



MANIPAL UNIVERSITY
JAIPUR

Major Project

Project - Synopsis

MCA - IV Sem

Subject Code: CA7270

Submitted By

Student Name-Harsh Gupta

Student Registration-23FS20MCA00002

Faculty Coordinator

Dr. Amit Hirawat, Associate Professor

DEPARTMENT OF COMPUTER APPLICATIONS

Title of the Project:

Employee Task Management System

Team Details:

- **Team Members:**
 - Harsh Gupta - Backend Development

Introduction:

The **Employee Task Management System** is a web-based application that streamlines the process of task assignment, tracking, and completion for employees in an organization. The backend, developed using **Node.js and SQL**, ensures efficient data management, user authentication, and seamless API communication. The system aims to improve **task management, time tracking, and performance monitoring** while providing a secure and scalable solution.

Objectives:

System Features (Backend):

User Authentication & Authorization:

- Secure **JWT-based authentication** for admins and employees.
- Role-based access control to restrict unauthorized actions.

Task Management API:

- Create, update, and delete tasks.
- Assign tasks to employees dynamically.
- Track task progress and completion status.

Time Tracking System:

- Employees can log time spent on tasks.

- Calculate and store remaining time for tasks.

Comments & Feedback API:

- Employees can add comments to assigned tasks.
- Managers can review and provide feedback.

Database Management:

- Efficient storage and retrieval of task and user data.
- Optimized SQL queries for better performance.

Architecture & Data Flow:

1. API Endpoints:

- **User Routes:** /register, /login, /profile
- **Task Routes:** /tasks, /tasks/:id (CRUD operations)
- **Comment Routes:** /tasks/:id/comments
- **Time Tracking Routes:** /tasks/:id/time-tracking

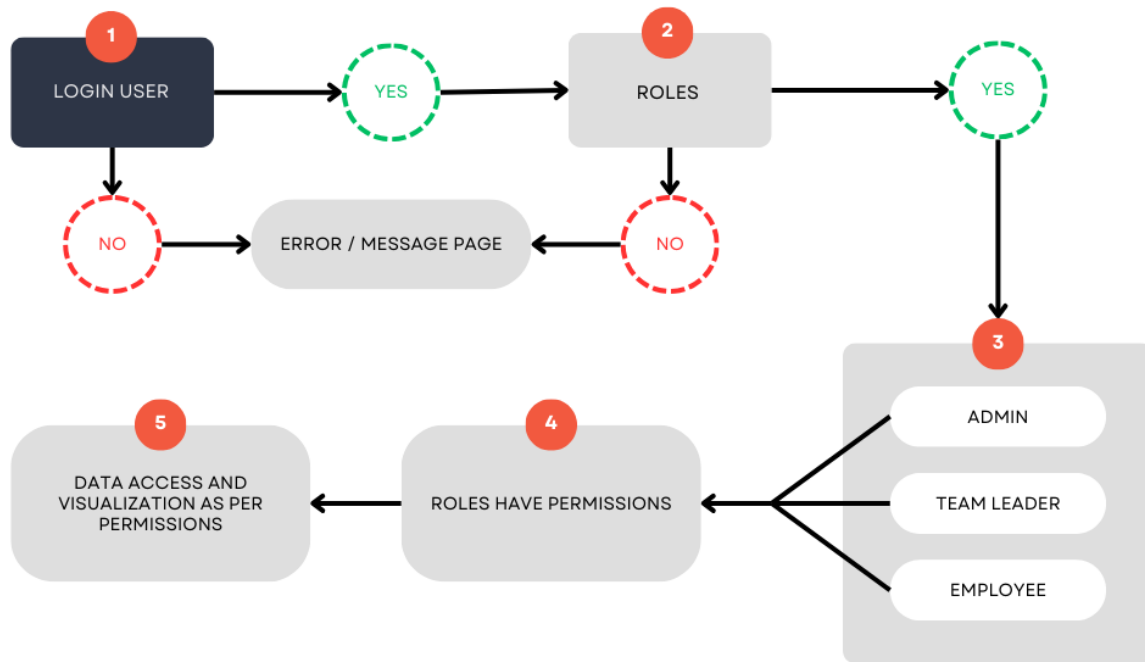
DFD (Data Flow Diagram):

The Data Flow Diagram outlines the flow of information:

1. **User Interactions:** Employees and Managers login and perform actions.
2. **Task Assignment:** Managers create and assign tasks with deadlines.
3. **Task Updates:** Employees update task progress, tracking time spent.
4. **Notifications & Reports:** System generates alerts and task performance reports.

Data Flow Diagram

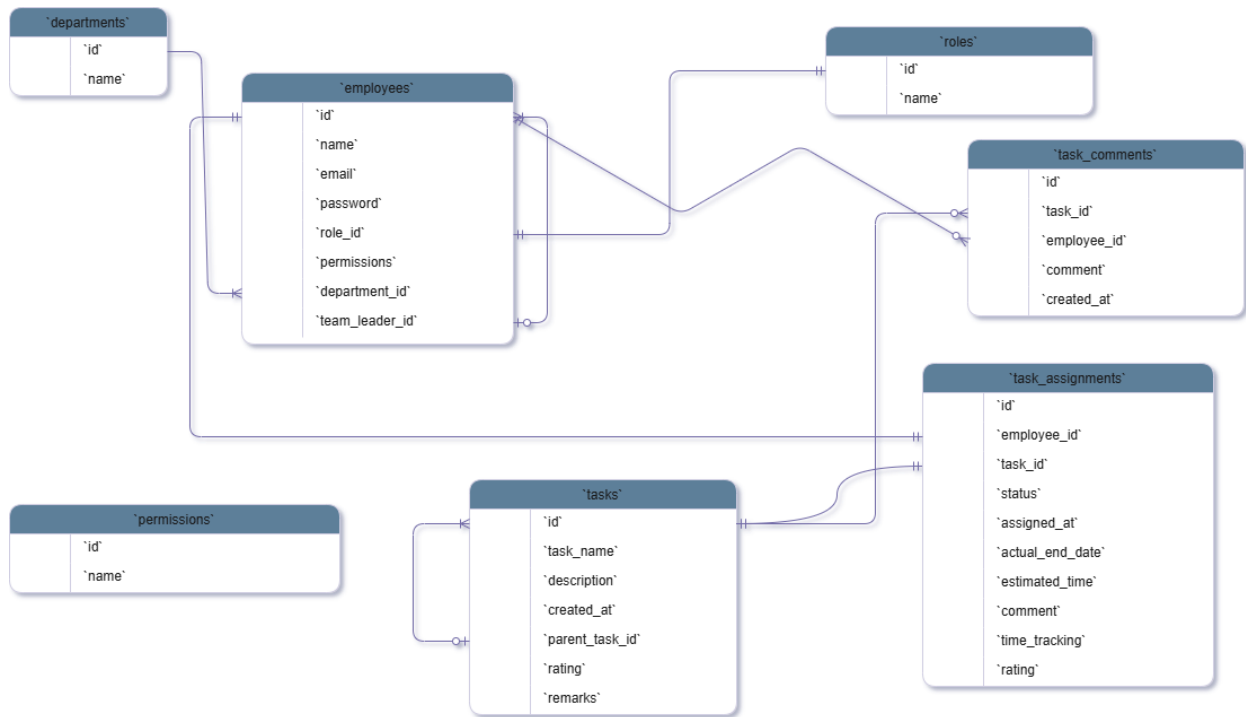
Flowchart



ER Diagram:

The Entity-Relationship diagram includes the following entities:

- **User** (_id, name, email, password, role, createdAt, updatedAt)
- **Task** (_id, title, description, assigned_to, status, start_date, end_date, estimate_time, actual_end_date, createdAt, updatedAt)
- **Comment** (_id, task_id, user_id, message, createdAt, updatedAt)
- **TimeTracking** (_id, task_id, user_id, time_spent, time_remaining, createdAt, updatedAt)
- **Notification** (_id, user_id, message, createdAt, updatedAt)



Project Timeline:

- **Phase 1:** Planning & Design - January 1 to January 20
- **Phase 2:** Frontend Development - January 21 to March 10
- **Phase 3:** Testing & Deployment - April 16 to May 5

Tools / Platform, Hardware and Software Requirement Specifications:

- **Backend Framework:** Node.js (Express.js)
- **Database:** SQL (PostgreSQL/MySQL)
- **Authentication:** JWT (JSON Web Token)
- **API Testing Tools:** Postman, Swagger
- **Deployment:** Docker, AWS/Digital Ocean
- **Version Control:** Git & GitHub

References:

- <https://www.npmjs.com/package/react>
- <https://nodejs.org/docs/latest/api/>
- https://www.w3schools.com/nodejs/nodejs_mysql.asp
- <https://www.npmjs.com/package/mysql>