Experiment -7

Student Name: Utkarsh Kumar **UID:22BCS10233**
Branch: BE-CSE **Section/Group:IOT_636-a**
Semester:6th **Date of Performance:4/04/2025**
Subject Name: Project Based Learning in Java with Lab **Subject Code: 22CSH-359**

7.1.1 Aim: 1.Build a program to perform CRUD operations (Create, Read, Update, Delete) on a database table Product with columns: ProductID, ProductName, Price, and Quantity. The program should include: Menu-driven options for each operation. Transaction handling to ensure data integrity.

7.1.2 Objective: To develop a Java application that performs CRUD (Create, Read, Update, Delete) operations on a `Product` database table using JDBC**.**

7.1.3 Code:

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

public class ProductCRUD
{
   static final String
DB_URL =
"jdbc:mysql://localhost:330
6/your_database";
   static final String USER
= "your_username";
   static final String PASS
= "your_password";

   public static void
main(String[] args) {
       Scanner sc = new
Scanner(System.in);
       try (Connection conn =
DriverManager.getConnecti
on(DB_URL, USER,
PASS)) {

conn.setAutoCommit(false)
```

```
;  // Enable transaction
handling
        int choice;

        do {

System.out.println("\n---
Product CRUD Menu ---");

System.out.println("1.
Create Product");

System.out.println("2. Read
Products");

System.out.println("3.
Update Product");

System.out.println("4.
Delete Product");

System.out.println("5.
Exit");

System.out.print("Enter
your choice: ");
            choice =
sc.nextInt();

            switch (choice) {
               case 1:

createProduct(conn, sc);
                  break;
               case 2:

readProducts(conn);
```

```
                    break;
                case 3:

updateProduct(conn, sc);
                    break;
                case 4:

deleteProduct(conn, sc);
                    break;
                case 5:

System.out.println("Exiting
...");
                    break;
                default:

System.out.println("Invalid
choice. Try again.");
                }
            } while (choice !=
5);

        } catch
(SQLException e) {

System.out.println("Databa
se connection error: " +
e.getMessage());
        }
    }

    private static void
createProduct(Connection
conn, Scanner sc) {
        try {

System.out.print("Enter
ProductID: ");
            int id = sc.nextInt();
            sc.nextLine();
```

```
System.out.print("Enter
ProductName: ");
        String name =
sc.nextLine();

System.out.print("Enter
Price: ");
        double price =
sc.nextDouble();

System.out.print("Enter
Quantity: ");
        int quantity =
sc.nextInt();

        String query =
"INSERT INTO Product
(ProductID, ProductName,
Price, Quantity) VALUES
(?, ?, ?, ?)";
        PreparedStatement
pstmt =
conn.prepareStatement(que
ry);
        pstmt.setInt(1, id);
        pstmt.setString(2,
name);
        pstmt.setDouble(3,
price);
        pstmt.setInt(4,
quantity);

pstmt.executeUpdate();

        conn.commit(); //
Commit transaction
```

```
System.out.println("Product
inserted successfully.");
        } catch
(SQLException e) {
            try {
                conn.rollback(); //
Rollback on error

System.out.println("Error
inserting product.
Transaction rolled back.");
            } catch
(SQLException rollbackEx)
{

System.out.println("Rollbac
k failed: " +
rollbackEx.getMessage());
            }
        }
    }

    private static void
readProducts(Connection
conn) {
        try {
            String query =
"SELECT * FROM
Product";
            Statement stmt =
conn.createStatement();
            ResultSet rs =
stmt.executeQuery(query);


System.out.println("\nProd
uctID | ProductName | Price
| Quantity");
            while (rs.next()) {
```

```
System.out.printf("%d | %s
| %.2f | %d\n",

rs.getInt("ProductID"),

rs.getString("ProductName"
),

rs.getDouble("Price"),

rs.getInt("Quantity"));
            }
        } catch
(SQLException e) {

System.out.println("Error
reading products: " +
e.getMessage());
        }
    }

    private static void
updateProduct(Connection
conn, Scanner sc) {
        try {

System.out.print("Enter
ProductID to update: ");
            int id = sc.nextInt();
            sc.nextLine();

System.out.print("Enter
new ProductName: ");
            String name =
sc.nextLine();
```

```
System.out.print("Enter
new Price: ");
        double price =
sc.nextDouble();

System.out.print("Enter
new Quantity: ");
        int quantity =
sc.nextInt();

        String query =
"UPDATE Product SET
ProductName = ?, Price =
?, Quantity = ? WHERE
ProductID = ?";
        PreparedStatement
pstmt =
conn.prepareStatement(que
ry);
        pstmt.setString(1,
name);
        pstmt.setDouble(2,
price);
        pstmt.setInt(3,
quantity);
        pstmt.setInt(4, id);
        int rows =
pstmt.executeUpdate();

        if (rows > 0) {
            conn.commit();

System.out.println("Product
updated successfully.");
        } else {

System.out.println("Product
not found.");
        }
```

```
        } catch
(SQLException e) {
            try {
                conn.rollback();

System.out.println("Error
updating product.
Transaction rolled back.");
            } catch
(SQLException rollbackEx)
{

System.out.println("Rollbac
k failed: " +
rollbackEx.getMessage());
            }
        }
    }

    private static void
deleteProduct(Connection
conn, Scanner sc) {
        try {

System.out.print("Enter
ProductID to delete: ");
            int id = sc.nextInt();

            String query =
"DELETE FROM Product
WHERE ProductID = ?";
            PreparedStatement
pstmt =
conn.prepareStatement(que
ry);
            pstmt.setInt(1, id);
```

```
            int rows =
pstmt.executeUpdate();

            if (rows > 0) {
                conn.commit();

System.out.println("Product
deleted successfully.");
            } else {

System.out.println("Product
not found.");
            }
        } catch
(SQLException e) {
            try {
                conn.rollback();

System.out.println("Error
deleting product.
Transaction rolled back.");
            } catch
(SQLException rollbackEx)
{

System.out.println("Rollbac
k failed: " +
rollbackEx.getMessage());
            }
        }
    }
}
```

7.1.4 Output:

```
--- Product CRUD Menu ---
1. Create Product
2. Read Products
3. Update Product
4. Delete Product
5. Exit
Enter your choice: 1
Enter ProductID: 101
Enter ProductName: Mouse
Enter Price: 799.50
Enter Quantity: 25
Product inserted successfully.

--- Product CRUD Menu ---
Enter your choice: 2
ProductID | ProductName | Price | Quantity
101       | Mouse       | 799.50 | 25

--- Product CRUD Menu ---
Enter your choice: 3
Enter ProductID to update: 101
Enter new ProductName: Wireless Mouse
Enter new Price: 999.99
Enter new Quantity: 30
Product updated successfully.

--- Product CRUD Menu ---
Enter your choice: 4
Enter ProductID to delete: 101
Product deleted successfully.

--- Product CRUD Menu ---
Enter your choice: 5
Exiting...
```

7.2.1Aim: 2. Develop a Java application using JDBC and MVC architecture to manage student data. The application should:

Use a Student class as the model with fields like StudentID, Name, Department, and Marks.

Include a database table to store student data.

Allow the user to perform CRUD operations through a simple menu-driven view.

Implement database operations in a separate controller class.

Implement both these programs.

7.2.2Objective: The objective of this Java application is to:Develop a student management system using JDBC and the MVC (Model-View-Controller) architecture to perform basic CRUD operations on a student database**.**

7.2.3Code:

```
import java.util.Scanner;

public class MainApp {
    public static void main(String[] args) {
        StudentController controller = new StudentController();
        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n--- Student Management ---");
            System.out.println("1. Add Student");
            System.out.println("2. View All Students");
            System.out.println("3. Update Student Marks");
            System.out.println("4. Delete Student");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter ID: ");
                    int id = sc.nextInt();
```

```
            sc.nextLine(); // clear buffer
            System.out.print("Enter Name: ");
            String name = sc.nextLine();
            System.out.print("Enter Department: ");
            String dept = sc.nextLine();
            System.out.print("Enter Marks: ");
            float marks = sc.nextFloat();

            Student s = new Student(id, name, dept, marks);
            controller.addStudent(s);
            break;

        case 2:
            controller.viewStudents();
            break;

        case 3:
            System.out.print("Enter ID to update marks: ");
            int uid = sc.nextInt();
            System.out.print("Enter new Marks: ");
            float newMarks = sc.nextFloat();
            controller.updateStudentMarks(uid, newMarks);
            break;

        case 4:
            System.out.print("Enter ID to delete: ");
            int did = sc.nextInt();
            controller.deleteStudent(did);
            break;

        case 5:
            System.out.println("Exiting...");
            break;

        default:
            System.out.println("Invalid choice!");
```

```
            }
        } while (choice != 5);

        sc.close();
    }
}
```

7.2.4Output:

```
--- Student Management ---
1. Add Student
2. View All Students
3. Update Student Marks
4. Delete Student
5. Exit
Enter your choice: 1
Enter ID: 101
Enter Name: Rahul
Enter Department: CSE
Enter Marks: 89.5
Student added successfully!

--- Student Management ---
Enter your choice: 2
ID: 101, Name: Rahul, Dept: CSE, Marks: 89.5

--- Student Management ---
Enter your choice: 3
Enter ID to update marks: 101
Enter new Marks: 92.0
Marks updated successfully!

--- Student Management ---
Enter your choice: 4
Enter ID to delete: 101
Student deleted successfully!
```