Database System Final Project

***Instructor: Ms. Zubaira Naz***

***Batch CS23 and AI23***

## 

## **Database Project Description**

**Group Member :**

BSCS-23070 Muhammad Abdullah

BSCS23212 Faizan Amir

BSCS23145 Ibrahim Bin Ahmed

### **1. Project Title:**

TitanFactory: Industrial Management Reinvented with Technology

### **2. Project Description:**

Purpose: This project aims to create a comprehensive industrial management system to streamline factory operations, employee management, machine maintenance, attendance tracking, and sales reporting. The system will integrate AI-based functionalities such as predictive reporting and a chatbot to assist with navigation and task execution. It also includes essential modules for payroll management, and notifications, ensuring smooth day-to-day operations.

**Objectives:**

* Factory Management: Track machine details, their maintenance history, and current operational status.
* Employee Management: Manage employees' attendance, roles, and departments efficiently.
* Sales Reporting: Generate and store sales reports with detailed salesperson performance.
* AI Integration: Use AI for report prediction (e.g., sales forecast) and provide chatbot assistance for navigation and task handling.
* Notification System: Notify users about critical events like task assignments, maintenance, or approvals.
* Payroll: Automate payroll calculations based on employee work hours, bonuses, and deductions.
* Authentication and Role Management: Secure login and role-based access to manage user permissions.

**Main Functionalities:**

* User Management & Authentication: Role-based access control (admin, manager, employee, IT, sales) with secure login.
* Machine Management: Monitor machine status, log maintenance activities, and ensure timely maintenance with managerial approvals.
* Attendance Tracking: Record employee check-ins, check-outs, and track attendance status.
* Sales Reporting: Keep records of sales reports with relevant performance data.
* AI Features: Provide predictive analysis for reports and assist users with a chatbot for navigation and help.
* Notifications: Notify users of critical updates, such as task assignments, approvals, or maintenance alerts.
* Payroll Management: Calculate and store payroll information based on hours worked, overtime, bonuses, and deductions.

This system is designed to optimize factory workflows, predict important trends, and provide seamless management across different departments. By combining real-time data with predictive analytics, the platform improves operational efficiency, reduces downtime, and enhances decision-making capabilities.

### 

### **3. Scope:**

**Key Features:**

1. User Management & Authentication: Role-based access control (admin, manager, employee , sales) with secure login.
2. Factory & Machine Management: Track machine statuses, log maintenance history, and approvals.
3. Attendance Tracking: Record employee check-ins/outs and attendance status.
4. Sales Reporting: Manage sales data and use AI for predictive sales trends.
5. AI Integration: Chatbot for system navigation and AI-powered report predictions.
6. Notifications: Notifications for important updates.
7. Payroll Management: Automate payroll calculations based on attendance and overtime.
8. Department & Factory Management: Manage department and factory details, including managers.

**Boundaries:**

* Included: Core functionalities like user management, machine tracking, attendance, sales, AI, and payroll.
* Excluded: Real-time IoT-based machine monitoring, full AI automation, advanced HR or financial systems, and third-party integrations.

The project focuses on optimizing factory workflows, improving decision-making with AI predictions, and streamlining management processes within a controlled scope.

### **4. Stakeholders:**

1. **Primary Users (Direct System Users)**:

* Admin: Manages the overall system, including user roles, permissions, and high-level reports.
* Factory Manager: Oversees factory operations, approves machine maintenance, monitors employee performance, and generates reports.
* Employees: Logs attendance and interacts with the AI chatbot for navigation.
* Sales Team: Inputs and tracks sales data, generates sales reports, and uses AI for predictive analysis.

1. **Secondary Users (Indirect Interactions):**

* Maintenance Staff : Updates machine maintenance records and logs tasks assigned by factory managers.

### **5. Entity-Relationship Diagram (ERD):**

### **Entities and Their Purposes**

1. **User**
   * **Purpose:** Store user information, including role-based access for various departments and system functionalities.
   * **Relevance:** Central to the system. Every other entity either directly or indirectly links back to the user.
   * **Key Relationships:**
     + Linked to **Department** (One-to-Many) – A user belongs to a department.
     + Linked to **Attendance** (One-to-Many) – Each user can have multiple attendance records.
     + Linked to **Sales\_Report** (One-to-Many) – Each salesperson user can generate multiple sales reports.
     + Linked to **Machine\_Maintenance\_History** (One-to-Many) – A user performs machine maintenance activities.
     + Linked to **Roles**: You may want to add a relationship to a "Role" or "Role\_Permissions" entity to better handle the different user roles (admin, employee, etc.) and their access permissions across different system features.
     + Linked to **Payroll** (One-to-One) – Each user has one payroll record.
     + Linked to **Notification** (One-to-Many) – A user receives multiple notifications.
     + Linked to **Chatbot\_Interaction** (One-to-Many) – A user can have multiple interactions with the chatbot.
2. **Department**
   * **Purpose:** Manage organizational department details.
   * **Relevance:** Links users to specific departments, helping with organization management.
   * **Key Relationships:**
     + Linked to **User** (One-to-Many) – A department has many users but each user belongs to one department.
     + **Manager\_ID** also links back to the **User** entity, indicating who manages the department.
3. **Machine**
   * **Purpose:** Track machine details within the factory.
   * **Relevance:** Ensures machinery management, including operational status, maintenance scheduling, and machine tracking.
   * **Key Relationships:**
     + Linked to **Factory** (One-to-Many) – A factory can have many machines, but each machine belongs to a single factory.
     + Linked to **Machine\_Maintenance\_History** (One-to-Many) – Each machine can have multiple maintenance logs.
4. **Machine\_Maintenance\_History**
   * **Purpose:** Record the maintenance history of machines.
   * **Relevance:** Critical for machine reliability and factory management.
   * **Key Relationships:**
     + Linked to **Machine** (Many-to-One) – Each maintenance record is linked to a specific machine.
     + Linked to **User** (Many-to-One) – Each maintenance activity is performed by a user.
     + **Manager\_Approval\_ID** links back to the **User** entity to indicate which user approved the maintenance.
5. **Attendance**
   * **Purpose:** Log employee attendance details.
   * **Relevance:** Essential for tracking employee work, attendance, and leave.
   * **Key Relationships:**
     + Linked to **User** (Many-to-One) – Each attendance record is linked to a specific user.
6. **Sales\_Report**
   * **Purpose:** Track and store sales data.
   * **Relevance:** Relevant for sales team users, tracking their contributions to the company’s revenue.
   * **Key Relationships:**
     + Linked to **User** (Many-to-One) – Each report is linked to a salesperson.
7. **Factory**
   * **Purpose:** Store and manage factory details.
   * **Relevance:** Links machines to specific factory locations, supporting industrial operations.
   * **Key Relationships:**
     + Linked to **Machine** (One-to-Many) – Each factory manages multiple machines.
     + Linked to **User** via **Manager\_ID** to specify the user managing the factory.
8. **Notification**
   * **Purpose:** Notify users of important events and updates.
   * **Relevance:** Ensures users receive critical updates relevant to their roles.
   * **Key Relationships:**
     + Linked to **User** (Many-to-One) – Each notification is assigned to a specific user or target role (e.g., admin, employee).
     + Can target all users through the target\_role column (e.g., 'all').
9. **Payroll**
   * **Purpose:** Manage payroll details, including salary, bonuses, and deductions for employees.
   * **Relevance:** Essential for employee compensation and management.
   * **Key Relationships:**
     + Linked to **User** (One-to-One) – Each user has a corresponding payroll record.
10. **Chatbot\_Interaction**
    * **Purpose:** Store user interactions with the AI chatbot.
    * **Relevance:** Keeps track of all communications between users and the system.
    * **Key Relationships:**
      + Linked to **User** (Many-to-One) – Each interaction is linked to a specific user for personalized responses.

### **11. Product**

* **Purpose**:
  + To manage detailed information about the products that the factory manufactures and/or sells.
* **Relevance**:
  + Essential for tracking product-specific data, which is crucial for sales reporting, inventory management, and production planning. Helps in analyzing product performance, pricing strategies, and market trends.
* **Key Relationships**:
  + **Linked to Sales\_Report** (Many-to-One): Each sales report is associated with specific products sold, allowing for detailed sales analysis per product.
  + **Linked to Inventory** (One-to-One): Each product has an associated inventory record to monitor stock levels and manage supply.

### **12. Inventory**

* **Purpose**:
  + To manage and monitor the stock levels of raw materials and finished products within the factory.
* **Relevance**:
  + Vital for ensuring that the factory maintains optimal inventory levels, preventing production delays due to stockouts and avoiding excess inventory that can lead to increased holding costs. Supports procurement planning and financial forecasting.
* **Key Relationships**:
  + **Linked to Product** (One-to-One): Each product has an associated inventory record to track its stock levels, reorder points, and movement.
  + **Linked to Supplier** (Many-to-One): Inventory items are sourced from suppliers, linking stock levels to supplier reliability and delivery performance.

### **13.Supplier**

* **Purpose**:
  + To manage detailed information about the suppliers who provide raw materials and products to the factory.
* **Relevance**:
  + Essential for maintaining a reliable supply chain, evaluating supplier performance, negotiating contracts, and ensuring timely procurement of necessary materials. Facilitates effective vendor management and cost control.
* **Key Relationships**:
  + **Linked to Inventory** (One-to-Many): Each supplier can provide multiple inventory items, enabling the factory to track which suppliers supply which products or materials.

**14. Role**

* + **Purpose:**Provide a dynamic way to manage roles
  + **Relevance:** Links users to roles

**Key Relationships:**

* + - Linked to **User** (One-to-Many) –One role can be assigned to many users

### **Entity Relationships**

#### **1. User ↔ Department**

* **Type**: One-to-Many
* **Description**: Each user is associated with one department, and each department can have multiple users.

#### **2. User ↔ Machine\_Maintenance\_History**

* **Type**: One-to-Many
* **Description**: A user can perform multiple maintenance activities for different machines.

#### **3. User ↔ Attendance**

* **Type**: One-to-Many
* **Description**: A user has multiple attendance records, but each attendance entry refers to a single user.

#### **4. User ↔ Sales\_Report**

* **Type**: One-to-Many
* **Description**: Sales users will have multiple sales reports associated with their activities.

#### **5. User ↔ Payroll**

* **Type**: One-to-One
* **Description**: Each user has one payroll entry, and each payroll entry refers to one user.

#### **6. User ↔ Notification**

* **Type**: One-to-Many
* **Description**: Each user can receive multiple notifications, but each notification is targeted to one user or role.

#### **7. User ↔ Chatbot\_Interaction**

* **Type**: One-to-Many
* **Description**: A user can have many chatbot interactions, but each interaction involves only one user.

#### **8. Factory ↔ Machine**

* **Type**: One-to-Many
* **Description**: Each factory houses multiple machines, and each machine belongs to one factory.

#### **9. Machine ↔ Machine\_Maintenance\_History**

* **Type**: One-to-Many
* **Description**: Each machine will have several maintenance records over time.

#### **10. Product ↔ Sales\_Report**

* **Type**: One-to-Many
* **Description**: Each product can be associated with multiple sales reports.

#### **11. Product ↔ Inventory**

* **Type**: One-to-One
* **Description**: Each product has one inventory record to monitor its stock levels and manage supply.

#### **12. Inventory ↔ Supplier**

* **Type**: Many-to-One
* **Description**: Each inventory item is supplied by one supplier, and a supplier can supply multiple inventory items..

#### **13. Supplier ↔ Inventory**

* **Type**: One-to-Many
* **Description**: Each supplier can supply multiple inventory items, enabling the factory to track which suppliers provide which products or materials.

**14. User ↔ Roles**

* **Type**: Many-to-One
* **Description**: A user is assigned one role, and each role can be assigned to multiple users.

### **6 .Schema Design:**

## **1. User**

**Purpose**: Store comprehensive user information and manage role-based access.

**Attributes:**

* user\_id (Primary Key) - INT, auto-increment
* first\_name - VARCHAR(100)
* last\_name - VARCHAR(100)
* email - VARCHAR(255), unique
* password - VARCHAR(255) (hashed password)
* role\_id (Foreign Key) - INT, references Roles(role\_id)
* phone\_number - VARCHAR(15)
* department\_id (Foreign Key) - INT, references Department(department\_id)
* currently\_employed - BOOLEAN (stores whether the user is currently employed: TRUE/FALSE)
* address - VARCHAR(255) (optional, for contact information)
* hire\_date - DATE (tracks when the employee was hired)
* job\_title - VARCHAR(100) (specifies the employee’s position)

**Primary Key:**

* user\_id

**Foreign Key:**

* department\_id references Department(department\_id)

**Candidate Key**:

* email

**Notes:**

* address: Useful for HR records and contact purposes.
* hire\_date and termination\_date: Essential for tracking employment duration and status.
* job\_title: Provides clarity on the employee's role within the company.

## **2. Department**

**Purpose:** Manage detailed department information.

**Attributes:**

* department\_id (Primary Key) - INT, auto-increment
* department\_name - VARCHAR(100)
* location - VARCHAR(255) (optional, specifies where the department is located)
* description - TEXT (optional, provides details about the department)

**Primary Key:**

* department\_id

**Notes:**

* location and description: Enhance the understanding of each department’s purpose and physical placement within the factory.

## **3. Machine**

**Purpose:** Track comprehensive details of machines used in the factory.

**Attributes:**

* machine\_id (Primary Key) - INT, auto-increment
* machine\_name - VARCHAR(100)
* model - VARCHAR(100)
* serial\_number - VARCHAR(100), unique (tracks individual machines)
* status - ENUM('active', 'maintenance', 'decommissioned')
* last\_maintenance\_date - DATE
* next\_maintenance\_due - DATE
* purchase\_date - DATE (when the machine was acquired)
* warranty\_expiry - DATE (optional, tracks warranty period)
* factory\_id (Foreign Key) - INT, references Factory(factory\_id)
* location - VARCHAR(255) (optional, specifies where the machine is located within the factory)

**Primary Key:**

* machine\_id

**Foreign Key:**

* factory\_id references Factory(factory\_id)

**Notes:**

* serial\_number: Ensures each machine can be uniquely identified.
* purchase\_date and warranty\_expiry: Important for asset management and maintenance planning.
* location: Helps in quickly locating machines within the factory premises.

## **4. Machine\_Maintenance\_History**

Purpose: Log detailed maintenance activities for machines.

**Attributes:**

* maintenance\_id (Primary Key) - INT, auto-increment
* machine\_id (Foreign Key) - INT, references Machine(machine\_id)
* performed\_by (Foreign Key) - INT, references User(user\_id)
* maintenance\_date - DATE
* maintenance\_details - TEXT
* maintenance\_type - ENUM('scheduled', 'unscheduled', 'emergency') (categorizes maintenance)
* maintenance\_cost - DECIMAL(10, 2) (tracks expenses)
* manager\_approval\_id (Foreign Key) - INT, references User(user\_id)
* next\_maintenance\_date - DATE (optional, schedules future maintenance)

**Primary Key:**

* maintenance\_id

**Foreign Keys:**

* machine\_id references Machine(machine\_id)
* performed\_by and manager\_approval\_id reference User(user\_id)

**Notes:**

* maintenance\_type and maintenance\_cost: Provide insights into the nature and financial aspects of maintenance activities.
* next\_maintenance\_date: Helps in proactive maintenance scheduling.

## 

## **5. Attendance**

Purpose: Record comprehensive employee attendance.

**Attributes:**

* attendance\_id (Primary Key) - INT, auto-increment
* user\_id (Foreign Key) - INT, references User(user\_id)
* date - DATE
* check\_in\_time - TIME
* check\_out\_time - TIME
* status - ENUM('present', 'absent', 'leave', 'late', 'early\_leave')
* remarks - TEXT (optional, for additional notes)

**Primary Key:**

* attendance\_id

**Foreign Keys:**

* user\_id references User(user\_id)

**Notes:**

* remarks: Allows for recording any special conditions or notes related to attendance.

## **6. Sales\_Report**

**Purpose:** Track detailed sales information.

**Attributes:**

* report\_id (Primary Key) - INT, auto-increment
* salesperson\_id (Foreign Key) - INT, references User(user\_id)
* product\_id (Foreign Key) - INT, references Product(product\_id) (if implementing products)
* sales\_amount - DECIMAL(10, 2)
* report\_date - DATE
* notes - TEXT

**Primary Key:**

* report\_id

**Foreign Keys:**

* salesperson\_id references User(user\_id)
* product\_id references Product(product\_id) (if products are implemented)

**Notes:**

* product\_id: Associates sales with specific products.
* region: Helps in analyzing sales geographically.
* target\_achievement: Useful for performance evaluations.

## **7. Factory**

Purpose: Store and manage detailed factory information.

**Attributes:**

* factory\_id (Primary Key) - INT, auto-increment
* factory\_name - VARCHAR(100)
* location - VARCHAR(255)
* manager\_id (Foreign Key) - INT, references User(user\_id)
* established\_date - DATE (optional, tracks when the factory was established)
* capacity - INT (optional, indicates production capacity)
* description - TEXT (optional, provides details about the factory)

**Primary Key:**

* factory\_id

**Foreign Key:**

* manager\_id references User(user\_id)

**Notes:**

* established\_date and capacity: Offer insights into the factory’s history and operational scale.
* description: Provides additional context about the factory.

## **8. Notification**

Purpose: Notify users of important events and updates.

**Attributes:**

* notification\_id (Primary Key) - INT, auto-increment
* user\_id (Foreign Key) - INT, references User(user\_id)
* message - TEXT
* created\_at - DATETIME
* target\_role - ENUM('admin', 'manager', 'employee', 'maintenance\_staff ', 'sales', 'all')
* notification\_type - ENUM('alert', 'reminder', 'info', 'warning') (categorizes notifications)

**Primary Key:**

* notification\_id

**Foreign Key:**

* user\_id references User(user\_id)

**Notes:**

* notification\_type: Helps in filtering and prioritizing notifications.
* is\_read: Useful for tracking user engagement with notifications.

## **9. Payroll**

Purpose: Manage comprehensive payroll information for employees.

**Attributes:**

* payroll\_id (Primary Key) - INT, auto-increment
* user\_id (Foreign Key) - INT, references User(user\_id)
* basic\_salary - DECIMAL(10, 2)
* overtime\_hours - INT
* overtime\_rate - DECIMAL(10, 2)
* bonuses - DECIMAL(10, 2)
* deductions - DECIMAL(10, 2)
* tax - DECIMAL(10, 2) (calculates applicable taxes)
* net\_salary - DECIMAL(10, 2)
* payment\_date - DATE
* bank\_account\_number - VARCHAR(50) (optional, for direct deposits)
* payment\_method - ENUM('bank\_transfer', 'cash', 'check') (optional)
* payment\_status ENUM('pending', 'completed', 'failed') NOT NULL DEFAULT 'pending'

**Primary Key**:

* payroll\_id

**Foreign Key:**

* user\_id references User(user\_id)

Notes:

* tax: Ensures accurate payroll calculations.
* bank\_account\_number and payment\_method: Facilitate various payment options for employees.

## **10. Chatbot\_Interaction**

**Purpose:** Store detailed user interactions with the AI chatbot.

**Attributes:**

* session\_id (Primary Key) - INT, auto-increment
* interaction\_id - INT
* user\_message - TEXT
* chatbot\_response - TEXT
* user\_id (Foreign Key) - INT, references User(user\_id)
* timestamp - DATETIME (records when the interaction occurred)
* User\_Feedback - ENUM('very\_helpful', 'helpful', 'neutral', 'unhelpful', 'very\_unhelpful') DEFAULT NULL

**Primary Key:**

* session\_id

**Foreign Key:**

* user\_id references User(user\_id)

**Notes:**

* timestamp: Essential for tracking interaction history and analyzing chatbot usage patterns.

## **11. Product**

**Purpose:** Manage product details for sales reporting.

**Attributes:**

* product\_id (Primary Key) - INT, auto-increment
* product\_name - VARCHAR(100)
* description - TEXT
* price - DECIMAL(10, 2)
* category - VARCHAR(100) (optional, categorizes products)
* stock\_quantity - INT (optional, tracks inventory levels)
* Inventory id INT foreign key inventory table

**Primary Key:**

* product\_id

**Notes:**

* stock\_quantity: Helps in managing inventory if integrated with sales and procurement.

## **12. Inventory**

Purpose: Manage inventory of raw materials and finished products.

**Attributes:**

* inventory\_id (Primary Key) - INT, auto-increment
* item\_name - VARCHAR(100)
* item\_type - ENUM('raw\_material', 'finished\_product')
* quantity - INT
* supplier\_id (Foreign Key) - INT, references Supplier(supplier\_id) (if implementing suppliers)

**Primary Key:**

* Inventory\_id

## **13. Supplier**

Purpose: Manage supplier details for inventory procurement.

**Attributes:**

* supplier\_id (Primary Key) - INT, auto-increment
* supplier\_name - VARCHAR(100)
* Email - VARCHAR(100)
* phone VARCHAR(100) ,
* address - VARCHAR(255)
* product\_supplied - TEXT (optional, lists products supplied)

**Primary Key:**

* supplier\_id

**Notes:**

* product\_supplied: Details the products or materials each supplier provides, useful for inventory management

**14. Roles Table**

Purpose: The roles table is introduced to manage user roles dynamically. This allows for easier management of roles within the system instead of relying on an enum in the User table. Roles can be updated, removed, or added as needed without requiring schema changes.

Attributes:

* role\_id (Primary Key) - INT, auto-increment
* role\_name - VARCHAR(100), unique (e.g., 'admin', 'manager', 'employee', 'maintenance\_staff', 'sales')

Primary Key:

* role\_id

