

# **Integrated Business Plus Employee Management System**

**CS F-212  
BITS PILANI**



**17th April, 2024**

**Prepared By:  
Group 23**

<b>Pratyaksh Kansal</b>	<b>2021B4A72783P</b>
<b>Suraj Pal</b>	<b>2021B2A72755P</b>
<b>Akshata Khandelwal</b>	<b>2021B4A71700P</b>
<b>Sayan Banerjee</b>	<b>2021B5A71516P</b>
<b>Vipul Singh</b>	<b>2020B2PS1283P</b>

## **Acknowledgements**

Our sincere appreciation goes out to Mr. Saurav SS , Mentor for DBS project , for his unflagging support, direction, and mentorship during this endeavor. This paper has been greatly influenced by his knowledge and insights.

We would like to express our sincere gratitude to the DBS team at BITS Pilani as a whole for giving us this important opportunity and facilitating a simple allocation and onboarding process. We are sincerely appreciative of the supportive atmosphere and tools provided to me.

# **Table of Contents**

## **1. Introduction**

### **1.1 Problem Statement**

### **1.2 Need**

## **2. Major Features of the Project**

## **3. E-R Model**

## **4. Relational Model**

## **5. Technical Details**

## **6. Conclusion**

## **7. Appendix**

# **1. Introduction**

## **1.1 Problem Statement:**

Many organizations struggle to manage their operations and employees due to the fragmented nature of their systems. The use of different systems for different projects leads to inefficiencies, errors and lack of planning in overall management. For example, one system may handle sales data while another manages employee data and payroll, creating duplicate data entry, inconsistencies, and report generation complications. This approach is chaotic, consumes valuable resources and time, hinders decision making and hinders organizational growth.

## **1.2 Need:**

These challenges are compounded by the complexity of today's industries and their growing workforce. Manual methods for tracking employee transactions, such as attendance and bookkeeping, are inaccurate, inflexible, and inadequate for tracking performance in different environments such as remote or flexible work schedules. A comprehensive plan to effectively integrate corporate management functions with personnel management solutions is urgently needed to ensure effectiveness addresses these issues. Such a system would provide the ability to accurately track labor hours, monitor performance and comply with labor laws, among other benefits.

## **1.2 Major Features of the Project**

### **1. Employee Information Management:**

- Capture and store comprehensive employee details including personal information, contact details, employment history, and qualifications.
- Allow for easy updating and maintenance of employee records, ensuring accuracy and compliance with legal requirements.

### **2. Attendance and Leave Management:**

- Implement a system for tracking employee attendance, including clock-in/clock-out times, breaks, and absences.
- Enable employees to request leaves, which are then processed, approved/rejected, and recorded in the system, ensuring accurate leave balances and compliance with company policies.

### **3. Project Management:**

- Create and manage multiple projects within the system, including project details, timelines, milestones, and resources allocation.
- Assign tasks to employees, set deadlines, and track task progress to ensure timely completion of project deliverables.
- Generate project reports and dashboards to provide insights into project status, resource utilization, and budget performance.

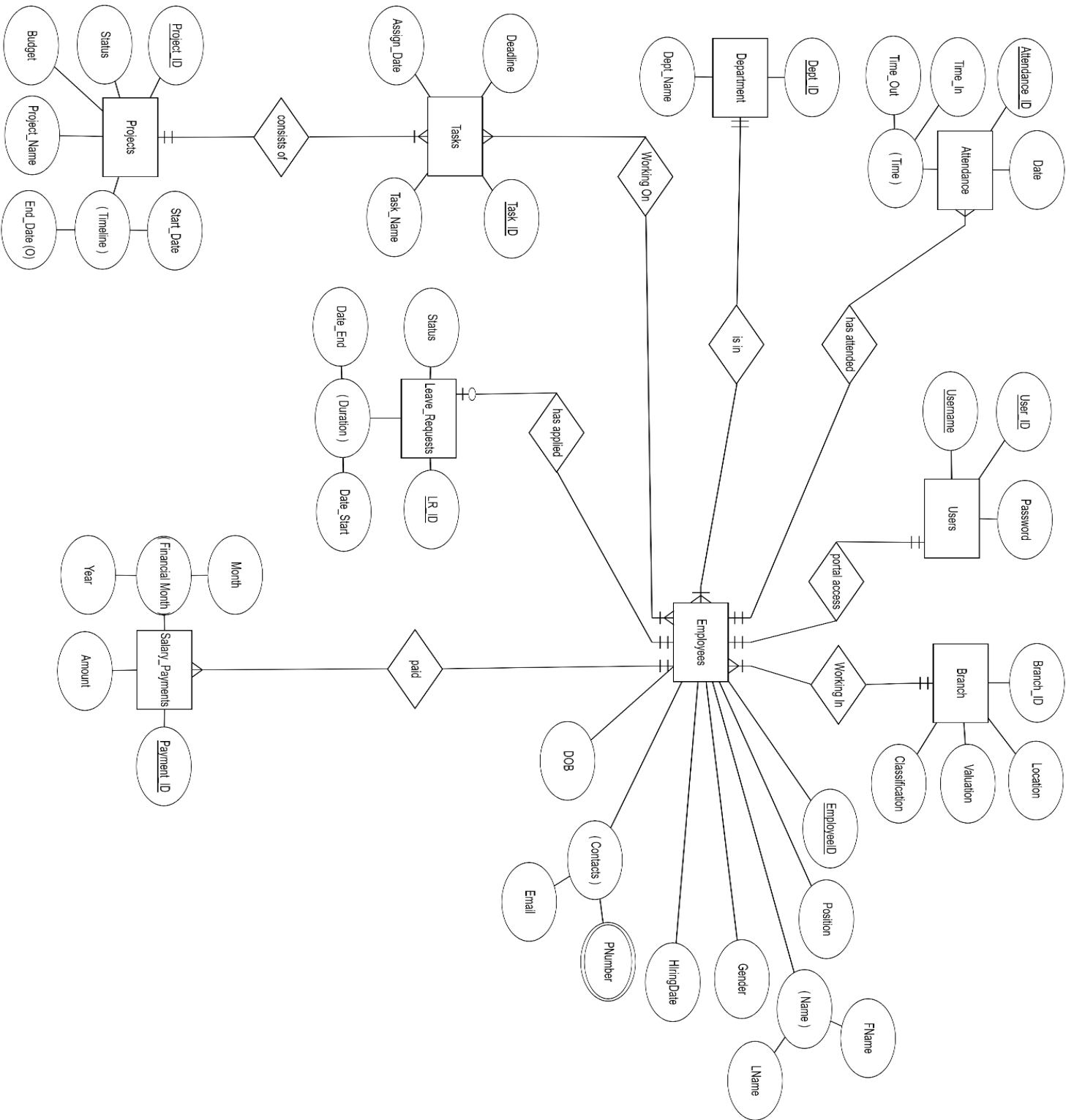
#### **4. Payroll Management:**

- Automate the calculation of employee salaries, bonuses, deductions, and taxes based on predefined rules and regulations.
- Generate pay stubs and reports for each pay period, facilitating transparent communication between the organization and its employees.

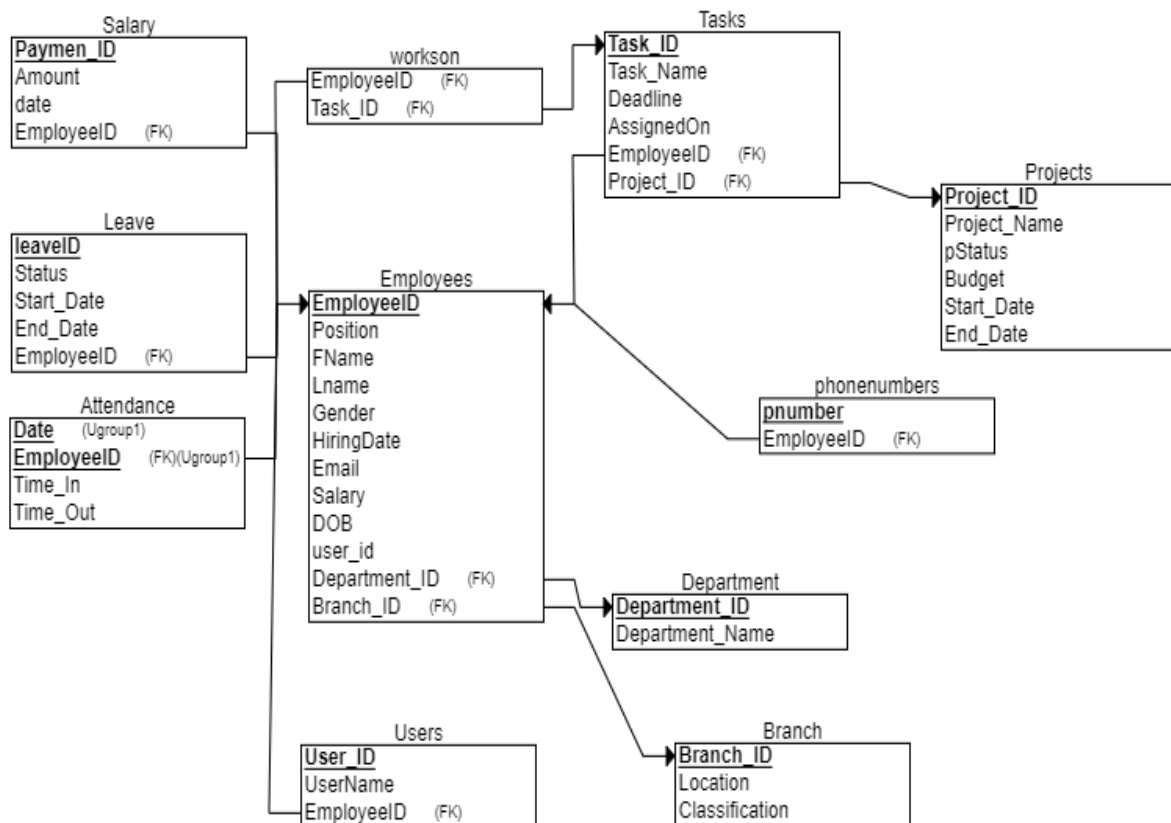
#### **5. Security and Access Control:**

- Implement robust security measures to protect sensitive employee and business data from unauthorized access or breaches.
- Define user roles and permissions to restrict access to specific functionalities and data based on job roles and responsibilities, ensuring data privacy and confidentiality.

## 1.3 E-R Model



## 5 Relational Model



### Explanation of the Relational Schema

#### 1. Department Table (department):

##### 1.1 Attributes:

1.1.1 deptID (Primary Key): Unique identifier for the department.

1.1.2 deptName: Name of the department.

Explanation: This table stores information about different departments within the organization.

#### 2. Branch Table (branch):

##### 2.1 Attributes:

2.1.1 branchID (Primary Key): Unique identifier for the branch.

2.1.2 location: Location of the branch.



2.1.3 classification: Classification of the branch (e.g., globalHQ, regionalHQ, etc.).  
Explanation: This table holds details about various branches of the organization.

### 3. Employee Table (employee):

#### 3.1 Attributes:

- 3.1.1 employeeID (Primary Key): Unique identifier for the employee.
- 3.1.2 fName: First name of the employee.
- 3.1.3 lName: Last name of the employee.
- 3.1.4 gender: Gender of the employee.
- 3.1.5 hiringDate: Date when the employee was hired.
- 3.1.6 email: Email address of the employee (unique).
- 3.1.7 salary: Salary of the employee (nullable).
- 3.1.8 DOB: Date of birth of the employee.
- 3.1.9 userID: User ID associated with the employee.
- 3.1.10 deptID: Foreign key referencing department table (nullable).
- 3.1.11 branchID: Foreign key referencing branch table (nullable).

Explanation: This table stores detailed information about employees, including their personal and professional details, along with references to the department and branch they belong to.

### 4. PhoneNumbers Table (phoneNumbers):

#### 4.1 Attributes:

- 4.1.1 pNumber (Primary Key): Phone number of the employee.
- 4.1.2 employeeID: Foreign key referencing employee table.

Explanation: This table maintains phone numbers associated with employees.

### 5. Users Table (users):

#### 5.1 Attributes:

- 5.1.1 userID (Primary Key): Unique identifier for the user.
- 5.1.2 userName: Username for login (unique).
- 5.1.3 hash: Hashed password for login.
- 5.1.4 salt: Salt used for hashing.
- 5.1.5 employeeID: Foreign key referencing employee table.

Explanation: This table manages user accounts for employees, storing login credentials.

### 6. Project Table (project):

#### 6.1 Attributes:

- 6.1.1 projectID (Primary Key): Unique identifier for the project.
- 6.1.2 projectName: Name of the project.
- 6.1.3 pstatus: Status of the project.
- 6.1.4 budget: Budget allocated for the project.
- 6.1.5 startDate: Start date of the project.
- 6.1.6 endDate: End date of the project (nullable).

Explanation: This table maintains information about various projects undertaken by the organization.

## 7. Task Table (task):

### 7.1 Attributes:

7.1.1 taskID (Primary Key): Unique identifier for the task (combination of project and employee ID).

7.1.1 taskName: Name of the task.

7.1.1 deadline: Deadline for the task.

7.1.1 assignedOn: Date and time when the task was assigned.

7.1.1 projectID: Foreign key referencing project table.

Explanation: This table stores details of tasks associated with projects.

## 8. WorksOn Table (worksOn):

### 8.1 Attributes:

8.1.1 taskID: Foreign key referencing task table.

8.1.2 employeeID: Foreign key referencing employee table.

Explanation: This table establishes a many-to-many relationship between employees and tasks they are working on.

## 9. Attendance Table (attendance):

### 9.1 Attributes:

9.1.1 employeeID: Foreign key referencing employee table.

9.1.2 date: Date of attendance.

9.1.3 intime: Time when the employee checked in (nullable).

9.1.4 outtime: Time when the employee checked out (nullable).

Explanation: This table tracks the attendance of employees on different dates.

## 10. Leaves Table (leaves):

### 10.1 Attributes:

10.1.1 leaveID (Primary Key): Unique identifier for the leave (combination of employee ID and date).

10.1.2 employeeID: Foreign key referencing employee table.

10.1.3 startDate: Start date of the leave.

10.1.4 endDate: End date of the leave.

10.1.5 status: Status of the leave request.

Explanation: This table manages leave requests submitted by employees.

## 11. Salary Table (salary):

### 11.1 Attributes:

11.1.1 paymentID (Primary Key): Unique identifier for the salary payment.

11.1.2 amount: Amount of the salary payment.

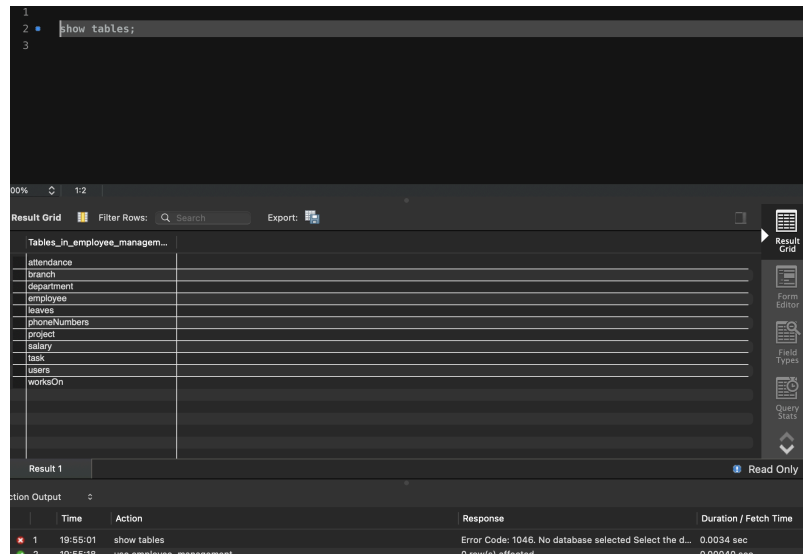
11.1.3 date: Date of the salary payment.

11.1.4 employeeID: Foreign key referencing employee table.

Explanation: This table stores information about salary payments made to employees.

## 6. Technical Details

**6.1 Tables:** Using the show tables we get all the tables present in the database.



In SQL, the DESCRIBE TABLE command (or DESC TABLE, DESCRIBE, or DESC) is used to specify the structure of a table in a database. Typically, that includes information about the columns (or fields) in the table, such as their names, data types, and any restrictions applied to them. The specific information provided by the DESCRIBE TABLE command includes:

**Column Name:** The name of each column in the table.

**Data Type:** The data type associated with each column, specifying the type of data that can be stored in that column (e.g., INTEGER, VARCHAR, DATE).

**Length/Precision:** For some data types (e.g. VARCHAR), the length or precision of data that can be stored in columns.

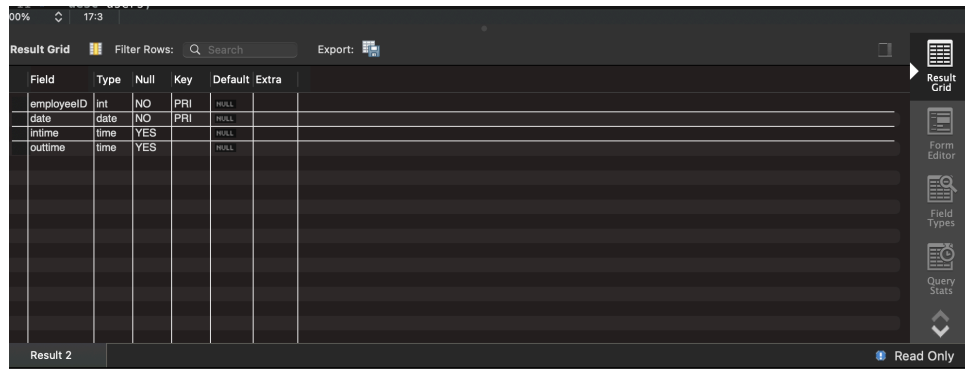
**Nullable:** Indicates whether the column allows NULL values (i.e., empty or nonexistent values).

**Default value:** If specified, a default value is assigned to the column when a new row is inserted and no value is assigned to the column.

**Constraint:** Any constraint applied to a column, such as a PRIMARY KEY, FOREIGN KEY, UNIQUE, or CHECK constraint

The above information for each table in the database has been added in form of screenshots.

## 6.1.1 Attendance table

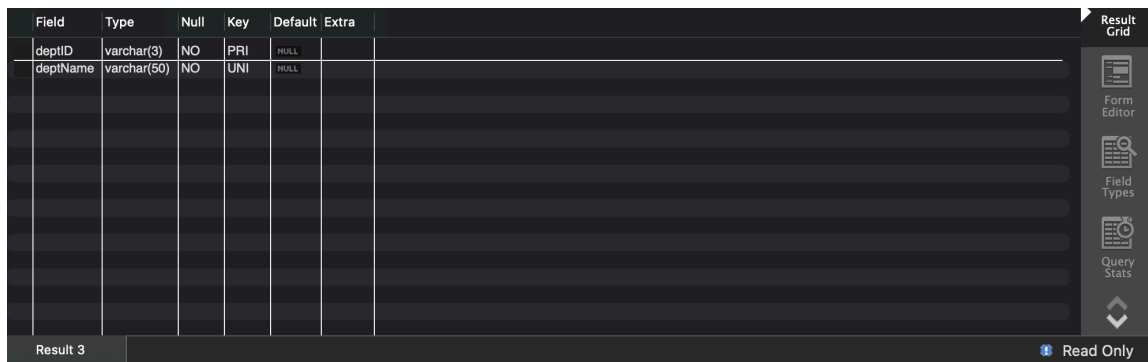


The screenshot shows the 'Result Grid' for the Attendance table. The table has the following structure:

Field	Type	Null	Key	Default	Extra
employeeID	int	NO	PRI	NULL	
date	date	NO	PRI	NULL	
inTime	time	YES		NULL	
outTime	time	YES		NULL	

The interface includes a 'Filter Rows' search bar, an 'Export' button, and a sidebar with 'Form Editor', 'Field Types', and 'Query Stats' options. The status bar at the bottom indicates 'Result 2' and 'Read Only'.

## 6.1.2 Department Table



The screenshot shows the 'Result Grid' for the Department Table. The table has the following structure:

Field	Type	Null	Key	Default	Extra
deptID	varchar(3)	NO	PRI	NULL	
deptName	varchar(50)	NO	UNI	NULL	

The interface includes a 'Filter Rows' search bar, an 'Export' button, and a sidebar with 'Form Editor', 'Field Types', and 'Query Stats' options. The status bar at the bottom indicates 'Result 3' and 'Read Only'.

## 6.1.3 Employee Table

100% 15:5

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
employeeID	int	NO	PRI	HULL	auto_increment
position	varchar(30)	NO		HULL	
firstName	varchar(50)	NO		HULL	
lastName	varchar(50)	NO		HULL	
gender	varchar(8)	NO		HULL	
hireDate	date	NO		HULL	
email	varchar(320)	NO	UNI	HULL	
salary	int	YES		HULL	
DOB	date	NO		HULL	
deptID	varchar(3)	YES	MUL	HULL	
branchID	int	YES	MUL	HULL	

Result 4 Read Only

## 6.1.4 Leave Table

100% 13:6

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
leaveID	varchar(10)	NO	PRI	HULL	
employeeID	int	NO	UNI	HULL	
startDate	date	NO		HULL	
endDate	date	NO		HULL	
status	varchar(8)	NO		Pending	

Result 5 Read Only

## 6.1.5 Phone Numbers Table

100% 19:7

Result Grid Filter Rows: Search Export:

Field	Type	Null	Key	Default	Extra
phoneNumber	varchar(20)	NO	PRI	HULL	
employeeID	int	NO	MUL	HULL	

Result 6 Read Only

## 6.1.6 Project Table



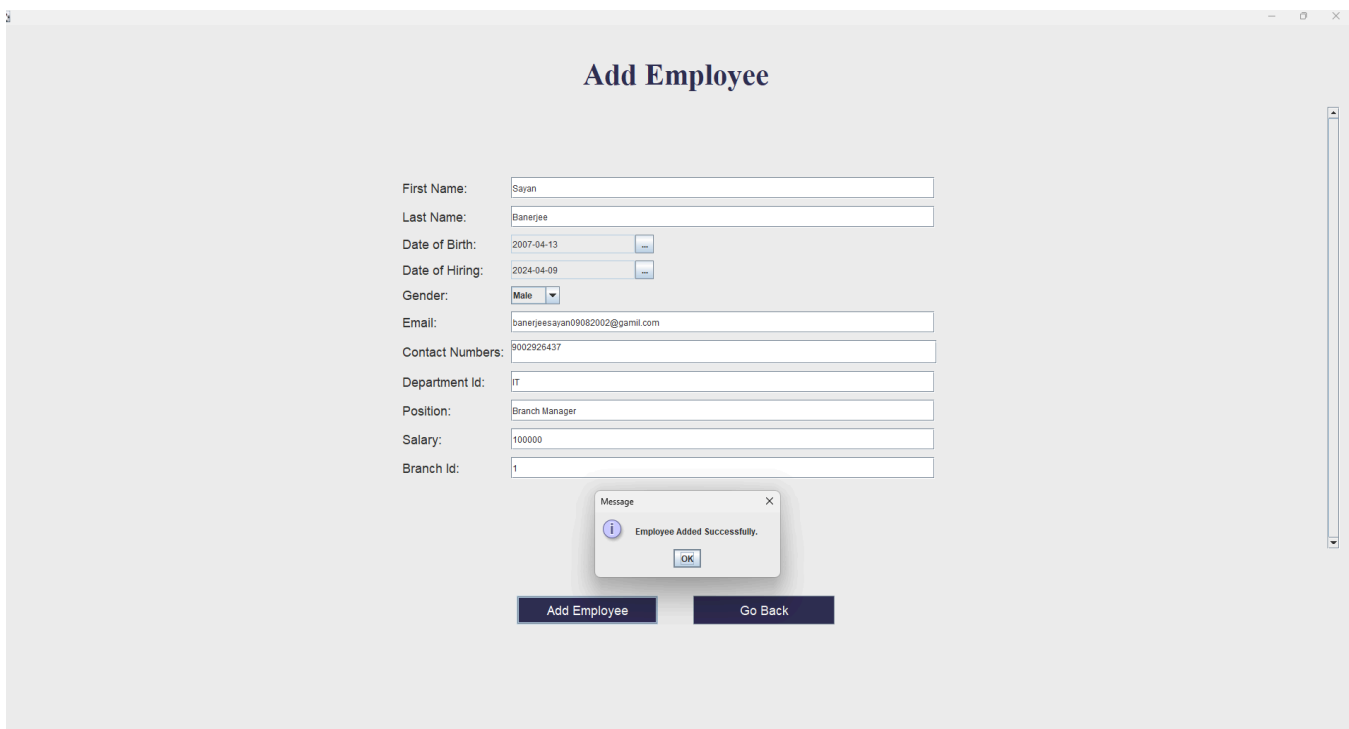
### 6.1.9 Tasks Table

[illegible]

## 6.2 CRUD Operations

**6.2.1 Insert Operations:** This operation is used to add new records (rows) to a table in the database. When performing an INSERT operation, you specify the table name and provide values for the columns in the table. These values can be constants, expressions, or results from other queries.

**6.2.1.1 Adding new Employee in the database: We are adding a new employee in the database**



The screenshot displays a web application window titled "Add Employee". The form contains the following fields and values:

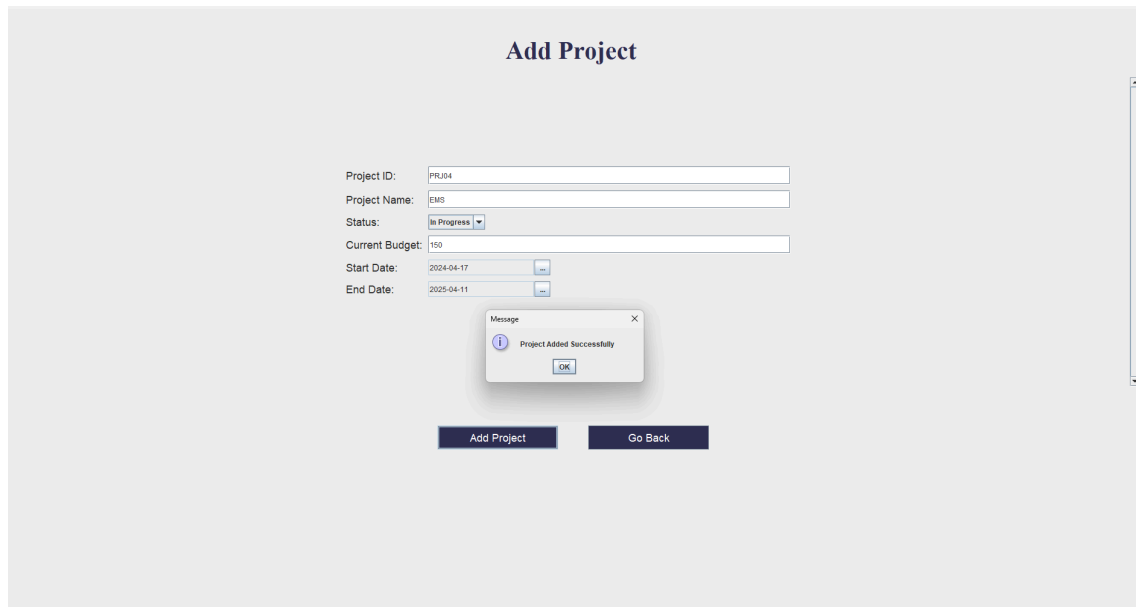
- First Name: Sayan
- Last Name: Banerjee
- Date of Birth: 2007-04-13
- Date of Hiring: 2024-04-09
- Gender: Male (selected from a dropdown)
- Email: banerjeesayan09082002@gmail.com
- Contact Numbers: 9002926437
- Department Id: IT
- Position: Branch Manager
- Salary: 100000
- Branch Id: 1

A modal dialog box titled "Message" is centered on the screen, displaying an information icon and the text "Employee Added Successfully." with an "OK" button.

At the bottom of the form, there are two buttons: "Add Employee" and "Go Back".



### 6.2.1.1 Adding new Project in the database: We are adding a new Project in the database:



The screenshot displays a web application interface titled "Add Project". The form contains the following fields and values:

- Project ID: PRJ04
- Project Name: EMS
- Status: In Progress (dropdown menu)
- Current Budget: 150
- Start Date: 2024-04-17
- End Date: 2025-04-11

A modal message box is centered on the screen, displaying the text "Project Added Successfully" with an "OK" button. At the bottom of the form, there are two buttons: "Add Project" and "Go Back".

**6.2.2 Search Operation:** SELECT operation is used to retrieve data from one or more tables in the database. When executing a SELECT query, you specify the columns you want to retrieve, the table(s) from which to retrieve the data, and optionally provide conditions, sorting, and grouping clauses

#### 6.2.2.1 Searching Employee in the Database:

## SEARCH/MANAGE EMPLOYEE

Select Date:

Employee ID:

Search

Present On

Present In

Assign Task

EmployeeID: 1

Position: Admin

Name: John Doe

Gender: Male

HiringDate: 2023-01-15

DOB: 1990-05-20

Email: john@example.com

Salary: 80000

DeptID: IT

BranchID: 1

### 6.2.2.3 Searching Project in the Database:

## MANAGE PROJECTS

Project Name:

Search

Edit Project

Add New Project

Back

Project ID: PRJ01

Project Name: Website Redesign

Status: In Progress

Budget: 50000

Start Date: 2023-04-01

End Date: null

**6.2.3 Update Operation:** UPDATE operation is used to modify existing records in a table. When performing an UPDATE operation, you specify the table name, set the new values for the columns you want to update, and optionally provide a condition to specify which records to

update.

### 6.2.3.1 Assigning new Tasks

### Assign Task

Employee ID:

3

Task Name:

DBS Presentation

Date of Deadline:

2024-04-30

...

Project ID:

PRJ02

Message

Task assigned successfully.

OK

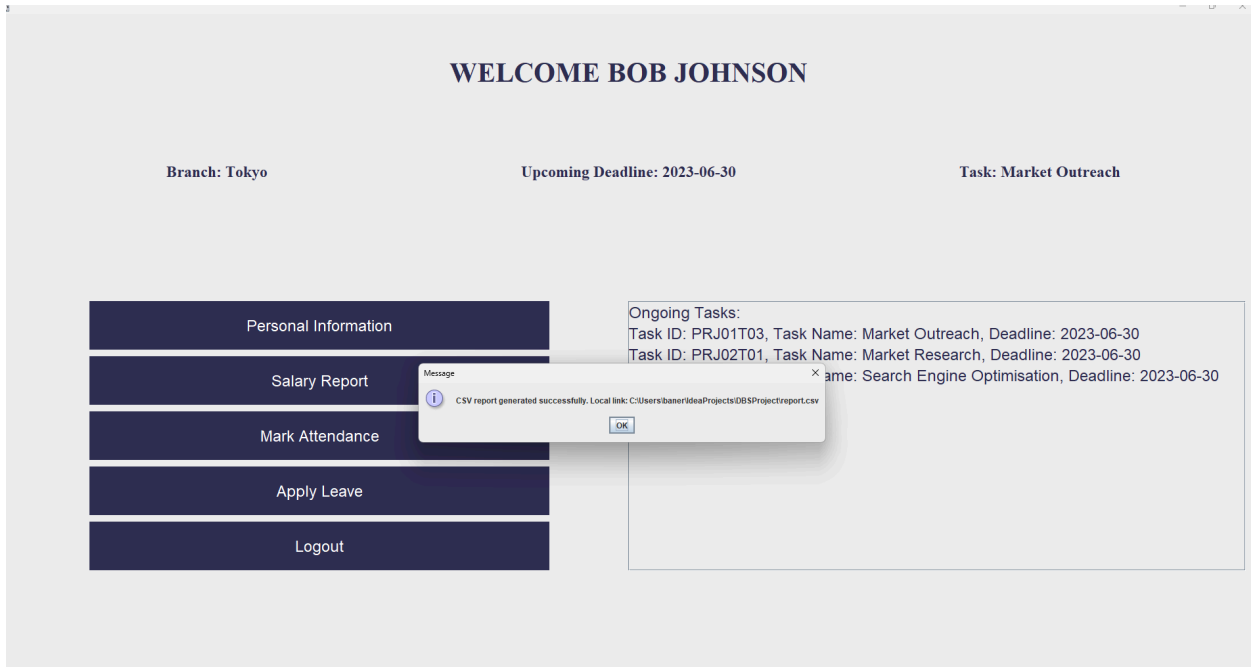
Add Employee

Go Back

# 6.3 Report Samples

We are generating system report the salary received by the employee during a particular year.

Below is the UI for the command to generate the salary report.



Below is the report generated in the .csv format

	A	B	C	D	E
1	Employee ID	First Name	Last Name	Salary Amount	Salary Date
2	3	Bob	Johnson	4000	2023-04-30
3	3	Bob	Johnson	4000	2023-05-30
4	3	Bob	Johnson	4000	2023-06-30

# Complex Queries

1. List employees who have recorded attendance for a particular date: Below is the query written in the sql:-

```
SELECT e.employeeID, e.fName, e.lName  
  
FROM employee e  
  
LEFT JOIN attendance a ON e.employeeID = a.employeeID  
  
WHERE a.date = '2024-04-10' AND a.employeeID IS NULL;
```

Below is the output for the required query:-

### SEARCH/MANAGE EMPLOYEE

Select Date: 2023-04-01

Employee ID:

Search

Present On

Present In

Assign Task

Following were present on selected day:

Employee ID - Name

1 - John Doe

2 - Alice Smith

## 2.Retrieve employees who have attended work on all working days of a specific month:

Below is the query written in sql:

```
SELECT a.employeeID, e.fName, e.lName, COUNT(DISTINCT date) AS  
attended_days  
FROM attendance a  
INNER JOIN employee e ON a.employeeID = e.employeeID  
WHERE MONTH(a.date) = 4 AND YEAR(a.date) = 2024  
GROUP BY a.employeeID, e.fName, e.lName  
HAVING COUNT(DISTINCT date) = DAY(LAST_DAY('2024-04-01'));
```

Below is the result for the query:

Following were present on selected day:

Employee ID – Name

10 – John Doe

---

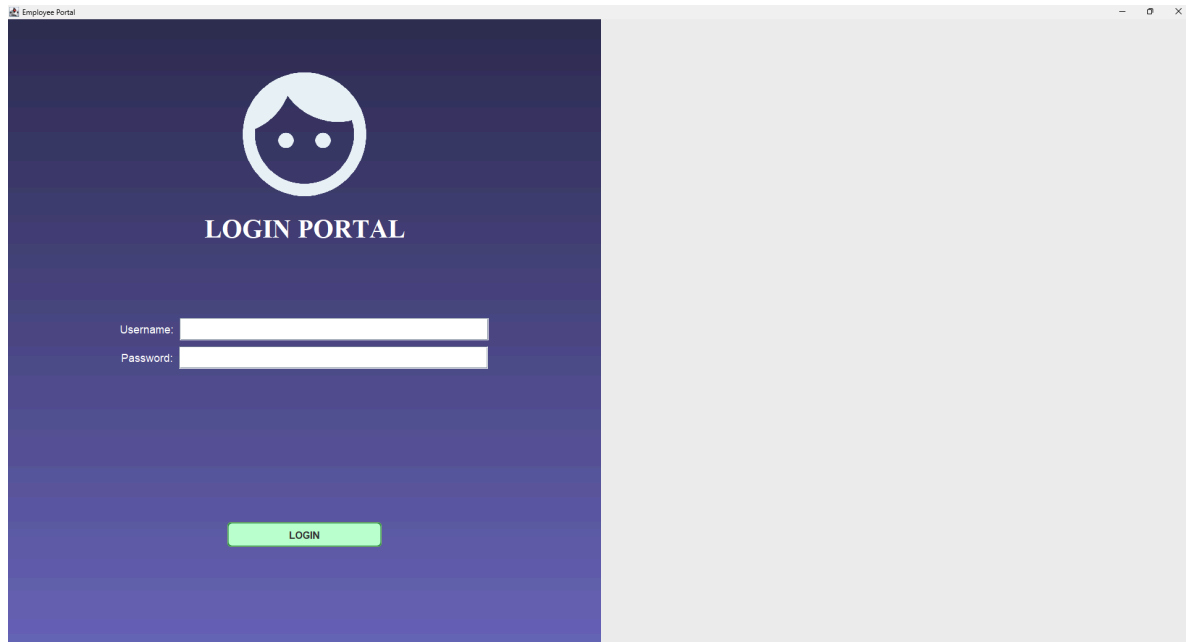
## 6. Conclusion

In conclusion, the development of an integrated Business Plus personnel management system represents an important step towards addressing the complex challenges that organizations face in managing their operations and human resources. By integrating various applications into an integration platform, this project aims to simplify the process, improve the accuracy of the data, and increase organizational productivity. Through features such as personnel information management, payroll implementation, and by implementing business analytics, the system facilitates better decision-making and resource allocation.

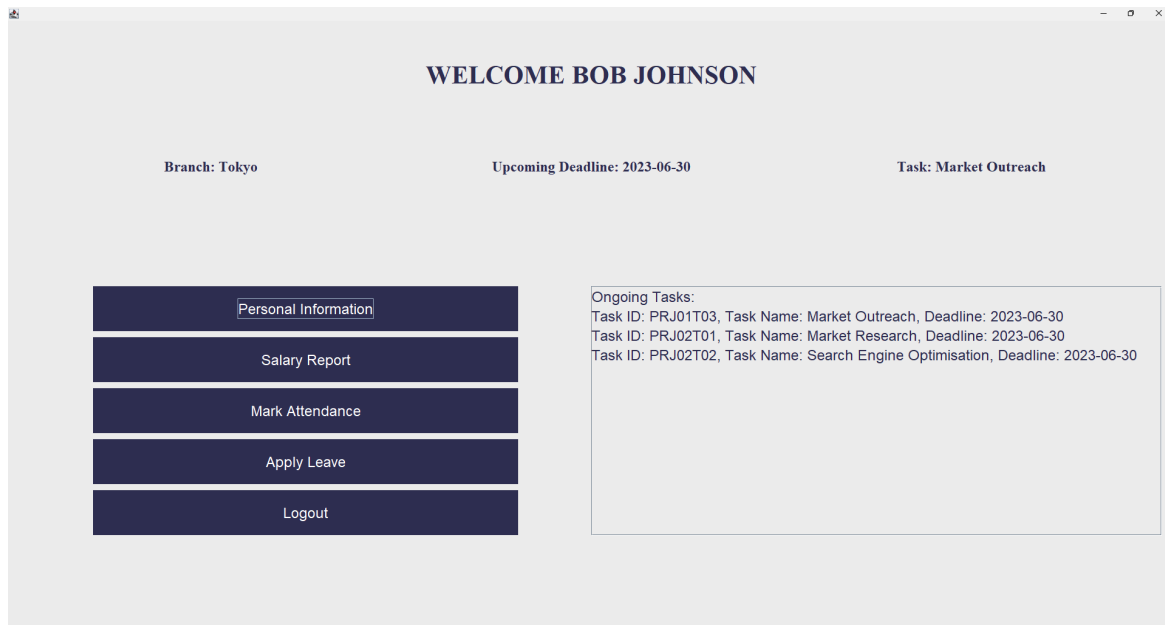
Additionally, the successful implementation of the framework provides valuable insights into real-world database management practices, including database design, data modeling, query optimization through hands-on experience students develop and design databases, using collaboration internal understanding is used, and user networks are built. Gain in-depth understanding of database systems and their role in supporting organizational goals. Overall, this project is not only a learning opportunity for students but for organizations seeking to increase their operational efficiency and competitiveness in today's dynamic business environment.

# Appendix

## Login Page

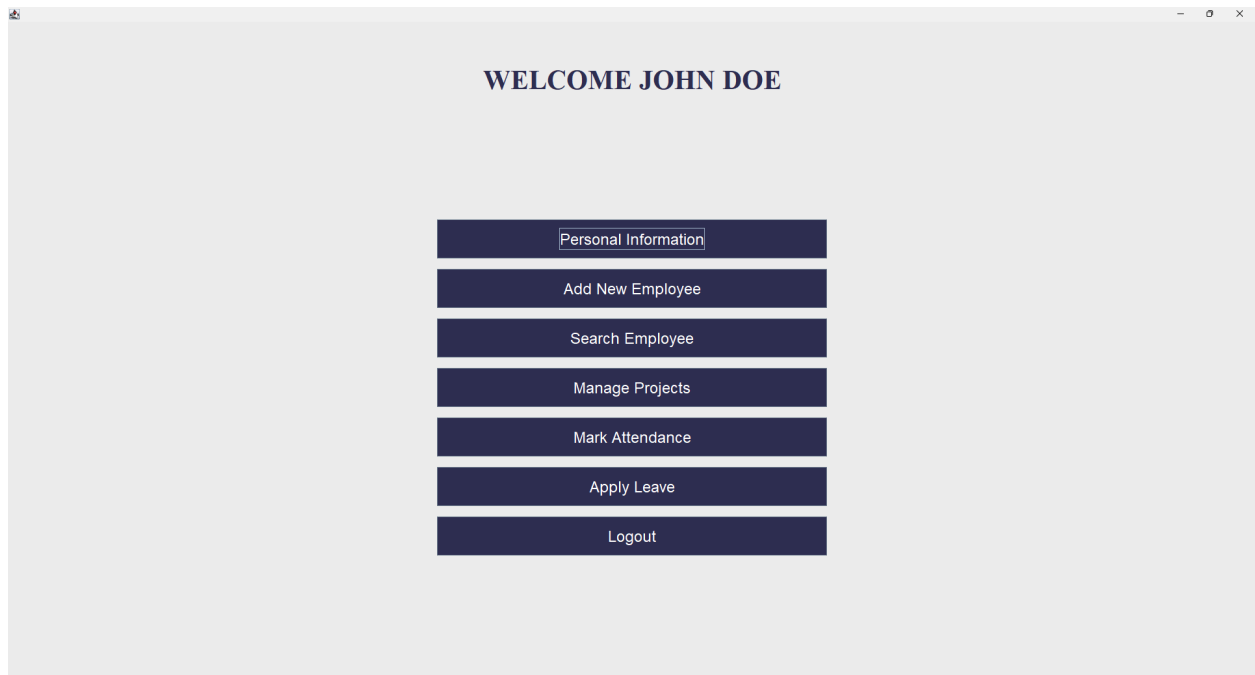


## Employee Page





# Admin Page



# Leave Creator Page

