the following activities:

Activity A: Define Requirements (Duration: 5 days)

Activity B: Design Database (Duration: 8 days)

Activity C: Develop Frontend (Duration: 10 days)

Activity D: Implement Backend (Duration: 12 days)

Activity E: Perform Testing (Duration: 6 days)

Activity F: Deployment (Duration: 8 days)

1. Use the information to construct a Precedence Diagram.

2. Calculate Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF), and Total Float for each activity.

3. Determine the Critical Path.

4. Calculate the total project duration.

**Solution:**

**Team Details:**

| **Sr. No.** | **Name** | **Enrollment No.** |
| --- | --- | --- |
| **Team Leader** | **Sarth Chaudhari** | **202303103510106** |
| **Team Member 1** | **Yash Patel** | **202203103510228** |
| **Team Member 2** | **Gati Shah** | **202203103510261** |
| **Team Member 3** | **Fenil Shilodre** | **202203103510041** |
| **Team Member 4** | **Angat Shah** | **202203103510097** |

## 

## Introduction

## Project scheduling is a fundamental aspect of software project management, ensuring tasks are completed efficiently within defined timeframes.

## The Critical Path Method (CPM) is used to estimate the total project duration and identify activities crucial to timely completion. This involves analyzing project activities, calculating scheduling parameters like Early Start (ES), Early Finish (EF), Late Start (LS), Late Finish (LF) and Total Float. By determining these values, we can accurately schedule the tasks, identify the critical path and ensure the project is completed within the required timeframe.

**Project Title : Airbnb System**

## Project Activities and Dependencies

| **Activity** | **Description** | **Duration (Weeks)** | **Predecessors** |
| --- | --- | --- | --- |
| A | Requirement Analysis | 5 | - |
| B | UI/UX Design | 8 | A |
| C | Backend Development | 10 | - |
| D | Frontend Development | 12 | B |
| E | Payment Gateway Integration | 6 | C |
| F | Testing | 8 | D, E |
| G | Deployment | 4 | F |

### 

### 

## 

## CPM Calculation :

| **Activity** | **ES** | **EF** | **LS** | **LF** | **Total Float** |
| --- | --- | --- | --- | --- | --- |
| A | 0 | 5 | 0 | 5 | 0 |
| B | 5 | 13 | 5 | 13 | 0 |
| C | 0 | 10 | 9 | 19 | 9 |
| D | 13 | 25 | 13 | 25 | 0 |
| E | 10 | 16 | 19 | 25 | 9 |
| F | 25 | 33 | 25 | 33 | 0 |
| G | 33 | 37 | 33 | 37 | 0 |

## Critical Path & Project Duration :

## The Critical Path : A → B → D → F → G

## The Total Project Duration : 37 Weeks

**Project Title : Farfetch E-commerce System**

## Project Activities and Dependencies

| **Activity** | **Description** | **Duration (Weeks)** | **Predecessors** |
| --- | --- | --- | --- |
| A | Requirement Analysis | 6 | - |
| B | Product Catalog Design | 7 | A |
| C | Recommendation Engine Development | 14 | - |
| D | Payment Gateway Integration | 10 | - |
| E | UI/UX Design | 8 | A |
| F | Frontend & Backend Development | 18 | B, C, E |
| G | Testing | 12 | D, F |
| H | Deployment | 5 | G |

### 

## 

## CPM Calculation :

| **Activity** | **ES** | **EF** | **LS** | **LF** | **Total Float** |
| --- | --- | --- | --- | --- | --- |
| A | 0 | 6 | 0 | 6 | 0 |
| B | 6 | 13 | 7 | 14 | 1 |
| C | 0 | 14 | 0 | 14 | 0 |
| D | 0 | 10 | 22 | 32 | 22 |
| E | 6 | 14 | 6 | 14 | 0 |
| F | 14 | 32 | 14 | 32 | 0 |
| G | 32 | 44 | 32 | 44 | 0 |
| H | 44 | 49 | 44 | 49 | 0 |

## Critical Path & Project Duration :

## The Critical Path : A → E → F → G → H

## The Total Project Duration : 49 Weeks

## Conclusion

## CPM effectively determines project scheduling and helps identify critical tasks that require close monitoring.

* **Farfetch requires a longer duration (49 weeks)** due to its complexity, multiple integrations and parallel dependencies, while **Airbnb’s timeline is shorter (37 weeks)** with a more linear critical path.
* Both projects benefit from the flexibility provided by float values in non-critical tasks, allowing schedule adjustments without affecting the critical path.
* Backend development is a significant factor in both projects, and optimizing it could reduce delays.
* Testing and deployment are crucial for both projects, emphasizing their importance in real-world software releases.