MINI PROJECT
LIBRARY MANAGEMENT SYSTEM

AIM:

To develop a Java-based Medical Shop Inventory System that efficiently manages medicine stock, tracks expiry dates, and updates quantities, ensuring smooth inventory operations for medical stores.

ALGORITHM:

1) Start.
2) Establish a connection to the MySQL database using JDBC.
3) Show the main menu with options to add a book, display books, or exit.
4) Prompt the user for book details (title, author, genre, and price).
5) Insert the entered details into the Books table in the database.
6) Retrieve all records from the Books table.
7) Format and display the book details (ID, title, author, genre, price).
8) Loop back to the main menu until the user chooses to exit.
9) Based on user input, call the corresponding function (add or display books)
10) Close the database connection and terminate the program.
11) Stop.

Program :

SQL Code:

```
CREATE DATABASE MedicalShop;
USE MedicalShop;
CREATE TABLE Medicines (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    quantity INT NOT NULL,
    price DOUBLE NOT NULL,
    expiry_date DATE NOT NULL
);
```

Java Code:

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

public class MedicalShopInventorySQL {
    private static final String DB_URL = "jdbc:mysql://localhost:3306/MedicalShop";
    private static final String DB_USER = "root"; // Replace with your DB username
    private static final String DB_PASSWORD = ""; // Replace with your DB password

    private static Connection connect() throws SQLException {
        return DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
    }
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int choice;

    do {
        System.out.println("\nMedical Shop Inventory System");
        System.out.println("1. Add Medicine");
        System.out.println("2. View Inventory");
        System.out.println("3. Search Medicine");
        System.out.println("4. Update Stock");
        System.out.println("5. Delete Medicine");
        System.out.println("6. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        switch (choice) {
            case 1 -> addMedicine(scanner);
            case 2 -> viewInventory();
            case 3 -> searchMedicine(scanner);
            case 4 -> updateStock(scanner);
            case 5 -> deleteMedicine(scanner);
            case 6 -> System.out.println("Exiting the system. Thank you!");
            default -> System.out.println("Invalid choice! Please try again.");
        }
    } while (choice != 6);
}

private static void addMedicine(Scanner scanner) {
    try (Connection conn = connect()) {
        System.out.print("Enter medicine name: ");
        String name = scanner.nextLine();
        System.out.print("Enter quantity: ");
        int quantity = scanner.nextInt();
        System.out.print("Enter price: ");
        double price = scanner.nextDouble();
        scanner.nextLine(); // Consume newline
        System.out.print("Enter expiry date (YYYY-MM-DD): ");
        String expiryDate = scanner.nextLine();

        String sql = "INSERT INTO Medicines (name, quantity, price, expiry_date) VALUES (?, ?, ?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString(1, name);
            stmt.setInt(2, quantity);
            stmt.setDouble(3, price);
            stmt.setDate(4, Date.valueOf(expiryDate));
            stmt.executeUpdate();
            System.out.println("Medicine added successfully!");
        }
    } catch (SQLException e) {
        System.out.println("Error: " + e.getMessage());
    }
}

private static void viewInventory() {
    try (Connection conn = connect()) {
        String sql = "SELECT * FROM Medicines";
        try (Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(sql)) {
            System.out.println("\nCurrent Inventory:");
            while (rs.next()) {
                System.out.printf("ID: %d, Name: %s, Quantity: %d, Price: ₹%.2f, Expiry: %s%n",
                        rs.getInt("id"), rs.getString("name"), rs.getInt("quantity"),
                        rs.getDouble("price"), rs.getDate("expiry_date"));
            }
        }
    } catch (SQLException e) {
        System.out.println("Error: " + e.getMessage());
    }
}

private static void searchMedicine(Scanner scanner) {
```

```
        try (Connection conn = connect()) {
            System.out.print("Enter medicine name to search: ");
            String name = scanner.nextLine();

            String sql = "SELECT * FROM Medicines WHERE name LIKE ?";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setString(1, "%" + name + "%");
                try (ResultSet rs = stmt.executeQuery()) {
                    if (!rs.next()) {
                        System.out.println("No medicine found.");
                    } else {
                        do {
                            System.out.printf("ID: %d, Name: %s, Quantity: %d, Price: ₹%.2f, Expiry: %s%n",
                                    rs.getInt("id"), rs.getString("name"), rs.getInt("quantity"),
                                    rs.getDouble("price"), rs.getDate("expiry_date"));
                        } while (rs.next());
                    }
                }
            }
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    private static void updateStock(Scanner scanner) {
        try (Connection conn = connect()) {
            System.out.print("Enter medicine ID to update: ");
            int id = scanner.nextInt();
            System.out.print("Enter new quantity: ");
            int quantity = scanner.nextInt();

            String sql = "UPDATE Medicines SET quantity = ? WHERE id = ?";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setInt(1, quantity);
                stmt.setInt(2, id);
                int rowsUpdated = stmt.executeUpdate();
                if (rowsUpdated > 0) {
                    System.out.println("Stock updated successfully!");
                } else {
                    System.out.println("No medicine found with the given ID.");
                }
            }
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    private static void deleteMedicine(Scanner scanner) {
        try (Connection conn = connect()) {
            System.out.print("Enter medicine ID to delete: ");
            int id = scanner.nextInt();

            String sql = "DELETE FROM Medicines WHERE id = ?";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setInt(1, id);
                int rowsDeleted = stmt.executeUpdate();
                if (rowsDeleted > 0) {
                    System.out.println("Medicine deleted successfully!");
                } else {
                    System.out.println("No medicine found with the given ID.");
                }
            }
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output :

```
Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 1

Enter medicine name: Paracetamol
Enter quantity: 50
Enter price: 1.25
Enter expiry date (YYYY-MM-DD): 2025-12-31
Medicine added successfully!

---------------------------------------

Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 2

Current Inventory:
ID: 1, Name: Paracetamol, Quantity: 50, Price: ₹1.25, Expiry: 2025-12-31

---------------------------------------

Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 3

Enter medicine name to search: Paracetamol
ID: 1, Name: Paracetamol, Quantity: 50, Price: ₹1.25, Expiry: 2025-12-31

---------------------------------------

Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 4

Enter medicine ID to update: 1
Enter new quantity: 100
Stock updated successfully!

---------------------------------------

Medical Shop Inventory System
1. Add Medicine
```

2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 2

Current Inventory:
ID: 1, Name: Paracetamol, Quantity: 100, Price: ₹1.25, Expiry: 2025-12-31

Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 5

Enter medicine ID to delete: 1
Medicine deleted successfully!

Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 2

Current Inventory:
Inventory is empty.

Medical Shop Inventory System
1. Add Medicine
2. View Inventory
3. Search Medicine
4. Update Stock
5. Delete Medicine
6. Exit
Enter your choice: 6

Exiting the system. Thank you!

Result:

The Medical Shop Inventory System efficiently manages medicine records, enabling users to add, view, search, update, and delete entries. It interacts with a MySQL database to store and retrieve details like name, quantity, price, and expiry date, confirming successful operations after each action.