CHAPTER 1

# INTRODUCTION

#### 1.1 INTRODUCTION TO DBMS

A database is simply an organized collection of related data, typically stored on disk, and accessible by many concurrent users, it is a logically coherent collection of data with some inherent meaning, representing some aspect of the real world and which is designed, built and populated with data for a specific purpose.

Databases are managed by a Database Management System (DBMS) which is a collection of programs that enables users to create and maintain a database.

Advantages of DBMS:

1. Redundancy is controlled.

2. Unauthorized access is restricted.

3. Providing multiple user interfaces.

4. Enforcing integrity constraints.

5. Providing backup and recovery.

#### 1.2 INTRODUCTION TO SQL

Structured Query Language (SQL), is a language used to request data from a database which includes database creation, deletion, retrieval of required tables and even manipulation of data held in a relational database management system.

SQL is considered as a Non-Procedural or a High-level language in which the expected result or operation is given without the specific details about how to accomplish the task. So, SQL is a declarative language.

Therefore, SQL is designed at a higher conceptual level of operation than procedural languages as procedural languages include only the information about opening and closing tables, loading and searching indexes, or flushing buffers and writing data to file systems, but the lower level logical and physical operations are not specified in SQL.

**1.3 PAYROLL MANAGEMENT SYSTEM**

A payroll management system describes the specialist software that can empower companies by streamlining and automatically carrying out the processes involved in payroll.

That is:

1. To create a database of the payroll of a company.
2. To search the records of employee or manager.
3. To check the salary and particular department of the employee.
4. To add and delete the employee in a table.
5. Store procedure used in the database to detect the salary depending on the leave taken by employee.

**1.4 SCOPE AND IMPORTANCE OF WORK\**

**Accuracy and Efficiency**: Payroll systems automate complex calculations for salaries, taxes, and

deductions, minimizing errors and saving time compared to manual processing.

**Compliance**: These systems ensure adherence to ever-changing tax regulations and reporting requirements, reducing the risk of penalties and legal issues.

**Employee Satisfaction**: Timely and accurate payments are crucial for employee morale. A good payroll system ensures employees receive their pay correctly and on time.

**Data Management:** and Reporting: Payroll systems centralize employee data and generate reports for accounting, budgeting, and HR purposes, streamlining financial processes.

CHAPTER 2

# DESIGN

The Entity–Relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram).** An **Entity Relationship Diagram (ERD)** shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data.

An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of database. ER diagrams are used to sketch out the design of a database.

### Entities

An entity is an 'object' in the real world with an independent existence and an entity type defines a collection (or set) of entities that have the same attributes. Each entity type in the database is described by its name and attributes. An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name.

### Relationships

A relationship among two or more entities represents an association among the entities and whenever an attribute of one entity refers to another entity, there exists a relationship between the two entities.

In a relationship, a foreign key of one table refers to the primary key of the other table and it is represented by a diamond shape in the ER diagram.

### Attributes

An attribute represents some property of interest that further describes an entity and the column header of the table shows the attributes. Each attribute in a table has a certain domain which allows it to accept a certain ‘set of values’ only.

The attribute values of each entity will define its characteristics in the table and is represented by oval in the ER diagram

### 2.1 DATABASE SCHEMA

1. Employee (emp\_id, Fname, Lname,Joining\_date, City, State, Salary, Dept\_id)
2. Department (Dept\_id, Dept\_name)
3. Salary (Salary\_id, Gross\_salary, Acc\_id)
4. Project (Dept\_id, Proj\_id, proj\_name, Proj\_desc)
5. Attendance (Emp\_id ,Atten\_id, Days\_worked)
6. Register(uname1,email,pswd1,pswd2)

### 2.2 ER DIAGRAM

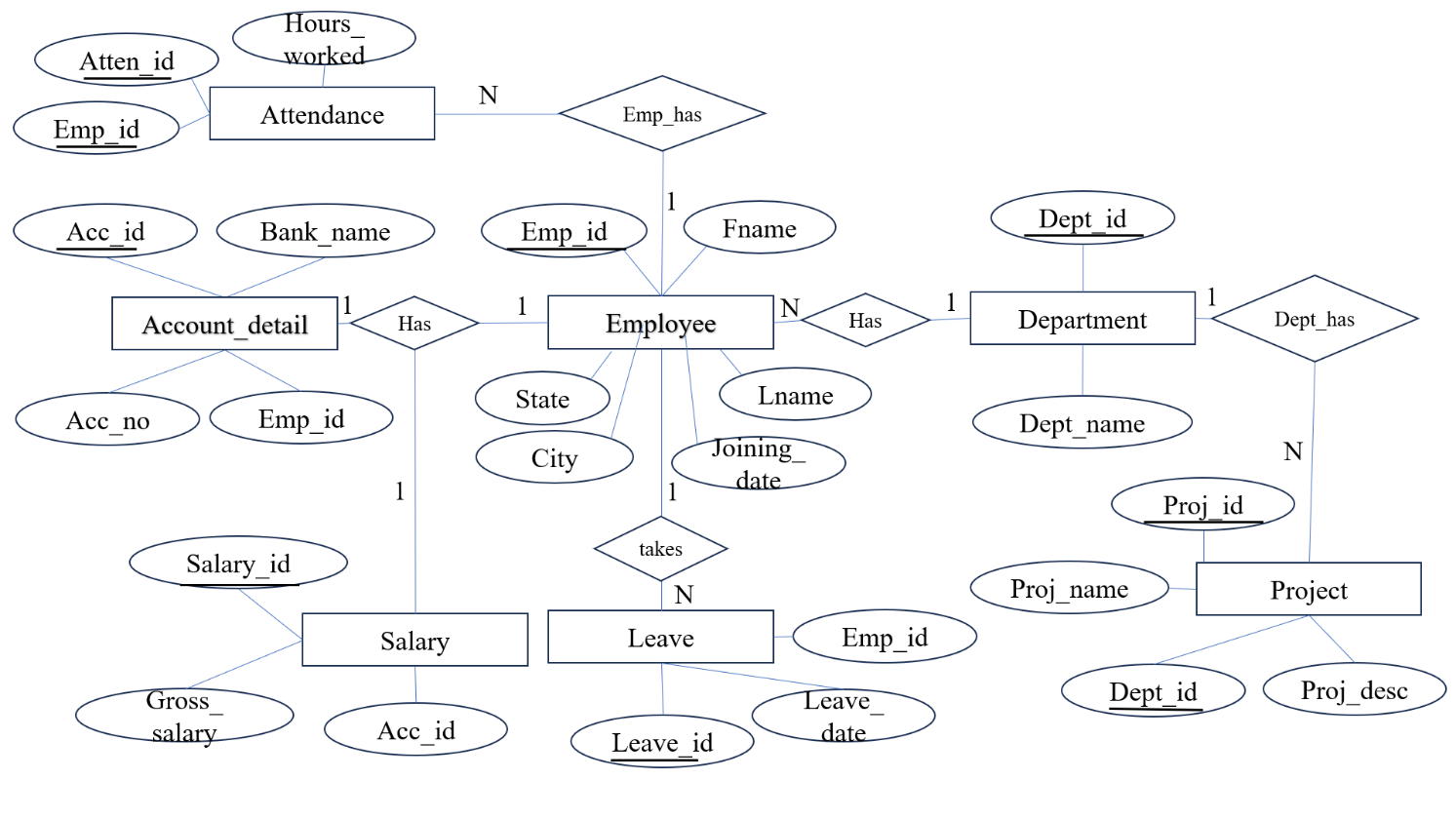
****

Figure 2.2.1: E R diagram for the database

**2.3 SCHEMA DIAGRAM**

****

### Figure 2.3.1: Schema diagram

CHAPTER 3

# IMPLEMENTATION

**3.1 LIST OF TABLES**

* **Register**
* **Employee**
* **Salary**
* **Department**
* **Attendance**
* **Leave**

#### 3.2 CREATION OF RELATIONS

**Register**

CREATE TABLE `register` (

  `uname1` varchar(20) NOT NULL,

  `email` varchar(25) NOT NULL,

  `upswd1` varchar(20) NOT NULL,

  `upswd2` varchar(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `register` (`uname1`, `email`, `upswd1`, `upswd2`) VALUES

('sathya', 'sathya@gmail.com', '123', '123'),

('shiva', 'shiva@gmail.com', 'shiva', 'shiva');

**Employee**

CREATE TABLE `employee` (

  `emp\_id` int(11) NOT NULL,

  `fname` varchar(50) NOT NULL,

  `lname` varchar(50) NOT NULL,

  `joining\_date` date NOT NULL,

  `city` varchar(50) NOT NULL,

  `states` varchar(50) NOT NULL,

  `salary` int(11) NOT NULL,

  `dept\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `employee` (`emp\_id`, `fname`, `lname`, `joining\_date`, `city`, `states`, `salary`, `dept\_id`) VALUES

(102, 'Vrushali', 'Patil', '2011-06-18', 'Boston', 'Massachusetts', 20000, 1),

(103, 'Pratik', 'Parija', '2013-09-19', 'Chicago', 'Illinois', 50000, 2),

(104, 'Chetan', 'Mistry', '2012-04-11', 'Miami', 'Florida', 60000, 3),

(105, 'Anugraha', 'Varkey', '2016-08-17', 'Atlanta', 'Georgia', 70000, 4),

(123, 'shiva', 'subramaniyam s', '2024-02-22', 'bangalore', 'Karnataka', 20000, 5),

(124, 'sathya', 'priya', '2024-02-16', 'bangalore', 'Karnataka', 50000, 1),

(125, 'shashank', 'b dev', '2024-02-23', 'bangalore', 'Karnataka', 50000, 3);

**Salary**

CREATE TABLE `salary` (

  `salary\_id` int(11) NOT NULL,

  `gross\_salary` int(11) NOT NULL,

  `acc\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `salary` (`salary\_id`, `gross\_salary`, `acc\_id`) VALUES

(11, 57600, 40),

(12, 76800, 40),

(13, 96000, 41),

(14, 115200, 43),

(15, 57600, 41);

**Department**

CREATE TABLE `department` (

  `dept\_id` int(10) NOT NULL,

  `dept\_name` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `department` (`dept\_id`, `dept\_name`) VALUES

(1, 'iot'),

(2, 'Software Development'),

(3, 'Data Analysis'),

(4, 'Data Science'),

(5, 'Business Intelligence'),

(16, 'Accounts');

**Attendance**

CREATE TABLE `attendance` (

  `emp\_id` int(10) NOT NULL,

  `atten\_id` int(10) NOT NULL,

  `hours\_worked` int(10) NOT NULL,

  `days\_worked` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `attendance` (`emp\_id`, `atten\_id`, `hours\_worked`, `days\_worked`) VALUES

(102, 93, 9, 20),

(103, 94, 10, 19),

(104, 95, 11, 15),

(105, 96, 12, 11),

(105, 97, 14, 10);

CHAPTER 4

#### RESULTS AND SNAPSHOTS

**4.1 SNAPSHOT OF THE RELATIONS IN DATABASE**

**Employee Info**

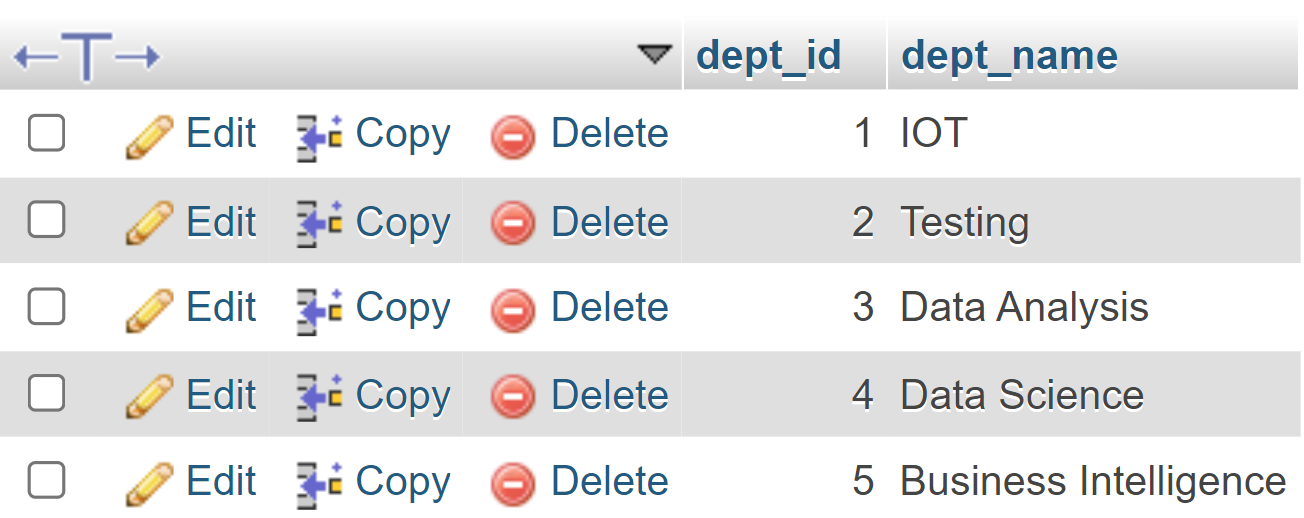
EMPLOYEE INFO table has attributes emp\_id, Fname, Lname,Joining\_date, City, State, Salary, Dept\_id as shown in the snapshot below.

STRUCTURE OF EMPLOYEE INFO

****

**Department Info**

DEPARTMENT LIST table has Dept\_id, Dept\_name as shown in table

STRUCTURE OF DEPARTMENT LIST****

**Salary Info**

SALARY TABLE has the attribute Salary\_id, Gross\_salary, Acc\_id is used shown in the table

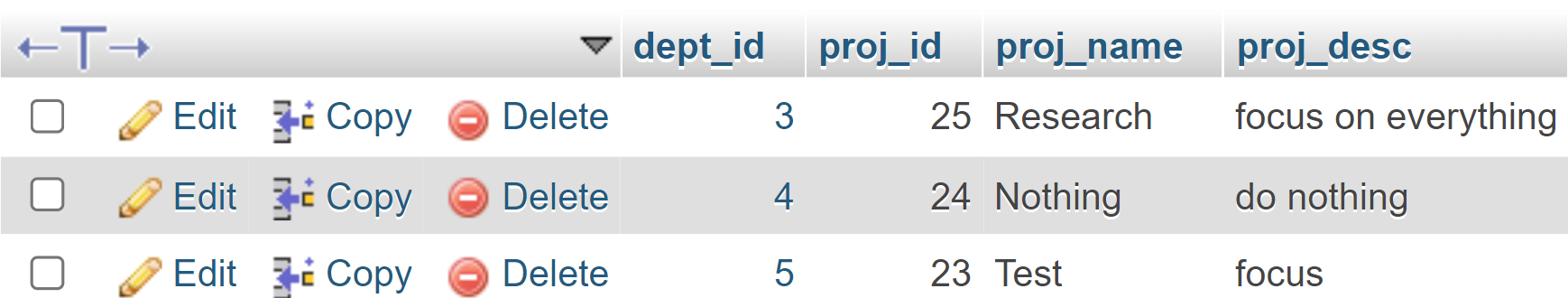
STRUCTURE OF SALARY TABLE



**Project Info**

PROJECT table has the attribute Dept\_id, Proj\_id, proj\_name, Proj\_desc as shown in Table

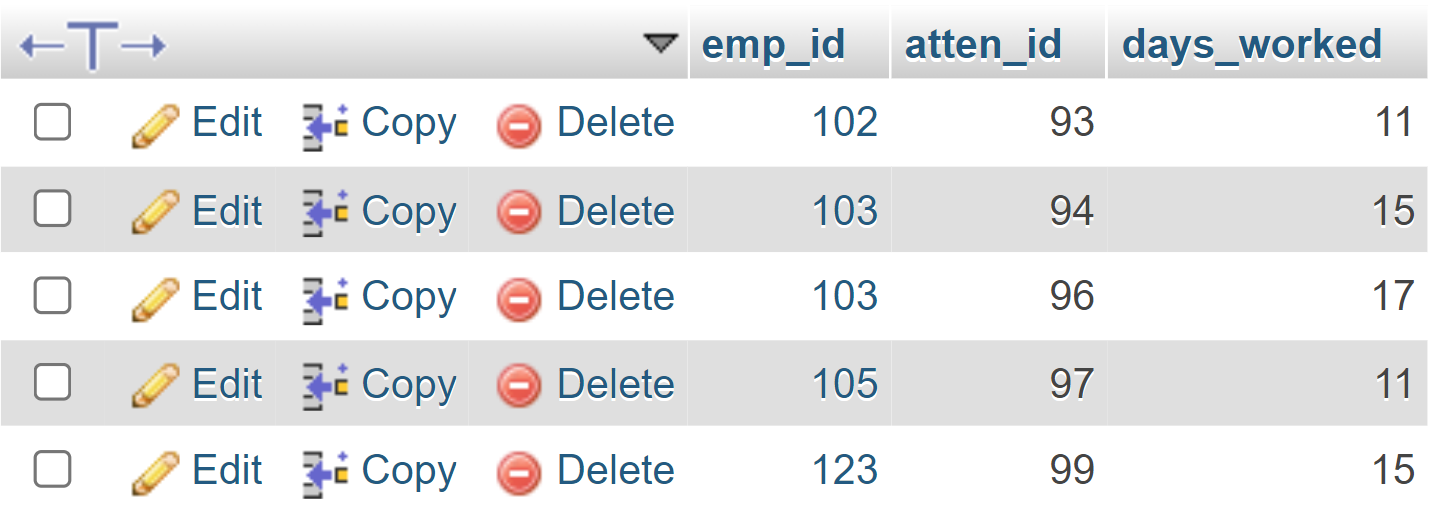
STRUCTURE OF PROJECT LIST



**Attendance Info**

Attendance table has the attribute Emp\_id ,Atten\_id, Days\_worked as shown in table

STRUCTURE OF ATTENDANCE



**4.2 CODE SNIPPETS**

**1. DATABASE CONNECTION**

The connect() / mysqli\_connect() function opens a new connection to the MySQL server with the following syntax : mysqli\_connect(host, username, password, dbname, port, socket).

<?php

   $host = "localhost";

$dbusername = "root";

$dbpassword = "";

$dbname = "project1";

    $conn = new mysqli($host, $dbusername, $dbpassword, $dbname);

    if($conn->connect\_error){

        die("Connection failed".$conn->connect\_error);

    }

    echo " ";

    ?>

**2. INSERT QUERY**

This query is used to insert an employee.

INSERT INTO `employee` (`emp\_id`, `fname`, `lname`, `joining\_date`, `city`, `states`, `salary`, `dept\_id`) VALUES

(102, 'Vrushali', 'Patil', '2011-06-18', 'Boston', 'Massachusetts', 20000, 1),

(103, 'Pratik', 'Parija', '2013-09-19', 'Chicago', 'Illinois', 50000, 2),

(104, 'Chetan', 'Mistry', '2012-04-11', 'Miami', 'Florida', 60000, 3),

(105, 'Anugraha', 'Varkey', '2016-08-17', 'Atlanta', 'Georgia', 70000, 4),

(123, 'shiva', 'subramaniyam s', '2024-02-22', 'bangalore', 'Karnataka', 20000, 5),

(124, 'sathya', 'priya', '2024-02-16', 'bangalore', 'Karnataka', 50000, 1),

(125, 'shashank', 'b dev', '2024-02-23', 'bangalore', 'Karnataka', 50000, 3);

**3. SELECT QUERY**

In this query, all the details are fetched using SELECT\* command.

SELECT \* FROM employee;

**4. ALTER QUERY**

Here the alter query is called to update the patient of an already existing based on its name and and patient id respectively.

UPDATE employee SET salary = new\_salary WHERE emp\_id = employee\_id;

**5. STORED PROCEDURE**

DELIMITER //

CREATE PROCEDURE UpdateSalary(IN salary INT)

BEGIN

    UPDATE employee AS e

    JOIN attendance AS a ON e.emp\_id = a.emp\_id

    SET e.salary =

        CASE

            WHEN a.days\_worked BETWEEN 10 AND 12 THEN e.salary / 3.1

            WHEN a.days\_worked BETWEEN 13 AND 15 THEN e.salary / 2.1

            WHEN a.days\_worked BETWEEN 16 AND 20 THEN e.salary / 1.1

            ELSE e.salary

        END;

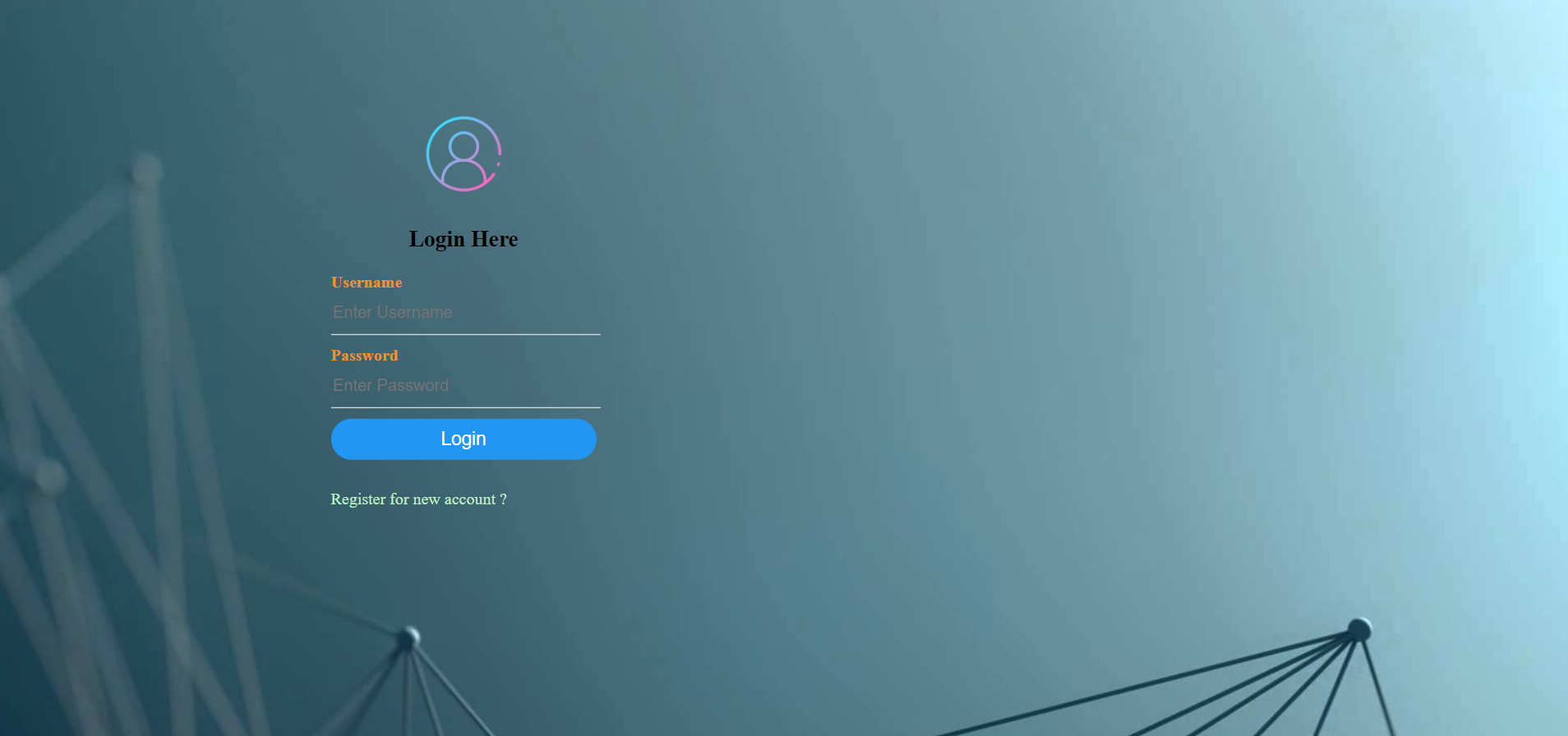
END //

DELIMITER ;

### 4.3 FRONTEND SNAPSHOTS

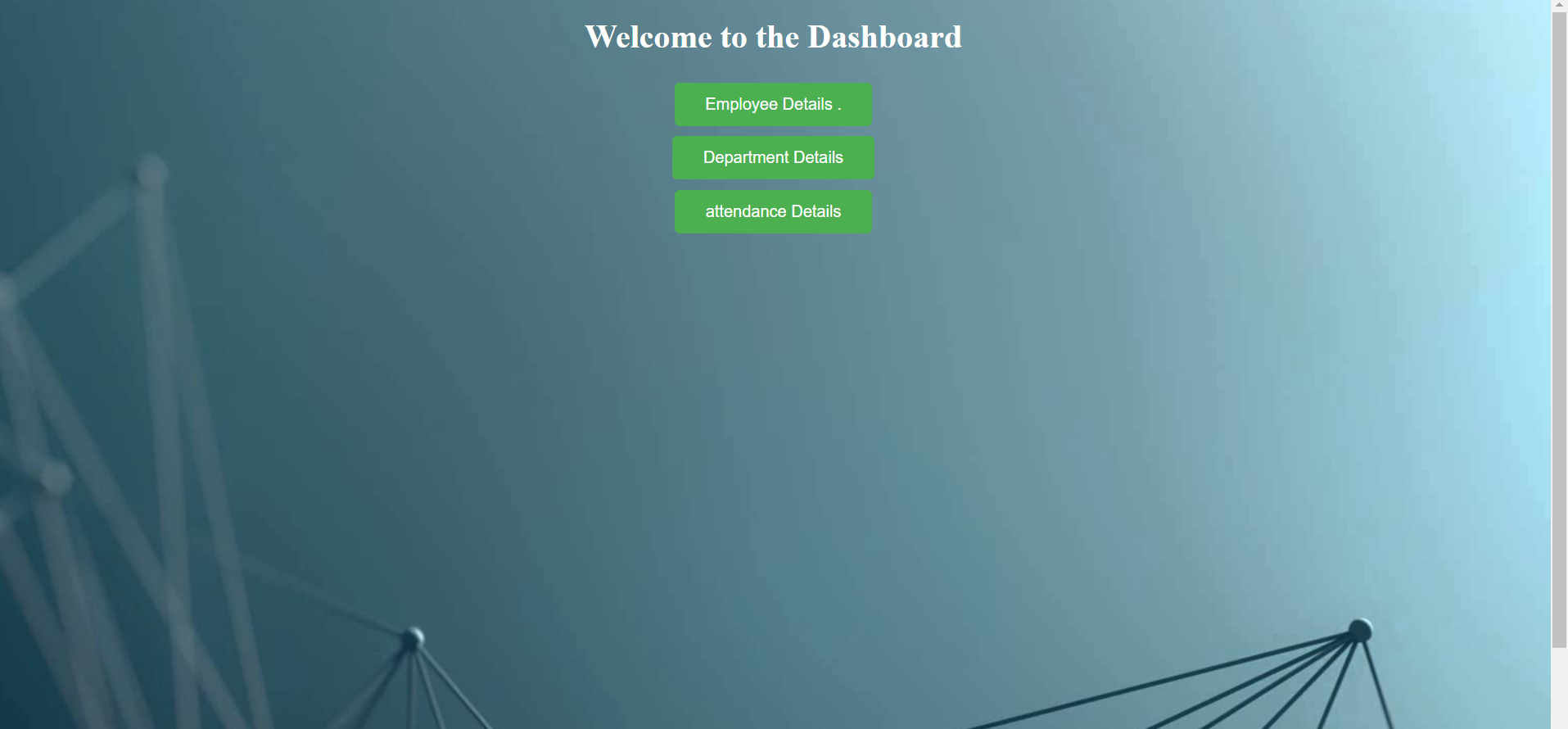
### HOME PAGE

### This is the first window when the application is executed.



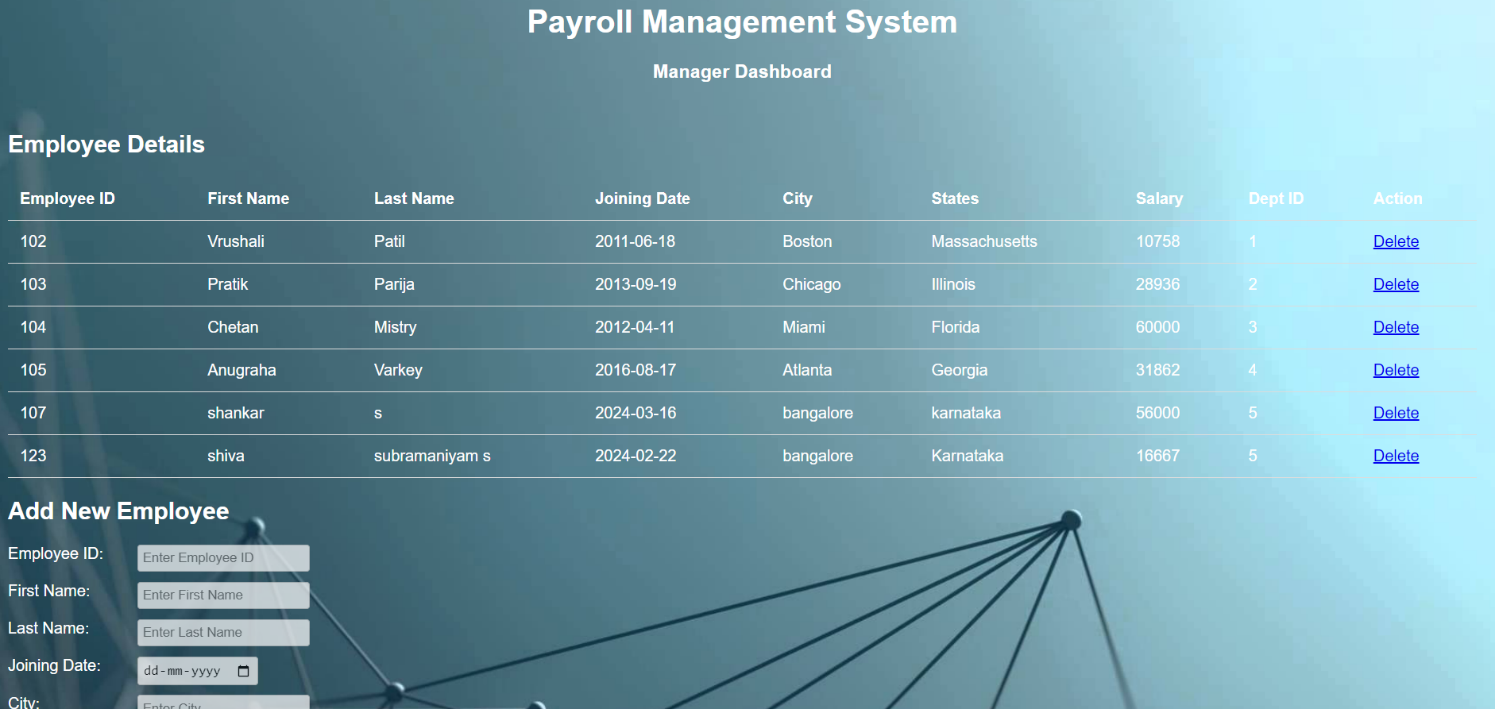
### ADMIN PAGE DASHBOARD

This page shows the different detail pages



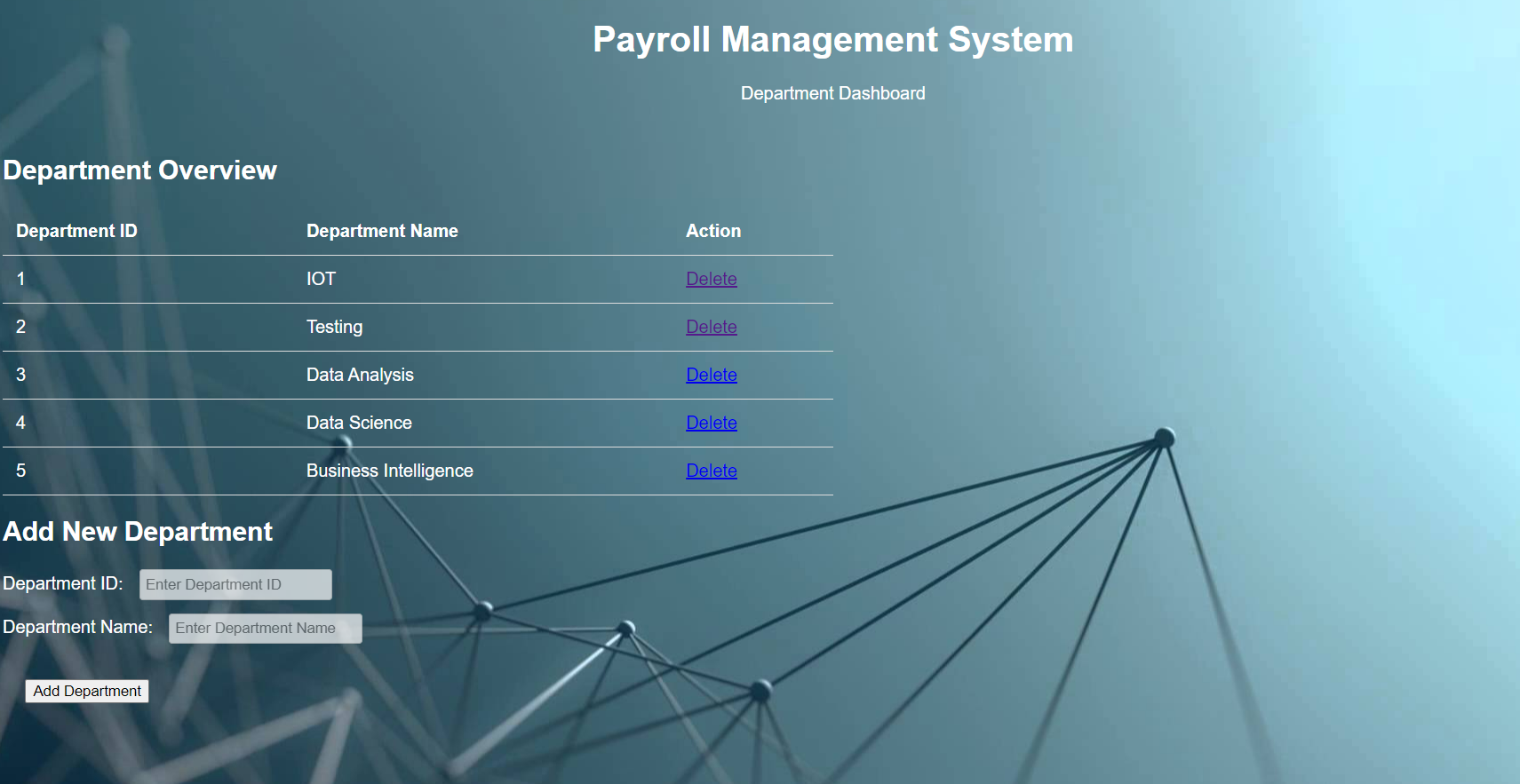
### ADMIN FIND EMPLOYEE RECORD

This page allows admin to add or delete a employee details



### ADMIN FIND DEPARTMENT RECORD

This page allows admin to add or delete a department details.



**ADMIN FIND ATTENDANCE RECORD**

This page allows admin to Update attendance of the employee.

### 

CHAPTER 5

#### CONCLUSION

Payroll Management System (PMS) acts as a vital tool for organizations to streamline and organize employee data. It centralizes the process of storing, retrieving, and analyzing employee payroll information, significantly improving the efficiency and accuracy of payroll operations. A PMS typically offers features like employee record management, timesheet tracking, salary calculation and disbursement, tax withholding and deductions, and comprehensive reporting tools.

Implementing a PMS also enhances data security and privacy by enforcing strict access controls and data encryption. This ensures employee payroll information remains confidential and protected. By leveraging a PMS, organizations can achieve better management and utilization of payroll data, leading to reduced errors, minimized costs, and improved compliance with tax regulations. Overall, a Payroll Management System proves to be a valuable asset for any organization seeking to optimize its payroll processes and ensure timely and accurate employee compensation.

### REFERENCES

1. MySQL Database <https://www.mysql.com/downloads/>
2. PHP

<http://php.net/>

1. ChatGPT, Gemini.