Assignment -5

CaseStudy1:OnlineCourseRegistration System

Objective: Allow students to register/unregister for courses and view course details

TableStructure:

CREATE DATABASE course_db;

USE course_db;

CREATE TABLE courses (course_id INT PRIMARY KEY, course_name VARCHAR(100), faculty VARCHAR(100), credits INT);

MY QUERIES:

use course_db;

create table courses(course_id INT PRIMARY KEY, course_name VARCHAR(100), faculty VARCHAR(100), credits INT);

select*from courses;

JDBC Connections:

Package Assignment5

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

public class CourseJDBC {
    static final String URL = "jdbc:mysql://localhost:3306/course_db";
    static USER = "john";
    static final String PASSWORD = "12345";

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            System.out.println("Connected to database.");

            while (true) {
```

```
            System.out.println("\n1. Add Course\n2. View Courses\n3. Update Course\n4. Delete Course\n5. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();
            sc.nextLine(); // consume newline

            switch (choice) {
                case 1:
                    addCourse(conn, sc);
                    break;
                case 2:
                    viewCourses(conn);
                    break;
                case 3:
                    updateCourse(conn, sc);
                    break;
                case 4:
                    deleteCourse(conn, sc);
                    break;
                case 5:
                    System.out.println("Exiting...");
                    return;
                default:
                    System.out.println("Invalid choice!");
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// Inserting the values
static void addCourse(Connection conn, Scanner sc) throws SQLException {
```

```
    System.out.print("Enter Course ID: ");
    int id = sc.nextInt();
    sc.nextLine();
    System.out.print("Enter Course Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Faculty: ");
    String faculty = sc.nextLine();
    System.out.print("Feedback: ");
    int credits = sc.nextInt();

    String sql = "INSERT INTO courses VALUES (?, ?, ?, ?)";
    try (PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setInt(1, id);
        stmt.setString(2, name);
        stmt.setString(3, faculty);
        stmt.setInt(4, Feedback);
        int rows = stmt.executeUpdate();
        System.out.println(rows + " course added.");
    }
}

// Select the value
static void viewCourses(Connection conn) throws SQLException {
    String sql = "SELECT * FROM courses";
    try (Statement stmt = conn.createStatement();
         ResultSet rs = stmt.executeQuery(sql)) {
        System.out.println("\nCourse List:");
        while (rs.next()) {
            System.out.println("ID: " + rs.getInt("course_id") +
                    ", Name: " + rs.getString("course_name") +
                    ", Faculty: " + rs.getString("faculty") +
                    ", Credits: " + rs.getInt("Feedback"));
```

```
        }
    }
}

// Update the values
static void updateCourse(Connection conn, Scanner sc) throws SQLException {
    System.out.print("Enter Course ID to update: ");
    int id = sc.nextInt();
    sc.nextLine();
    System.out.print("Enter new Faculty: ");
    String faculty = sc.nextLine();
    System.out.print("Enter new Credits: ");
    int credits = sc.nextInt();

    String sql = "UPDATE courses SET faculty = ?, Feedback = ? WHERE course_id = ?";
    try (PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setString(1, faculty);
        stmt.setInt(2, feedback);
        stmt.setInt(3, id);
        int rows = stmt.executeUpdate();
        System.out.println(rows + " course updated.");
    }
}

// DELETE
static void deleteCourse(Connection conn, Scanner sc) throws SQLException {
    System.out.print("Enter Course ID to delete: ");
    int id = sc.nextInt();

    String sql = "DELETE FROM courses WHERE course_id = ?";
    try (PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setInt(1, id);
```

```
            int rows = stmt.executeUpdate();
            System.out.println(rows + " course deleted.");
        }
    }
}
```

Case Study 2: Product Inventory System

Objective:

Track product stock in a retail store.

Table Structure:C

CREATE DATABASE inventory_db;

USE inventory_db;

CREATE TABLE products (

product_id INT PRIMARY KEY, product_name VARCHAR(100), quantity INT, price DECIMAL(10,2));

Package Assignment5

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

public class ProductInventory {

    String URL = "jdbc:mysql://localhost:3306/inventory_db";
    String USER = "chinni";
    String PASSWORD = "12345”

   public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);

        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            System.out.println("Connected to Inventory Database");

            while (true) {
                System.out.println("\n1. Add Product\n2. View Products\n3. Update Product Quantity\n4. Delete Product\n5. Exit");
                System.out.print("Enter your choice: ");
                int choice = sc.nextInt();
                sc.nextLine();

                switch (choice) {
                    case 1:
                        addProduct(conn, sc);
                        break;
                    case 2:
                        viewProducts(conn);
                        break;
                    case 3:
                        updateQuantity(conn, sc);
                        break;
                    case 4:
                        deleteProduct(conn, sc);
                        break;
                    case 5:
                        System.out.println("Exiting Program.");
                        return;
                    default:
                        System.out.println("Invalid choice.");
                }
            }
        } catch (SQLException e) {
```

```
            e.printStackTrace();
        }
    }

    // Insert the values
    private static void addProduct(Connection conn, Scanner sc) throws SQLException {
        System.out.print("Enter Product ID: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Product Name: ");
        String name = sc.nextLine();
        System.out.print("Enter Quantity: ");
        int quantity = sc.nextInt();
        System.out.print("Enter Price: ");
        double price = sc.nextDouble();

        String sql = "INSERT INTO products VALUES (?, ?, ?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(1, id);
            stmt.setString(2, name);
            stmt.setInt(3, quantity);
            stmt.setDouble(4, price);
            int rows = stmt.executeUpdate();
            System.out.println(rows + " product added.");
        }
    }

    // Select the value
    private static void viewProducts(Connection conn) throws SQLException {
        String sql = "SELECT * FROM products";
        try (Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(sql)) {
```

```
        System.out.println("\nCurrent Inventory:");
        while (rs.next()) {
            System.out.println("ID: " + rs.getInt("product_id")
                    + ", Name: " + rs.getString("product_name")
                    + ", Quantity: " + rs.getInt("quantity")
                    + ", Price: ₹" + rs.getDouble("price"));
        }
    }
}

// update the values
private static void updateQuantity(Connection conn, Scanner sc) throws SQLException {
    System.out.print("Enter Product ID to update: ");
    int id = sc.nextInt();
    System.out.print("Enter New Quantity: ");
    int quantity = sc.nextInt();

    String sql = "UPDATE products SET quantity = ? WHERE product_id = ?";
    try (PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setInt(1, quantity);
        stmt.setInt(2, id);
        int rows = stmt.executeUpdate();
        System.out.println(rows + " product updated.");
    }
}

// Delete the values
private static void deleteProduct(Connection conn, Scanner sc) throws SQLException {
    System.out.print("Enter Product ID to delete: ");
    int id = sc.nextInt();
```

```
        String sql = "DELETE FROM products WHERE product_id = ?";
        try (PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(1, id);
            int rows = stmt.executeUpdate();
            System.out.println(rows + " product deleted.");
        }
    }
}
```