SOFTWARE ENGINEERING

DELIVERABLE 2

SANJANA M (PES1UG22CS527)

SANNIDHI V K (PES1UG22CS533)

Project Plan Document: Railway Management System

1. Project Lifecycle Model:
   Chosen Model: Agile methodology
   Given that DBMS projects often require iterative development, especially with evolving queries and database structures, Agile allows for continuous feedback and iteration. It helps in early detection of issues and ensures adaptability to changes in user requirements or database optimizations.

2. Tools for Project Lifecycle:

- Planning Tool: Excel or Trello or Jira for task assignment, tracking progress, and collaboration.
- Design Tool: Canva or Lucidchart for ER diagrams and data flow designs.
- Version Control: Git (GitHub) to manage versions of database schemas, stored procedures, and application code.
- Development Tool: MySQL for the database, and Python (with MySQL connector or Flask framework) for scripting or connecting the database with a web interface.
- Bug Tracking: GitHub Issues for identifying and tracking bugs in the system.
- Testing Tool: PHPUnit for testing any PHP code or JUnit for Java-based applications; Postman for API testing (if applicable).

3. Deliverables and Components:
   Reuse Components:
   -Standard SQL queries for CRUD operations (Create, Read, Update, Delete).
   -Predefined libraries for connecting the database (e.g., MySQL Connector for Python).
   Build Components:
   -Custom triggers and stored procedures for automatic updates or logs.
   -A detailed entity-relationship (ER) model to reflect railway-related entities (e.g., trains, stations, tickets, passengers).
   -Web or app interface to interact with the database for booking and inquiry functionalities.

   Standard SQL queries can be reused across different DBMS projects, whereas specific stored procedures and UI elements will be custom-built to fit the requirements of a Railway Management System

4. Work Breakdown Structure (WBS):
   Database Setup:
   - Requirements gathering.
   - ER Diagram design.

- Normalization of tables.
- Setting up tables and relationships.
- Creating stored procedures and triggers.

User Interface:

- Design web interface (frontend).
- Integrate with the database (backend).

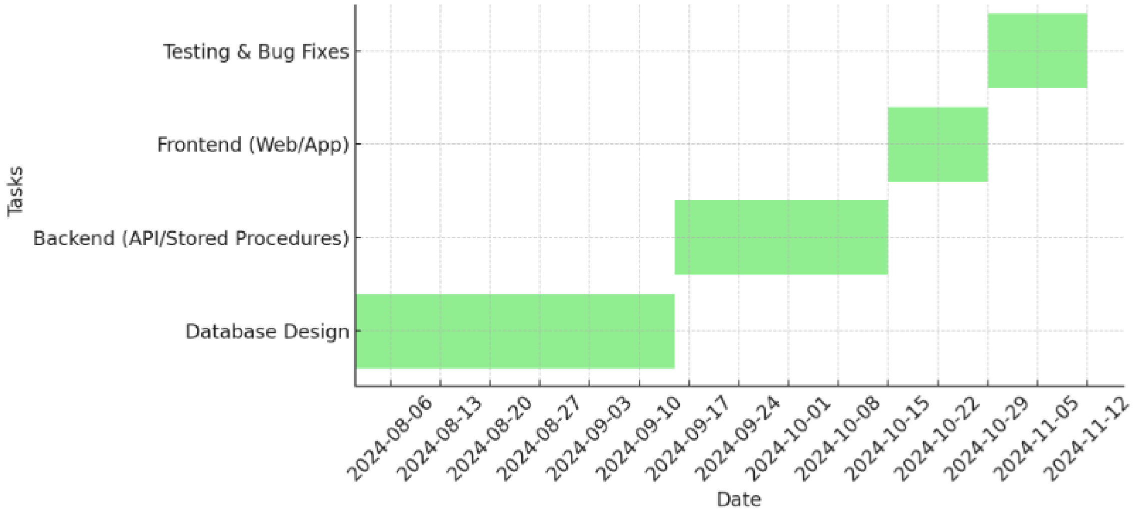
Core Functions:

- Ticket booking module.
- Ticket cancellation and refund system.
- Train schedule and availability tracking.
- User authentication and role management (admin vs user).

5. Effort Estimation and Gantt Chart

Database Design: 1.5 person-months (including ER diagrams, table creation, and normalization).
Backend (API/Stored Procedures): 1 person-month.
Frontend (Web/App): 0.5 person-month.
Testing & Bug Fixes: 0.5 person-month.
Total Effort: ~3.5 person-months (divided among 2 people).

6. Gantt chart:



7. Coding details:

The main coding components will include:
-SQL scripts for setting up the database.
-Stored procedures for railway operations like booking and cancellation.
-Backend code (using Python or any other language to interact with the DB).