2026

Lab 06 Report

SUBMITTED BY:

DANIYA AAMIR

Problem 1)

(a) As an initial step enhance the schema of PVFC database. Write enhanced schema below.

To implement Role-Based Access Control (RBAC), the existing Pine Valley Furniture Company (PVFC) database schema was enhanced by introducing a "Bridge Schema" that separates authentication credentials from the core customer demographic data. Two new tables were created and logically linked to the existing CUSTOMER_t table:

1. **Roles Table:** Acts as a static lookup table defining the distinct access levels within the system (e.g., RoleId 1 = Admin, 2 = Customer, 3 = Employee).
2. **Users Table:** Stores the secure login credentials (Username, Password) and acts as the bridge. It utilizes RoleId as a Foreign Key to enforce permissions, and Customer_Id as a Foreign Key to strictly link the login account to a specific customer profile in the CUSTOMER_t table.

Enhanced Schema (SQL Representation):

SQL

```
-- 1. Lookup table for RBAC tiers

CREATE TABLE Roles (

   RoleId INT PRIMARY KEY,

   RoleName VARCHAR(50) NOT NULL

);


-- 2. Bridge table for Authentication & Authorization

CREATE TABLE Users (

   UserId INT IDENTITY(1,1) PRIMARY KEY,

   Username VARCHAR(100) NOT NULL UNIQUE,

   Password VARCHAR(255) NOT NULL,

   RoleId INT NOT NULL,
```

```
    Customer_Id INT NULL,


    -- Enforcing Referential Integrity
    CONSTRAINT FK_Users_Roles FOREIGN KEY (RoleId) REFERENCES Roles(RoleId),
    CONSTRAINT FK_Users_Customer FOREIGN KEY (Customer_Id) REFERENCES
CUSTOMER_t(Customer_Id)
);
```

(b) Next enhance the solutions for above mentioned problems to take advantage of modified schema and incorporate role-based access control for all interfaces. Write the updated code for any one of the above problems below?

The interfaces were enhanced by implementing "Gatekeeper" validation checks within the backend Page_Load events, utilizing server-side Session variables assigned during login.

Below is the updated code for the **Product Catalog Update (Lab 5, Problem 3)** interface. This specific enhancement restricts catalog access solely to Administrators (Role 1) and Employees (Role 3), while forcefully redirecting unauthenticated Guests and regular Customers (Role 2) to the login screen.

Updated Code (catalog.aspx.vb):

VB.Net

```
Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load


    ' GATEKEEPER: Check if the session is empty (Unauthenticated Guest)
    ' OR if the assigned Role is a normal Customer (Role 2)
    If Session("RoleId") Is Nothing OrElse Convert.ToInt32(Session("RoleId")) = 2 Then


        ' Intercept the unauthorized access and redirect to the authentication page
        Response.Redirect("login.aspx", False)
```

```
        Exit Sub ' Immediately halt further execution of the page lifecycle


    End If


    ' Only fetch and bind the catalog data to the GridView if they passed the RBAC security check
    If Not IsPostBack Then
        BindCatalog()
    End If
End Sub
```

(c) Develop test cases for role based access control?

Below are the system test cases developed to validate that the RBAC mechanism effectively authenticates users and authorizes interface access based on the enhanced schema:

Test Case	Description	Input Data / Action	Expected Output
1	Admin Authentication & Authorization	Username: admin@pvfc.com Role ID: 1	System validates credentials against the Users table; assigns Session("RoleId") = 1; grants access to all interfaces including Catalog and Admin Dashboard.
2	Employee Authentication & Authorization	Username: employee@pvfc.com Role ID: 3	System validates credentials; assigns Session("RoleId") = 3; grants access to Catalog management but restricts Customer Profile updates.
3	Customer Authentication & Authorization	Username: customer@pvfc.com	System validates credentials; assigns Session("RoleId") = 2; dynamically hides Catalog navigation links; grants access

		Role ID: 2	strictly to Checkout and Profile Update.
4	Unauthorized Access Interception	User attempts to navigate directly to catalog.aspx via the browser URL bar without logging in.	Gatekeeper detects Session("RoleId") Is Nothing; immediately redirects the user to login.aspx and blocks data binding.
5	Session Destruction & Privilege Revocation	Authenticated user clicks the "Logout" button.	Session.Clear() and Session.Abandon() execute. The user's server footprint is destroyed, returning them to Guest status and revoking all RBAC privileges.

Some things beyond lab statements

Deployment Architecture & Server Constraints

1. The "Shared Brain" Deployment Challenge

During deployment to the live server (Somee.com), a critical issue arose regarding application boundaries. The project requires maintaining separate, accessible directories for each incremental lab (e.g., /lab5/ and /lab6/). However, ASP.NET Web Applications compile all backend code into a single .dll file located in the root bin folder. Because of this "Shared Brain" architecture, clicking a link in the older Lab 5 folder unintentionally triggered the new security gatekeepers written for Lab 6, resulting in broken links and 404 errors.

2. Overcoming Hosting Limitations (150MB Quota)

The standard enterprise solution to this problem is converting each folder into a separate IIS Application (Virtual Directory) with its own bin folder. However, the free Somee hosting tier imposes a strict 150MB storage quota. The Roslyn compiler data within a single bin folder averages 30MB-50MB. Uploading separate bin folders for every lab would rapidly crash the server storage and leave no room for database expansion.

3. The URL Bypass Resolution

To satisfy both the storage constraints and the grading requirements, a dynamic URL-checking mechanism was implemented within the page lifecycle. The Page_Load gatekeepers were wrapped in a conditional check (If Not Request.RawUrl.ToLower().Contains("/lab5/") Then).

This lightweight code modification allows the entire site to operate off a single bin folder (saving ~50MB per lab). The server reads the requested URL dynamically; if it detects the professor grading the legacy /lab5/ directory, it temporarily suspends RBAC execution. If the user is in /lab6/, strict security protocols are enforced.