Case Study: Simple Employee Management System

Project Structure

```
employee-management/
|── package.json
|── app.js
|── config/
|   └── db.js
|── models/
|   └── employeeModel.js
|── services/
|   └── employeeService.js
```

Step 1: Database Setup (MySQL)

Run this in MySQL Workbench / CLI:

```
CREATE DATABASE employeeDB;
USE employeeDB;

CREATE TABLE employees (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   email VARCHAR(100) NOT NULL UNIQUE,
   department VARCHAR(50),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Step 2: Install Dependencies

```
mkdir employee-management
cd employee-management
npm init -y
npm install mysql2
```

Step 3: Database Connection

File: config/db.js

```
const mysql = require("mysql2/promise");

async function connectDB() {
  const connection = await mysql.createConnection({
    host: "localhost",
    user: "root",          //  your MySQL username
    password: "yourpassword", //  your MySQL password
    database: "employeeDB"
  });
  console.log(" Connected to MySQL Database");
  return connection;
}

module.exports = connectDB;
```

Step 4: Employee Model (CRUD Queries)

File: models/employeeModel.js

```
// Contains raw SQL queries


module.exports = {
  insert: "INSERT INTO employees (name, email, department) VALUES (?, ?, ?)",
  selectAll: "SELECT * FROM employees",
  update: "UPDATE employees SET department = ? WHERE id = ?",
  delete: "DELETE FROM employees WHERE id = ?"
};
```

Step 5: Employee Service (Business Logic)

File: services/employeeService.js

```
const queries = require("../models/employeeModel");


async function addEmployee(connection, name, email, department) {
  const [result] = await connection.execute(queries.insert, [name, email, department]);
  console.log("Employee Added with ID:", result.insertId);
}


async function listEmployees(connection) {
  const [rows] = await connection.execute(queries.selectAll);
  console.log("Employees List:");
  console.table(rows);
}


async function updateEmployee(connection, id, newDepartment) {
  const [result] = await connection.execute(queries.update, [newDepartment, id]);
  console.log(` Employee ID ${id} updated, affected rows: ${result.affectedRows}`);
}

async function deleteEmployee(connection, id) {
  const [result] = await connection.execute(queries.delete, [id]);
  console.log(`Employee ID ${id} deleted, affected rows: ${result.affectedRows}`);
}

module.exports = { addEmployee, listEmployees, updateEmployee, deleteEmployee };
```

Step 6: Main Runner (Workflow)

File: app.js

```
const connectDB = require("./config/db");
const { addEmployee, listEmployees, updateEmployee, deleteEmployee } = require("./services/employeeService");

async function run() {
  const connection = await connectDB();

  // Add Employees
  await addEmployee(connection, "Alice", "alice@example.com", "HR");
  await addEmployee(connection, "Bob", "bob@example.com", "Engineering");
  await addEmployee(connection, "Charlie", "charlie@example.com", "Finance");

  // List Employees
  await listEmployees(connection);
  // Update Employee
  await updateEmployee(connection, 2, "IT");  // Update Bob’s department

  // List Again
  await listEmployees(connection);

  // Delete Employee
  await deleteEmployee(connection, 1); // Delete Alice

  // Final List
  await listEmployees(connection);

  await connection.end();
  console.log(" Connection Closed");
}

run();
```

Run the Project

```
node app.js
```
