Practical No-03

Sample Practical Example Outline

1. Introduction

Project Title:

Movie Recommendation System

Objective:

The objective of this project is to build and deploy a fully functional movie recommendation system within a time frame of 6 months. The system will suggest movies to users based on their preferences using machine learning techniques such as collaborative filtering, content-based filtering, or hybrid approaches.

Importance of Scheduling:

Scheduling plays a critical role in project success. It ensures:

Timely delivery of each module.

Efficient use of available resources.

Clear task allocation and progress monitoring.

Proper management of dependencies between tasks.

2. Task Identification

Work Breakdown Structure (WBS):

To effectively manage the development process, the project has been broken down into the following key phases:

Requirements Gathering:

Understand the user needs, project goals, and technical requirements.

Design:

UI/UX design for the frontend interface.

System architecture and database schema design.

Data Collection & Preprocessing:

Gather movie data (e.g., MovieLens, IMDb, TMDb datasets).

Clean, filter, and prepare the data for use in training algorithms.

Model Development:

Choose appropriate algorithms (Collaborative, Content-Based, Hybrid).

Train and validate the recommendation model.

Frontend Development:

Create a user-friendly interface for users to browse and rate movies.

Backend Integration:

Set up APIs to connect frontend, model, and database.

Testing:

Conduct unit, integration, and user acceptance testing.

Deployment:

Host the system on a cloud platform or local server.

Ensure scalability and security.

Maintenance:

Monitor system performance and apply updates as needed.

Task Dependencies:

Testing cannot begin until the development of the model, backend, and frontend is complete.

Deployment is only possible after successful testing.

Maintenance begins post-deployment.

3. Resource Scheduling

Resources Needed:

Data & Model Development:

3 Machine Learning Engineers

Required tools: Python, Scikit-learn, TensorFlow, Pandas, Jupyter

Frontend Development:

2 UI/UX Developers

Tools: HTML/CSS, JavaScript, React/Vue

Backend Development & Integration:

2 Developers

Tools: Node.js/Django/Flask, REST APIs, SQL/NoSQL databases

Testing:

2 QA Testers

Tools: Selenium, Postman, JMeter

Allocation:

Developers are assigned based on their skill sets:

ML Engineers handle data processing and model training.

UI/UX developers work on designing and implementing the interface.

Backend developers focus on integration and server-side logic.

Testers ensure that the application is bug-free and performs as expected.

4. Creating the Gantt Chart

To visualize the timeline, we use a Gantt chart that shows task durations, dependencies, and start-end dates.

Tasks:

Each task and its sub-tasks are represented in the Gantt chart. Example tasks:

Week 1–2: Requirement Analysis

Week 3–4: UI/UX Design and System Architecture

Week 5–8: Data Collection & Preprocessing

Week 9–12: Model Development

Week 13–16: Frontend and Backend Development

Week 17–20: Integration and Testing

Week 21–22: Deployment

Week 23–24: Final Maintenance and Documentation

Timeline:

6 months, divided into weekly sprints for better control and adaptability.

Dependencies:

Frontend depends on UI/UX design completion.

Model training depends on data preprocessing.

Testing depends on completion of all development phases.

5. Monitoring

Weekly meetings to evaluate task progress and address bottlenecks.

Use project management tools like Jira, Trello, or Asana for progress tracking.

Adjust the schedule if any task is delayed (e.g., unexpected errors in model accuracy or API failures).

6. Conclusion

Outcome:

The Gantt chart and scheduling approach helped streamline the development and deployment process of the movie recommendation system. Each team was aware of their roles and deadlines.

Impact:

Efficient scheduling ensured that the team stayed on track, reduced idle time, and optimized resource utilization.

Key Learning:

A well-planned and monitored schedule significantly increases the chances of delivering a successful and functional system on time.