Discover. Learn. Empower.
DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Student Name: Sujal Chandel **UID: 22BCS15017**
Branch: CSE **Section: 22BCS_IOT-619/A**

EXPERIMENT – 5 (Project Based Learning with Java)

CODE JAVA(Employee Managment System)

```
package com.assignment4;


import java.sql.*;
import java.util.Scanner;


public class EmployeeManagement {
    private static Connection conn = DatabaseConnection.getConnection();
    private static Scanner scanner = new Scanner(System.in);


    public static void addEmployee() {
        try {
            System.out.print("Enter Employee Name: ");
            String name = scanner.next();
            System.out.print("Enter Salary: ");
            double salary = scanner.nextDouble();


            String query = "INSERT INTO employees (name, salary) VALUES (?,
?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, name);
            stmt.setDouble(2, salary);
            stmt.executeUpdate();
            System.out.println("☑ Employee Added Successfully!");
        } catch (SQLException e) {
            System.out.println("❌ Error adding employee: " + e.getMessage());
        }
    }
```

```
    public static void displayEmployees() {
        try {
            String query = "SELECT id, name, salary FROM employees";
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            System.out.println("\nID | Name | Salary");
            System.out.println("---------------------");
            while (rs.next()) {
                System.out.println(rs.getInt("id") + " | " +
rs.getString("name") + " | " + rs.getDouble("salary"));
            }
        } catch (SQLException e) {
            System.out.println("❌ Error displaying employees: " +
e.getMessage());
        }
    }
}
```

CODE JAVA(Card collection):

```
package com.assignment4;

import java.sql.*;
import java.util.Scanner;

public class CardCollection {
    private static Connection conn = DatabaseConnection.getConnection();
    private static Scanner scanner = new Scanner(System.in);

    public static void addCard() {
        try {
            System.out.print("Enter Card Symbol: ");
            String symbol = scanner.next();
            System.out.print("Enter Card Rank: ");
            int rank = scanner.nextInt();

            String query = "INSERT INTO cards (symbol, rank) VALUES (?, ?)";
            PreparedStatement stmt = conn.prepareStatement(query);
            stmt.setString(1, symbol);
            stmt.setInt(2, rank);
            stmt.executeUpdate();
            System.out.println("☑ Card Added Successfully!");
        } catch (SQLException e) {
            System.out.println("❌ Error adding card: " + e.getMessage());
        }
    }

    public static void displayCards() {
        try {
```

```
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM cards");
            while (rs.next()) {
                System.out.println("Symbol: " + rs.getString("symbol") + ",
Rank: " + rs.getInt("rank"));
            }
        } catch (SQLException e) {
            System.out.println("❌ Error displaying cards: " + e.getMessage());
        }
    }
}
```

CODE JAVA(Ticket Booking System):

```
package com.assignment4;

import java.sql.*; import java.util.Scanner;

public class TicketBookingSystem { private static Scanner scanner = new
Scanner(System.in); private static Connection conn =
DatabaseConnection.getConnection();

public static void bookTicket() {
    try {
        System.out.print("Enter Passenger Name: ");
        String name = scanner.next();


        // Find next available seat
        String seatQuery = "SELECT seat_number FROM tickets WHERE status =
'Available' LIMIT 1";
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(seatQuery);


        if (rs.next()) {
            int seatNumber = rs.getInt("seat_number");


            // Book the ticket
            String updateQuery = "UPDATE tickets SET status = 'Booked',
passenger_name = ? WHERE seat_number = ?";
            PreparedStatement updateStmt = conn.prepareStatement(updateQuery);
            updateStmt.setString(1, name);
            updateStmt.setInt(2, seatNumber);
            updateStmt.executeUpdate();


            System.out.println("✅ Ticket Booked Successfully! Seat Number: " +
seatNumber);
        } else {
            System.out.println("⚠ No available seats.");
```

```
        }
    } catch (SQLException e) {
        System.out.println("❌ Error booking ticket: " + e.getMessage());
    }
}

public static void displayTickets() {
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM tickets");

        System.out.println("\nPassenger Name | Seat Number | Status");
        System.out.println("---------------------------------");
        while (rs.next()) {
            System.out.println(rs.getString("passenger_name") + " | " +
rs.getInt("seat_number") + " | " + rs.getString("status"));
        }
    } catch (SQLException e) {
        System.out.println("❌ Error displaying tickets: " + e.getMessage());
    }
}}
```

CODE JAVA(Main)

```
package com.assignment4;


import java.sql.SQLException;
import java.util.Scanner;


public class Main {
    private static Scanner scanner = new Scanner(System.in);


    public static void main(String[] args) throws SQLException {
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Manage Cards");
            System.out.println("2. Manage Employees");
            System.out.println("3. Book Tickets");
            System.out.println("4. Exit");
            System.out.print("Select an option: ");
            int choice = scanner.nextInt();


            switch (choice) {
                case 1:
                    System.out.println("1. Add Card");
                    System.out.println("2. Display Cards");
                    int cardChoice = scanner.nextInt();
                    if (cardChoice == 1) CardCollection.addCard();
                    else if (cardChoice == 2) CardCollection.displayCards();
                    break;
                case 2:
                    System.out.println("1. Add Employee");
                    System.out.println("2. Display Employees");
                    int empChoice = scanner.nextInt();
                    if (empChoice == 1) EmployeeManagement.addEmployee();
                    else if (empChoice == 2)
EmployeeManagement.displayEmployees();
                    break;
                case 3:
```

```
                    System.out.println("1. Book Ticket");
                    System.out.println("2. Display Tickets");
                    int ticketChoice = scanner.nextInt();
                    if (ticketChoice == 1) TicketBookingSystem.bookTicket();
                    else if (ticketChoice == 2)
TicketBookingSystem.displayTickets();
                    break;
                case 4:
                    System.out.println("Exiting... Goodbye!");
                    return;
                default:
                    System.out.println("❌ Invalid Choice! Try again.");
            }
        }
    }
}
```

CODE JAVA(DatabaseConnection):

*To connect SQL database to make all above 3 codes working

```
package com.assignment4;
import java.sql.*;

public class DatabaseConnection {
    private static final String URL =
"jdbc:mysql://localhost:3306/pblj_assignment4";
    private static final String USER = "root"; // Change if needed
    private static final String PASSWORD = "Sujal@2004"; // Change to your
MySQL password

    private static Connection conn;

    static {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("☑ Connected to MySQL Database!");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static Connection getConnection() {
        return conn;
    }
}
```

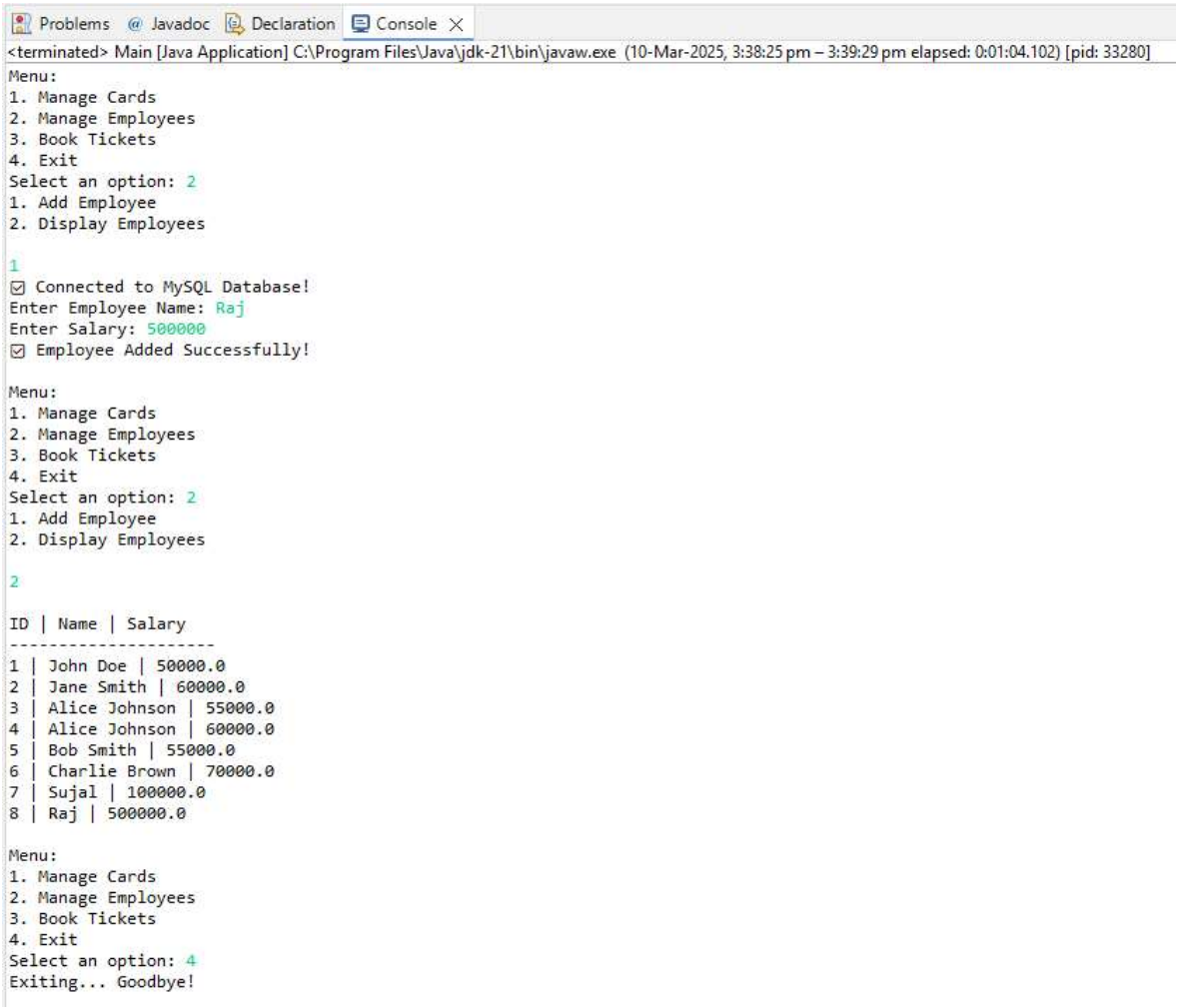
SQL Shell:

*used to create database and tables in backend

E.G. image.

```
MySQL  localhost:33060+ ssl  pblj_assignment4  SQL > SHOW TABLES;
+----------------------------+
| Tables_in_pblj_assignment4 |
+----------------------------+
| cards                      |
| employees                  |
| tickets                    |
+----------------------------+
3 rows in set (0.0016 sec)
MySQL  localhost:33060+ ssl  pblj_assignment4  SQL > DESCRIBE employees ;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| id     | int          | NO   | PRI | NULL    | auto_increment |
| name   | varchar(100) | NO   |     | NULL    |                |
| salary | double       | NO   |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
3 rows in set (0.0016 sec)
MySQL  localhost:33060+ ssl  pblj_assignment4  SQL > SELECT * FROM employees;
+----+---------------+--------+
| id | name          | salary |
+----+---------------+--------+
|  1 | John Doe      |  50000 |
|  2 | Jane Smith    |  60000 |
|  3 | Alice Johnson |  55000 |
|  4 | Alice Johnson |  60000 |
|  5 | Bob Smith     |  55000 |
|  6 | Charlie Brown |  70000 |
+----+---------------+--------+
6 rows in set (0.0007 sec)
MySQL  localhost:33060+ ssl  pblj_assignment4  SQL > SHOW TABLE STATUS;
+-----------+--------+---------+------------+------+----------------+-------------+-----------------+--------------+-----------+----------------+---------------------+---------------------+------------+-------
------------+----------+--------------------+---------+
| Name      | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length | Data_free | Auto_increment | Create_time         | Update_time         | Check_time | Collat
ion          | Checksum | Create_options | Comment |
+-----------+--------+---------+------------+------+----------------+-------------+-----------------+--------------+-----------+----------------+---------------------+---------------------+------------+-------
------------+----------+--------------------+---------+
| cards     | InnoDB |      10 | Dynamic    |    9 |           1820 |       16384 |               0 |            0 |         0 |             10 | 2025-02-27 20:24:24 | 2025-02-27 21:55:17 | NULL       | utf8mb
4_0900_ai_ci |     NULL |                    |         |
| employees | InnoDB |      10 | Dynamic    |    6 |           2730 |       16384 |               0 |            0 |         0 |              7 | 2025-02-27 21:46:17 | 2025-02-27 21:49:59 | NULL       | utf8mb
4_0900_ai_ci |     NULL |                    |         |
| tickets   | InnoDB |      10 | Dynamic    |    5 |           3276 |       16384 |               0 |            0 |         0 |           NULL | 2025-02-27 20:24:49 | 2025-02-27 21:39:48 | NULL       | utf8mb
4_0900_ai_ci |     NULL |                    |         |
+-----------+--------+---------+------------+------+----------------+-------------+-----------------+--------------+-----------+----------------+---------------------+---------------------+------------+-------
------------+----------+--------------------+---------+
3 rows in set (0.0021 sec)
MySQL  localhost:33060+ ssl  pblj_assignment4  SQL >
```

Output:



```
Problems  Javadoc  Declaration  Console
<terminated> Main [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (10-Mar-2025, 3:38:25 pm – 3:39:29 pm elapsed: 0:01:04.102) [pid: 33280]
Menu:
1. Manage Cards
2. Manage Employees
3. Book Tickets
4. Exit
Select an option: 2
1. Add Employee
2. Display Employees

1
☑ Connected to MySQL Database!
Enter Employee Name: Raj
Enter Salary: 500000
☑ Employee Added Successfully!

Menu:
1. Manage Cards
2. Manage Employees
3. Book Tickets
4. Exit
Select an option: 2
1. Add Employee
2. Display Employees

2

ID | Name | Salary
---------------------
1 | John Doe | 50000.0
2 | Jane Smith | 60000.0
3 | Alice Johnson | 55000.0
4 | Alice Johnson | 60000.0
5 | Bob Smith | 55000.0
6 | Charlie Brown | 70000.0
7 | Sujal | 100000.0
8 | Raj | 500000.0

Menu:
1. Manage Cards
2. Manage Employees
3. Book Tickets
4. Exit
Select an option: 4
Exiting... Goodbye!
```