Lab File

Software Project Management [CSE432]

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted To:
Dr. Vasudha Vashisht
Associate Professor
CSE, ASET

Submitted By:
Utkarsh Dudeja
A2305220145
7CSE1-Y

Department of Computer Science & Engineering
AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
AMITY UNIVERSITY, UTTAR PRADESH
2023

LAB INDEX

Practical-1

Aim: To prepare Gantt Chart for Machine Learning Project.

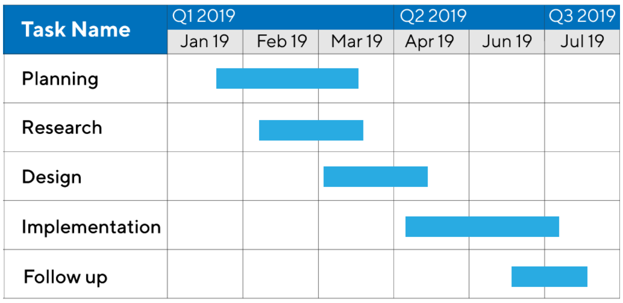
Software used: Microsoft Excel

Introduction to Gantt Chart: A Gantt chart is a visual tool used in project management to represent a project's schedule and tasks over time.

- **Visual Representation**: Gantt charts are graphical representations that display tasks, activities, or events along a horizontal timeline.
- **Time Axis**: The horizontal axis represents time, typically divided into days, weeks, months, or other relevant units.
- **Tasks and Activities**: Each task or activity is represented as a separate horizontal bar on the chart.
- **Task Duration**: The length of the bar corresponds to the duration of the task. Longer bars represent longer tasks.
- **Dependencies**: Gantt charts can show task dependencies, indicating which tasks need to be completed before others can start.
- **Task Relationships**: The chart illustrates the order of tasks and how they relate to each other, helping to plan and manage workflow.
- **Milestones**: Important project milestones, such as project start and completion dates, are often marked on the chart.
- **Resource Allocation**: Gantt charts can show the allocation of resources (people, equipment) to specific tasks.
- **Progress Tracking**: As tasks are completed, the chart can be updated to show the progress made.
- **Adjustable**: Gantt charts are flexible and can be adjusted as project circumstances change.
- **Communication**: They facilitate communication among project team members, stakeholders, and managers by providing a clear overview of the project's status.

Overall, Gantt charts offer a visual way to plan, monitor, and manage projects effectively by displaying task relationships, timelines, and progress in a single, easy-to-understand format.

Gantt Chart



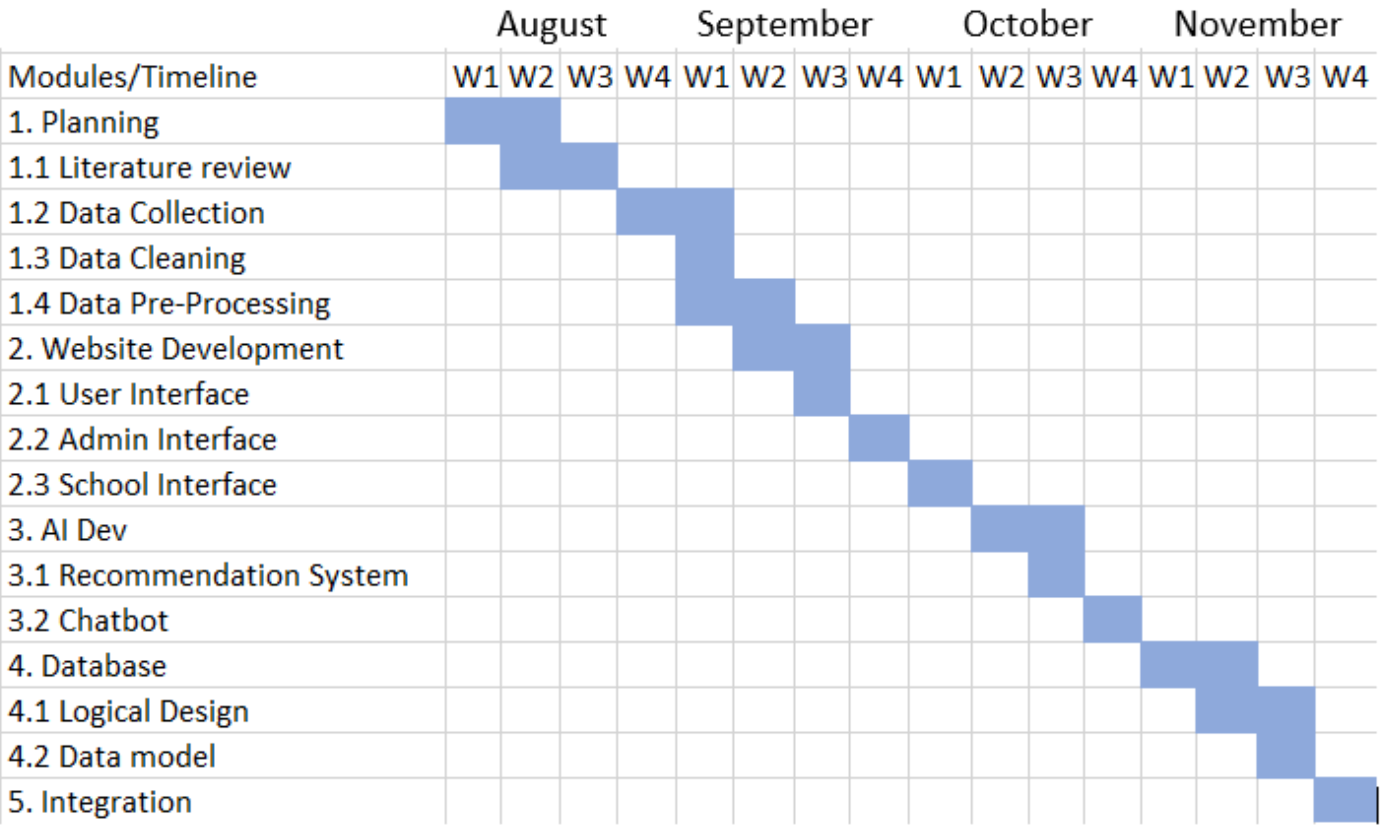
Introduction to Project:

Connect schools distinguishes itself as a trailblazing platform in the field of education by providing a unique method of school choosing. Connects shools uses AI and machine learning algorithms to generate personalized recommendations for students, assisting them in choosing the best colleges and universities.

This platform uses a variety of variables, including location, curriculum, extracurricular activities, academic achievement, and personal preferences, to provide personalized recommendations that are catered to the needs of each student.

Connect schools can analyze enormous amounts of data and produce insightful analysis thanks to the integration of AI and machine learning. The system continuously improves its recommendations over time by taking into account user interactions and comments, making sure that it stays current and relevant in the face of the always shifting educational scene.

Implementation of Gantt Chart using School Management System:



	August				September				October				November			
Modules/Timeline	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
1. Planning	▨	▨														
1.1 Literature review		▨	▨													
1.2 Data Collection				▨	▨											
1.3 Data Cleaning					▨											
1.4 Data Pre-Processing					▨	▨										
2. Website Development						▨	▨									
2.1 User Interface							▨									
2.2 Admin Interface								▨								
2.3 School Interface									▨							
3. AI Dev										▨	▨					
3.1 Recommendation System											▨					
3.2 Chatbot												▨				
4. Database													▨	▨		
4.1 Logical Design														▨	▨	
4.2 Data model															▨	
5. Integration																▨

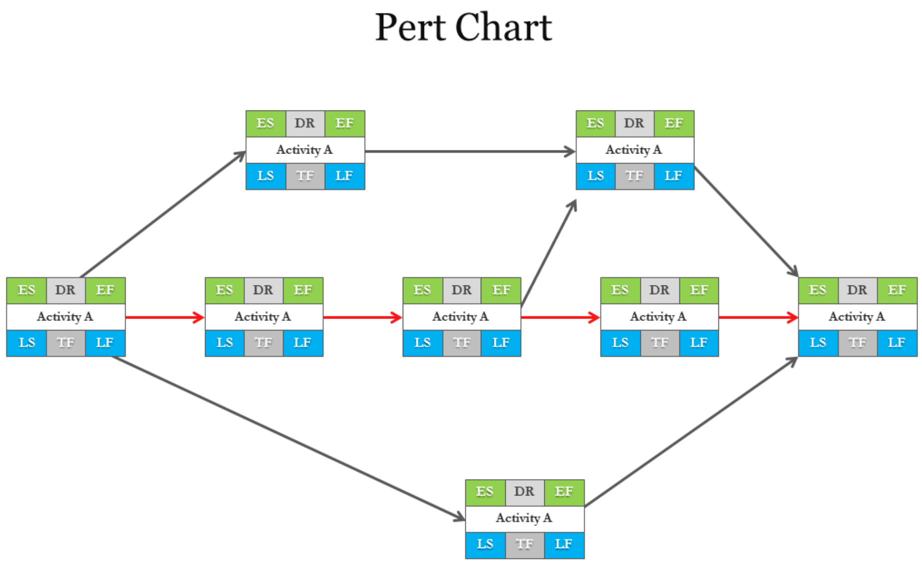
Practical-2

Aim: To prepare PERT Chart for a web application.

Software used: draw.io

Introduction to PERT Chart: A (Project Evaluation and Review Technique) PERT chart is a project management tool used to schedule, organize and coordinate tasks within a project.

- It provides a graphical representation of a project's timeline that enables project managers to break down each task in the project for analysis.
- The PERT chart template uses nodes -- drawn as rectangles or circles -- to represent events and milestones throughout the project.
- The nodes are connected by vectors -- drawn as lines -- which represent the various tasks that need to be completed. PERT charts provide project managers with an estimation of the minimum amount of time needed to complete a project. Managers can also analyze the work breakdown and task connections, as well as assess the risk associated with the project.

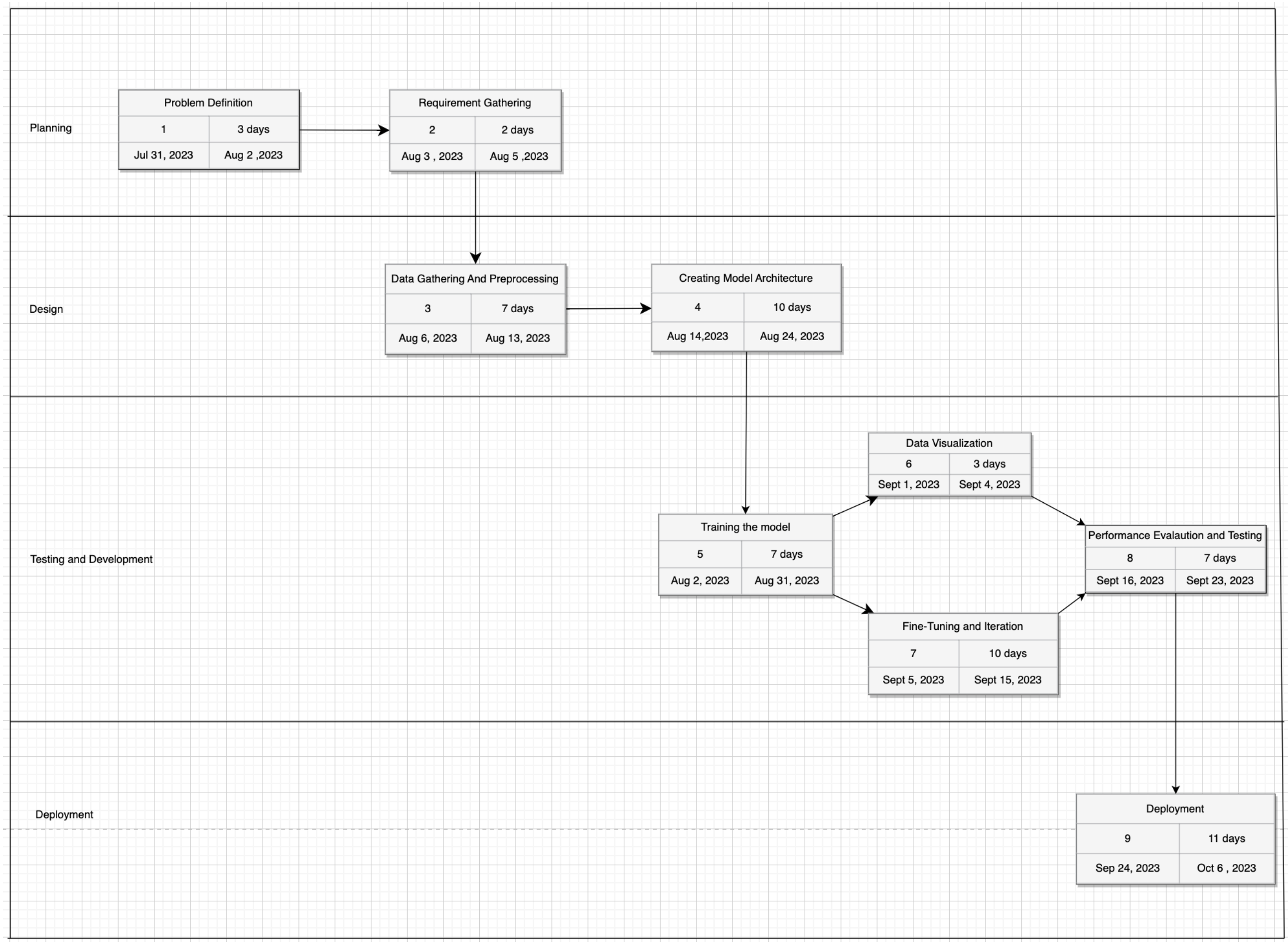


Introduction to Project:

The PERT chart outlines the development journey of a dynamic and user-friendly web application. This project involves several interconnected stages, beginning with the planning phase. In this stage, the project team will gather requirements, define objectives, and allocate resources

- The PERT chart shows the tasks and dependencies involved in developing a dynamic and user-friendly web application.
- The project begins with the requirements gathering phase, followed by design, development, deployment, and maintenance and support.
- The critical path is the sequence of tasks that must be completed on time in order for the project to finish on time.
- Tasks with more slack are less critical and can be rescheduled if necessary.
- The PERT chart can be used by the project team to plan and coordinate their work, and to identify and mitigate potential risks.

Implementation of PERT Chart for a Web Application:



Practical-3

Aim: To prepare SRS document for Intelligent Self-Driving Car System for Mountainous Terrain.

Software used: draw.io

Introduction to SRS Document : An SRS document, or Software Requirements Specification document, is a comprehensive and formal document that outlines the detailed requirements for a software project. It serves as a blueprint for the development team, providing a clear and structured description of what the software is expected to achieve and how it should behave. The SRS document acts as a communication bridge between clients, stakeholders, and the development team, ensuring everyone has a shared understanding of the project's scope and objectives.

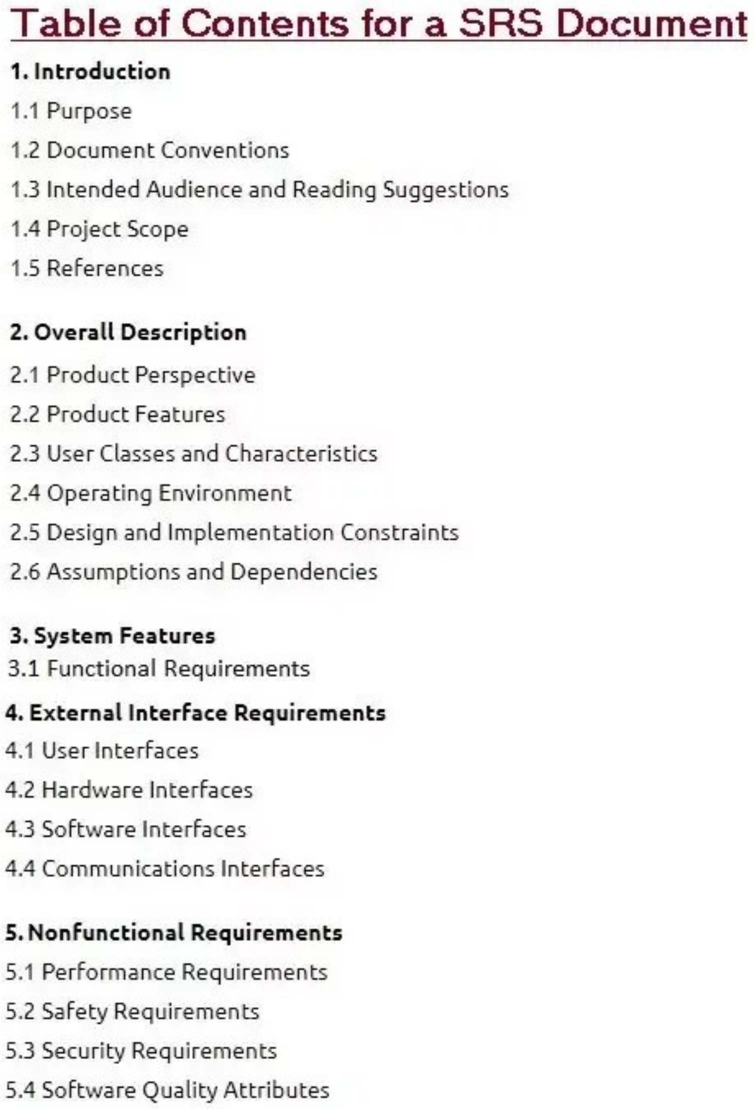


Table of Contents for a SRS Document

1. Introduction

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 References

2. Overall Description

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

3. System Features

3.1 Functional Requirements

4. External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

Introduction to Project:

The Intelligent Self-Driving Car System for Mountainous Terrain represents a pioneering leap in transportation technology. This system incorporates state-of-the-art self-driving technology and dynamic weather prediction capabilities to navigate the unique challenges posed by mountainous terrain. Like the Diabetes Prediction System, our project prioritizes user-friendly interfaces, robust security, and adherence to regulatory standards, aiming to revolutionize transportation safety and efficiency in complex environments.

Implementation of SRS document for Intelligent Self-Driving Car System for Mountainous Terrain:

An Intelligent Self-Driving Car System with Dynamic Weather Prediction for Mountainous Terrain

1. Introduction

The Intelligent Self-Driving Car System with Dynamic Weather Prediction for Mountainous Terrain is an advanced autonomous driving system designed to navigate vehicles in challenging mountainous regions, considering changing weather conditions. This system utilizes artificial intelligence, computer vision, and machine learning algorithms to ensure safe and efficient navigation, taking into account real-time weather updates and road conditions. This SRS document outlines the functional and non-functional requirements of the system.

2. Scope

The scope of the Intelligent Self-Driving Car System includes the following features:

- Autonomous navigation in mountainous terrain
- Real-time weather prediction and adaptation
- Obstacle detection and avoidance
- Lane detection and following
- Dynamic speed control based on road conditions and weather
- Human-machine interface for manual intervention
- Logging and monitoring system for system performance evaluation

3. Functional Requirements

3.1 Autonomous Navigation

The self-driving car system should be capable of safely navigating through mountainous terrains by using sensor inputs, such as Lidar, radar, and cameras, to detect road lanes, obstacles, and other vehicles. It should control acceleration, braking, and steering to follow a given path and maintain a safe distance from other vehicles.

3.2 Dynamic Weather Prediction

The system should integrate with a weather prediction service to receive real-time weather updates for the current route and location. The system should utilize this information to adjust driving strategies, such as reducing speed or finding alternative routes in case of adverse weather conditions.

3.3 Obstacle Detection and Avoidance

The system should be able to identify obstacles, such as rocks, fallen trees, or debris on the road, and take appropriate actions to avoid collisions. It should also be able to detect pedestrians and animals on or near the road.

3.4 Lane Detection and Following

The system should accurately detect lane markings and follow them safely. It should handle curved and narrow mountain roads while maintaining the appropriate speed and keeping within the lane boundaries.

3.5 Dynamic Speed Control

The system should adjust the vehicle's speed based on road conditions and weather predictions. It should reduce speed during heavy rain, snow, fog, or low visibility and increase speed during clear weather and good road conditions.

3.6 Human-Machine Interface

The system should provide a user-friendly interface to allow manual intervention by a human driver if necessary. The interface should be accessible and intuitive, enabling the human driver to take over control quickly and safely.

3.7 Logging and Monitoring System

The system should maintain logs of system activities, sensor data, and weather predictions. It should also include monitoring tools to evaluate the system's performance and identify potential issues.

4. Non-Functional Requirements

4.1 Safety

The system must prioritize safety and be designed to avoid accidents and collisions. It should meet or exceed industry safety standards.

4.2 Reliability

The self-driving car system should be highly reliable, with a low failure rate and a robust fault tolerance mechanism.

4.3 Performance

The system should perform real-time processing of sensor data and weather predictions efficiently to ensure smooth and responsive driving.

4.4 Scalability

The system should be scalable to accommodate updates and improvements as technology and algorithms evolve.

4.5 Security

The system should implement robust security measures to prevent unauthorized access and potential cyber-attacks.

4.6 Legal and Regulatory Compliance

The system should comply with all applicable laws, regulations, and standards related to autonomous driving and data privacy.

5. Constraints

- The system should operate within the limitations of the vehicle's hardware and sensors.
- Real-time weather data availability and accuracy depend on the chosen weather prediction service.

6. Assumptions

- The vehicle is equipped with suitable sensors for autonomous driving in mountainous terrain.
- The system has access to a reliable and accurate weather prediction service.

7. Glossary

- Lidar: Light Detection and Ranging, a remote sensing method that uses lasers to measure distances and create a 3D map of the surroundings.
- Radar: Radio Detection and Ranging, a sensing method that uses radio waves to detect the presence and location of objects.
- Computer Vision: A field of artificial intelligence that enables machines to interpret visual information from the world.

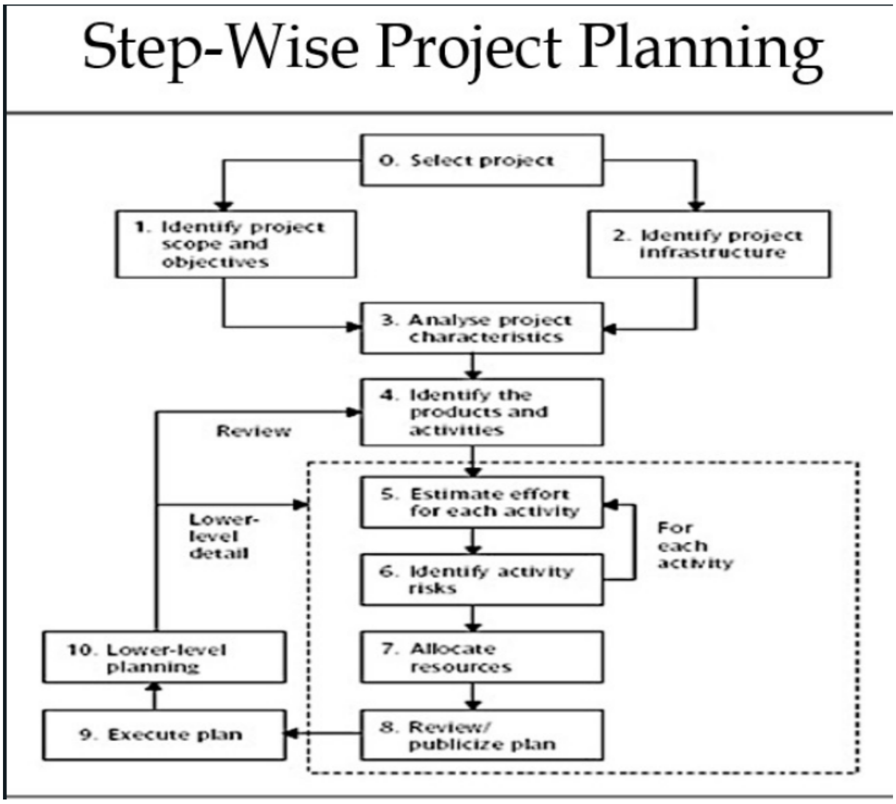
This SRS document outlines the essential requirements of the Intelligent Self-Driving Car System with Dynamic Weather Prediction for Mountainous Terrain. Further development, design, and implementation should follow these guidelines to build a functional and reliable autonomous driving solution.

Practical-4

Aim: To prepare a Step Wise Project Planning Chart for automotive e-commerce website.

Software used: draw.io

Introduction to Step Wise Project Planning Chart: A Step Wise Project Planning Chart is a visual roadmap for project management, presenting tasks in a sequential order to ensure a clear, transparent, and efficient project progression. It helps with resource allocation, timeline management, and serves as a communication tool for project managers to keep stakeholders informed and ensure project success. This chart provides a systematic approach to project execution, helping teams prioritize tasks and allocate resources effectively. It enhances project control by enabling quick identification of bottlenecks and deviations from the planned schedule. Additionally, it promotes accountability within the team, as each task and its responsible party are clearly defined, ensuring smoother project delivery.


Step-Wise Project Planning
0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
5. Estimate effort for each activity
Lower-level detail
For each activity
6. Identify activity risks
7. Allocate resources
10. Lower-level planning
9. Execute plan
8. Review/ publicize plan

Introduction to Project:

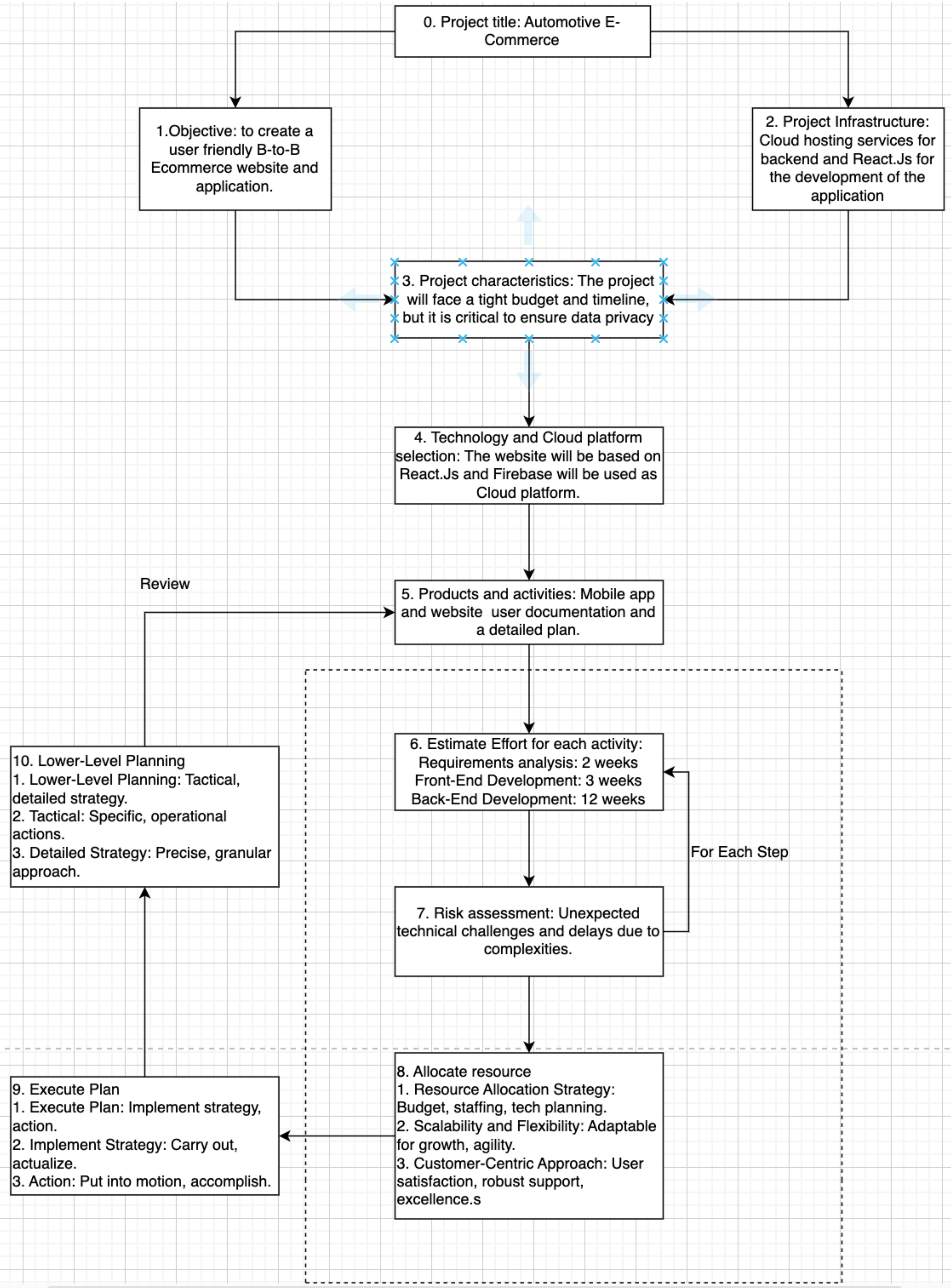
Creating an automotive e-commerce website is a dynamic venture that demands meticulous planning and execution. We begin by building a robust database of vehicles, ensuring it remains updated with the latest models, prices, and features. This foundation guarantees our users access to comprehensive and accurate information when exploring their automotive choices.

To set our platform apart, we focus on personalization. Machine learning algorithms analyze user behavior and preferences, offering tailored recommendations and an intuitive search experience. This customization simplifies the decision-making process and empowers users to discover their ideal vehicle with ease.

Quality assurance is paramount. Rigorous testing and continuous optimization underpin our commitment to a seamless and responsive user interface. Feedback from our users informs these enhancements, ensuring our website consistently meets high-performance standards and user satisfaction.

In an ever-evolving e-commerce landscape, ethical practices and regulatory compliance are non-negotiable. Our dedication to these principles guarantees a secure and trustworthy automotive shopping experience. We aspire to revolutionize the way people engage with online vehicle purchases, making it an enjoyable, informed, and hassle-free journey.

Implementation of Step Wise Project Planning Chart for automative e-commerce website:



0. Project title: Automotive E-Commerce

1.Objective: to create a user friendly B-to-B Ecommerce website and application.

2. Project Infrastructure: Cloud hosting services for backend and React.Js for the development of the application

3. Project characteristics: The project will face a tight budget and timeline, but it is critical to ensure data privacy

4. Technology and Cloud platform selection: The website will be based on React.Js and Firebase will be used as Cloud platform.

5. Products and activities: Mobile app and website user documentation and a detailed plan.

6. Estimate Effort for each activity:
Requirements analysis: 2 weeks
Front-End Development: 3 weeks
Back-End Development: 12 weeks

7. Risk assessment: Unexpected technical challenges and delays due to complexities.

8. Allocate resource
1. Resource Allocation Strategy: Budget, staffing, tech planning.
2. Scalability and Flexibility: Adaptable for growth, agility.
3. Customer-Centric Approach: User satisfaction, robust support, excellence.s

9. Execute Plan
1. Execute Plan: Implement strategy, action.
2. Implement Strategy: Carry out, actualize.
3. Action: Put into motion, accomplish.

10. Lower-Level Planning
1. Lower-Level Planning: Tactical, detailed strategy.
2. Tactical: Specific, operational actions.
3. Detailed Strategy: Precise, granular approach.

Practical-5

Aim: To prepare a Critical Path Method (CPM) diagram.

Software used: draw.io

Introduction to Critical Path Method (CPM) diagram :

Critical Path Method (CPM) is a method used in project planning, generally for project scheduling for the on-time completion of the project. It actually helps in the determination of the earliest time by which the whole project can be completed. There are two main concepts in this method namely critical task and critical path. Critical task is the task/activity which can't be delayed otherwise the completion of the whole project will be delayed. It must be completed on-time before starting the other dependent tasks.

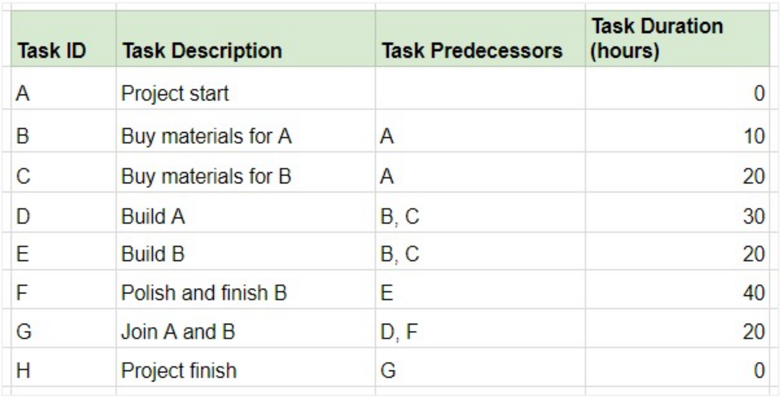
Critical path is a sequence of critical tasks/activities and is the largest path in the project network. It gives us the minimum time which is required to complete the whole project. The activities in the critical path are known as critical activities and if these activities are delayed then the completion of the whole project is also delayed.

Major steps of the Critical Path Method:

- Identifying the activities
- Construct the project network
- Perform time estimation using forward and backward pass
- Identify the critical path

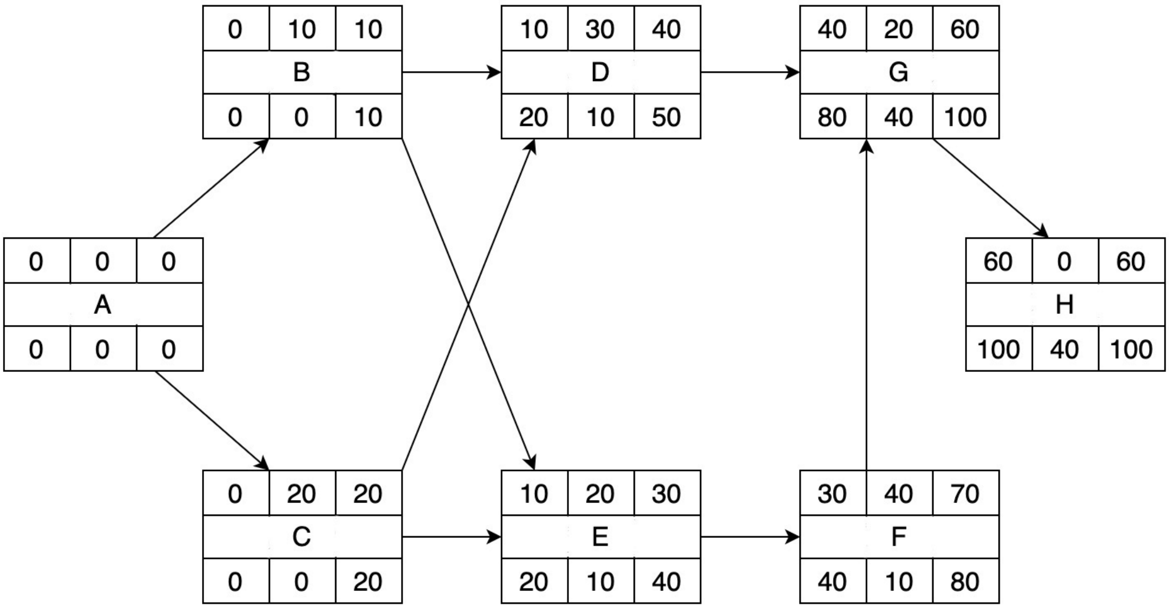
Introduction to Project:

Make a CPM for this table



Task ID	Task Description	Task Predecessors	Task Duration (hours)
A	Project start		0
B	Buy materials for A	A	10
C	Buy materials for B	A	20
D	Build A	B, C	30
E	Build B	B, C	20
F	Polish and finish B	E	40
G	Join A and B	D, F	20
H	Project finish	G	0

Implementation of Critical Path Method (CPM) diagram:



Practical-6

Aim: To prepare a Ball Chart Diagram.

Software used: draw.io

Introduction to Ball Chart Diagram: A ball chart SPM diagram is a type of Gantt chart that is used to visualize the timeline and dependencies of a software project. It uses balls instead of bars to represent tasks, and the balls are color-coded to represent the status of the task. The arrows between the balls show the dependencies between tasks.

The ball chart SPM diagram is a useful tool for project managers to track the progress of a software project and identify any potential bottlenecks. It can also be used to communicate the project timeline and dependencies to other stakeholders.

The arrows between the tasks would show that the development task cannot start until the design task is finished. The project manager could use this information to monitor the development task closely and take action if necessary to ensure that it is completed on schedule.

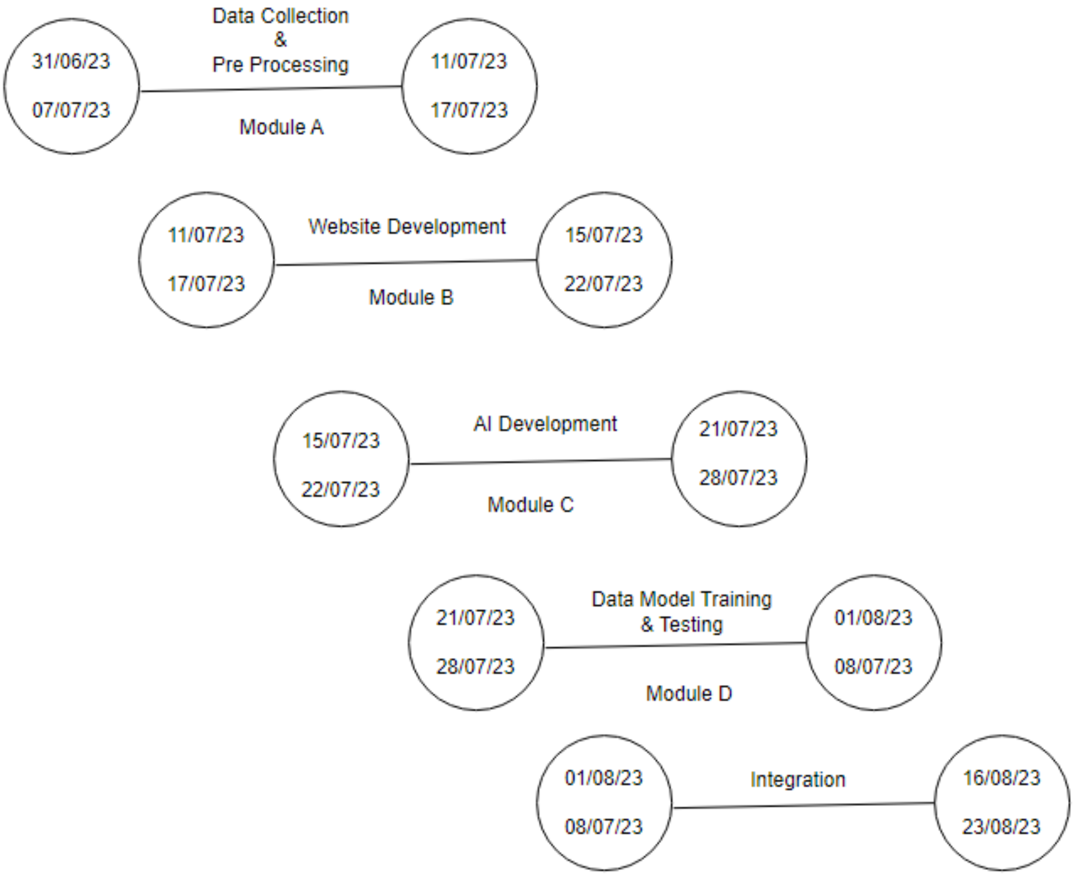
Overall, the ball chart SPM diagram is a valuable tool for project managers to track the progress of software projects and identify any potential bottlenecks.

Introduction to Project:

The project is scheduled to start on June 30th and end on August 1st, with four modules:

Module A must be completed before Module B can start, and Module C must be completed before Module D can start. All four modules must be completed before the data can be used to train and test the machine learning model. The project is on track to be completed on time and within budget.

Implementation of Ball Chart Diagram:



Practical-7

Aim: To prepare Timeline Chart Diagram.

Software used: draw.io

Introduction to Timeline Chart Diagram: A timeline chart of software project management is a visual representation of the tasks and activities that need to be completed in order to deliver a software project on time and within budget. It shows the start and end dates for each task, as well as the dependencies between tasks.

- Timeline charts are an essential tool for software project managers, as they help to:
- Plan and schedule the project: Timeline charts help project managers to break down the project into smaller tasks and to estimate the time required to complete each task. This information can then be used to create a realistic project schedule.
- Track progress: Timeline charts can be used to track the progress of the project and to identify any tasks that are falling behind schedule. This allows project managers to take corrective action early on, before the project is delayed.
- Communicate with stakeholders: Timeline charts can be used to communicate the project schedule to stakeholders, such as clients and executives. This helps to keep everyone informed of the progress of the project and to manage expectations.

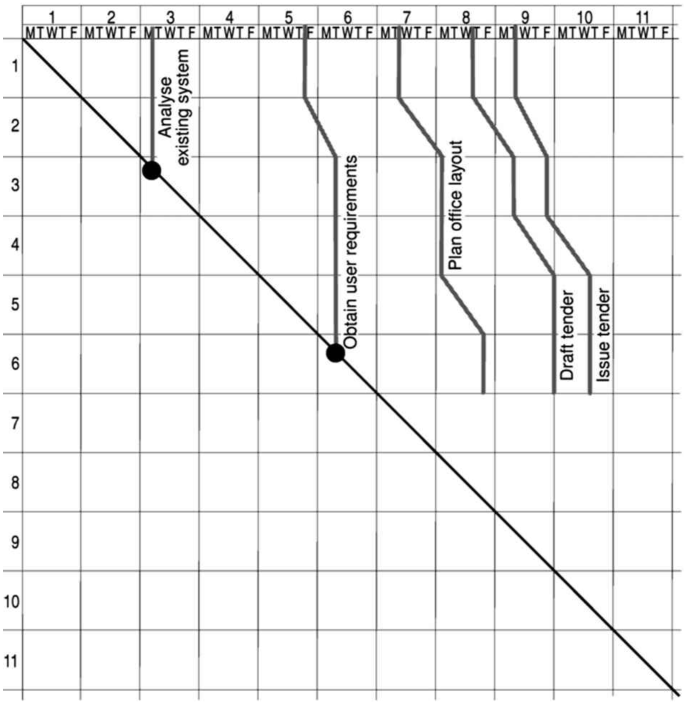
Introduction to Project:

The timeline diagram in the image shows the progress of a software project, from obtaining user requirements to planning the office layout. The diagram is divided into weeks, with each week labeled MTWTF. The tasks are listed in chronological order, with the start and end dates for each task indicated.

The timeline diagram shows that the project is expected to take 11 weeks to complete. The first three weeks are spent on analysing the existing system, drafting the tender, and issuing the tender. The next two weeks are spent obtaining user requirements and planning the office layout. The remaining six weeks are spent on developing and testing the software.

The timeline diagram is a useful tool for project management because it allows you to visualize the progress of the project and identify any potential bottlenecks. It can also be used to communicate the project schedule to stakeholders.

Implementation of Timeline Chart Diagram:



Practical-8

Aim: To prepare a Checkpoints and Milestones of a project.

Software used: draw.io

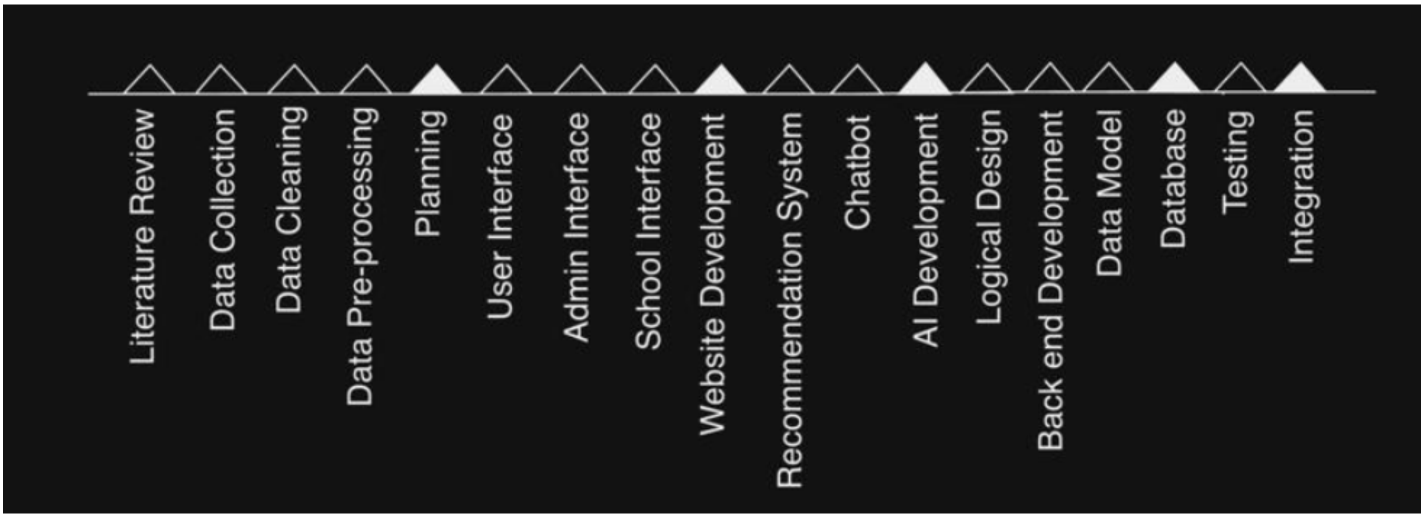
Introduction to Checkpoints and Milestones of a project.:
A milestone project diagram is a type of timeline diagram that focuses on the key milestones in a project. Milestones are important events in a project's life cycle that mark significant progress. They can be used to track the progress of the project, identify potential bottlenecks, and communicate the project schedule to stakeholders.

- Milestone project diagrams typically show the following information:
- The name of each milestone
- The start and end dates for each milestone
- The status of each milestone (e.g., complete, in progress, incomplete)
- Dependencies between milestones

Milestone project diagrams can be created using a variety of tools, including project management software, spreadsheets, and drawing software.

Introduction to Project:
The project begins with a literature review to identify the key research questions that the project will address. Once the research questions have been identified, the project team will collect data to answer them. The data will be cleaned and pre-processed to prepare it for analysis. The project team will then develop a plan for analysing the data, which will include identifying the methods that will be used and the timeline for the analysis. Once the plan is in place, the project team will begin developing the user interface, admin interface, school interface, and website for the data collection project. They will also develop a recommendation system to recommend data collection tasks to users and a chatbot to answer users' questions.
In parallel with the development of the front-end and back-end components of the project, the project team will develop AI algorithms to analyze the data collected. The algorithms will be developed using a logical design and a data model. The data will be stored in a database, which will be developed and integrated with the AI algorithms.
Finally, the project team will test the data collection project to ensure that it meets the requirements and that it is free of bugs. Once the testing is complete, the project will be ready to be deployed.

Implementation of Checkpoints and Milestones of a project:



Practical-9

Aim: To prepare a Risk Matrix of a project.

Software used: draw.io

Introduction to Risk Matrix of a project:

A risk matrix diagram is a tool used by project managers to identify and prioritize risks. It is a grid that plots the likelihood and impact of each risk. The likelihood of a risk is the probability that it will occur, and the impact of a risk is the severity of the consequences if it does occur.
The risk matrix diagram can be used to:

- Identify the risks that are most likely to occur and have the greatest impact on the project.
- Prioritize the risks and develop mitigation strategies accordingly.
- Track the progress of risk mitigation and identify any new risks that may have arisen.

Introduction to Project:

The risk matrix diagram shows the risks that are most likely to occur and have the greatest impact on the project. The risks involved are:
1. Data Security data security breach: The risk of unauthorized access or disclosure of sensitive data, potentially leading to privacy violations and legal consequences.

2. Model Training and Integration model overfitting: The risk of machine learning models performing well on training data but poorly on real-world data due to overly complex or over-optimized models.

3. User Interface User interface complexity: The risk of confusing or overwhelming users with a complex and unintuitive interface, leading to user dissatisfaction and reduced usability.

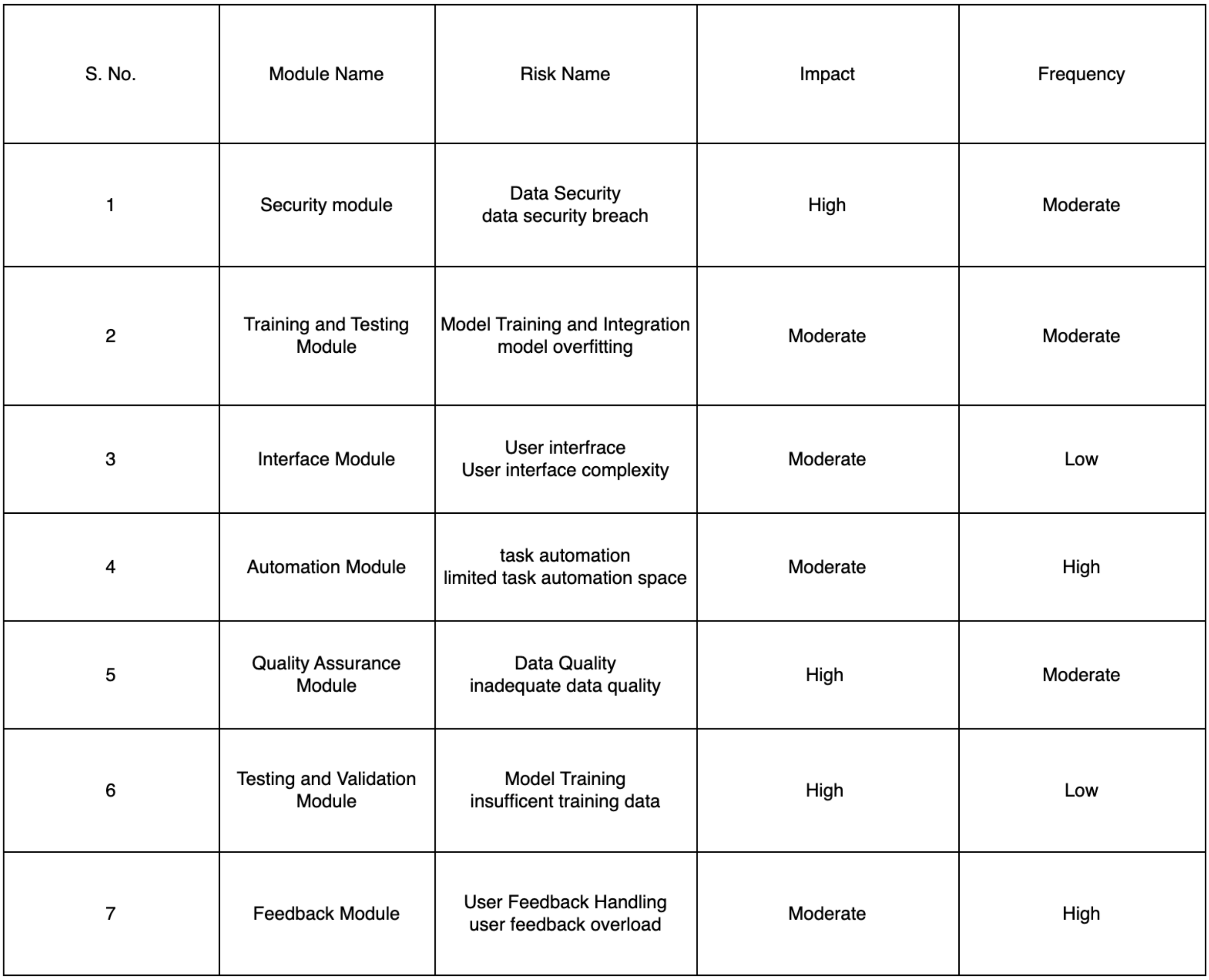
4. Task Automation limited task automation space: The risk of being unable to automate certain tasks efficiently, reducing the potential benefits of automation in a workflow.

5. Data Quality inadequate data quality: The risk of using incomplete, inaccurate, or inconsistent data, which can lead to erroneous insights and decisions.

6. Model Training insufficient training data: The risk of models underperforming due to a lack of diverse and representative training data, limiting their effectiveness.

7. User Feedback Handling user feedback overload: The risk of being overwhelmed by a large volume of user feedback, making it challenging to effectively manage and address user concerns and suggestions.

Implementation of Risk Matrix of a project:



S. No.	Module Name	Risk Name	Impact	Frequency
1	Security module	Data Security data security breach	High	Moderate
2	Training and Testing Module	Model Training and Integration model overfitting	Moderate	Moderate
3	Interface Module	User interfrace User interface complexity	Moderate	Low
4	Automation Module	task automation limited task automation space	Moderate	High
5	Quality Assurance Module	Data Quality inadequate data quality	High	Moderate
6	Testing and Validation Module	Model Training insufficent training data	High	Low
7	Feedback Module	User Feedback Handling user feedback overload	Moderate	High