Bushra Fatima

Computer and Information Sciences

Internet Application Development

Lab Number: 13

Problem: 1 & 3

Security Features of the Inventory Management System

1. Introduction

The Inventory Management System web application basically maintains essential data related to products, customers, orders, and users. Since it is a web-based application, I have added some security features in it to protect sensitive information and ensure secure user interactions.

2. Security Features And Test Cases

The following are some of the features that have been implemented in the semester project are there are some that are still in progress and soon within some time they will be completed

Role-Based Access Control (RBAC)

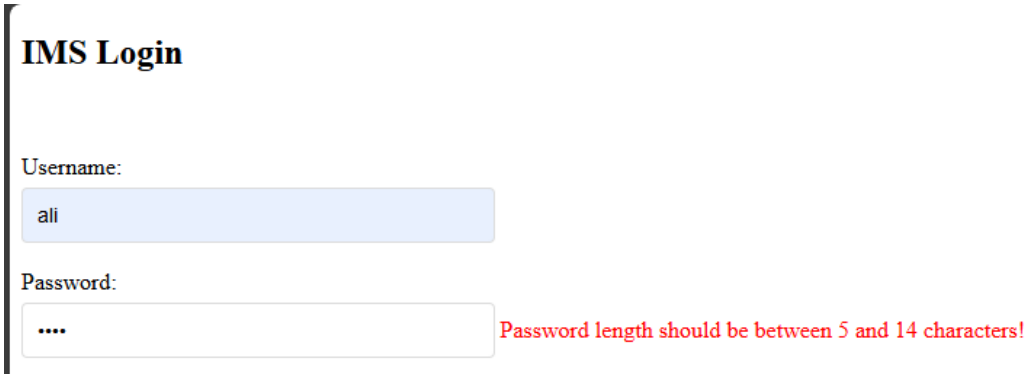
It ensures that users access pages according to their role. This helps in maintaining security that a viewer can never go to an admin page and take control of those functionalities that are accessible only to the Admin.

Test Case:

Password Check (Custom Validation)

It validates passwords against a set of security criteria, including length and complexity requirements. Here I have add only the a condition that controls the length of the password.

Test Case:



Email Validation (Regular Expression)

It uses a regular expression to validate email addresses, ensuring that users enter a correctly formatted email. This reduces errors during data insertion in the database tables and prevents invalid inputs from compromising the system.

Test Case:

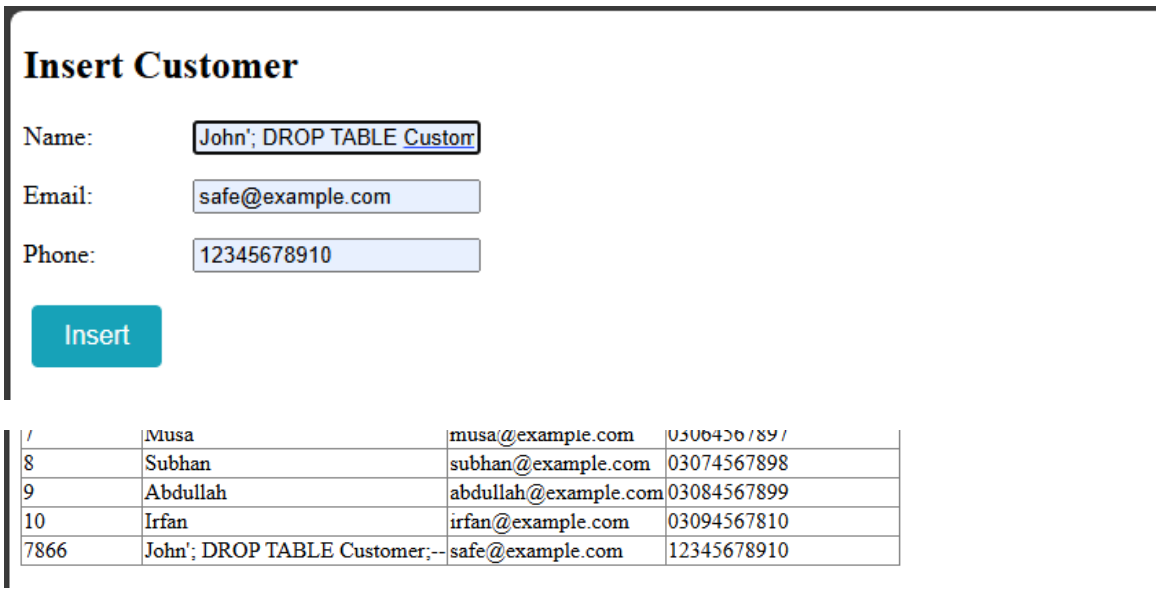
SQL Parameterized Queries

It uses parameterized SQL queries to prevent SQL injection attacks. Instead that the user input is directly embedded in the SQL statement, parameters are used, ensuring that the database remains secure from malicious inputs. This ensures that any special characters or SQL injection attempts in the input are treated as data, not as part of the SQL command.

Test Case:

```
(CustomerID, CustomerName, CustomerEmail, CustomerPhoneNumber) VALUES (@id, @name, @email, @phone)", con)
```

```
cmd.Parameters.AddWithValue("@id", newCustomerID)
cmd.Parameters.AddWithValue("@name", txtCustomerName.Text)
cmd.Parameters.AddWithValue("@email", txtCustomerEmail.Text)
cmd.Parameters.AddWithValue("@phone", txtCustomerPhone.Text)
```

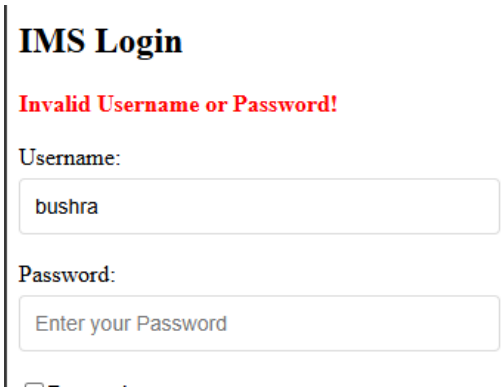


7	Musa	musa@example.com	03064567897
8	Subhan	subhan@example.com	03074567898
9	Abdullah	abdullah@example.com	03084567899
10	Irfan	irfan@example.com	03094567810
7866	John'; DROP TABLE Customer;--	safe@example.com	12345678910

Username and Password Check from Database

It verifies whether the entered username and password exist in the database. And also, the username and password do match or not. This step prevents unauthorized login attempts by users who may try to guess valid usernames.

Test Case:



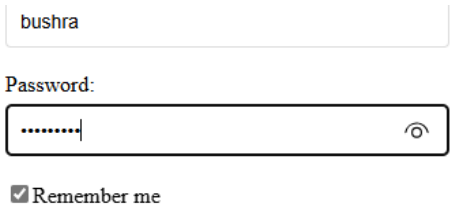
Cookies

Cookies are used that whenever a customer enters username and password and check the remember me box its user name is stored in the cookie and it remains stored until the user logs out from the page.

Test Case:

```
Protected Sub Page_Load(sender As Object, e As EventArgs)
    If Not IsPostBack Then
        If Request.Cookies("Username") IsNot Nothing Then
            txtUsername.Text = Request.Cookies("Username").Value
            chkRemember.Checked = True
        End If
    End If
End Sub
```

```
If chkRemember.Checked Then
    Dim rememberCookie As New HttpCookie("Username", txtUsername.Text)
    rememberCookie.Expires = DateTime.Now.AddDays(7)
    Response.Cookies.Add(rememberCookie)
Else
    If Request.Cookies("Username") IsNot Nothing Then
        Response.Cookies("Username").Expires = DateTime.Now.AddDays(-1)
    End If
End If
```



Session Management

Now what happen here is cookies store data on client-side where as with session management we store the data stored in cookie on the server-side and uses session ID to identify the user. But since time was short I was unable to do this hopefully it will be complete by the next submission.