

Internet Application Development

Lab 11B

Dated: Wednesday, 23rd April 2025

Registration No: 03-3-1-058-2022

Full Name: ZAINAB TARIQ

Online surveys are a great way to collect and store information. It gives you the convenience of compiling data to hold an event, find a result, or even prove a point. An online survey is a data collection method through which information is compiled with the help of different types of questions. These surveys can be created for many reasons, to choose a place to visit for your group, to get an idea about the opinions of your employees, or even to measure and improve your customer experience. Develop an application for preparation of a survey and its online conduction. The key features to be considered are as follows:

Divide the users into three types of roles:

- (i) Survey builder
- (ii) Surveyor
- (iii) Survey administrator

Implement identification for users of above roles.

Implement authorization so that appropriate interfaces are displayed to each user.

Survey administrator will set the identification or anonymity of surveyors.

Develop suitable interface for survey builder to enter survey questions including the following:

- (i) Question statement
- (ii) Answer options

Allow to select question types as multiple choice and true/ false.

Develop a suitable interface for surveyor to go through survey and answer questions.

Develop a survey results analysis interface.

Login.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" CodeFile="Login.aspx.vb" Inherits="Login" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Login Page</title>
```

```

<style>
    .login-box {
        width: 300px;
        margin: 100px auto;
        border: 1px solid #ccc;
        padding: 20px;
        font-family: Arial, sans-serif;
    }
    .hint {
        margin-top: 15px;
        font-size: 12px;
        color: #555;
        background: #f9f9f9;
        padding: 10px;
        border: 1px dashed #aaa;
    }
    label, input, button {
        display: block;
        width: 100%;
        margin-bottom: 10px;
    }
    input {
        padding: 6px;
        font-size: 14px;
    }
    button {
        padding: 8px;
        font-size: 16px;
        background-color: #0078d7;
        color: white;
        border: none;
        cursor: pointer;
    }
    .error-message {
        color: red;
        font-weight: bold;
        margin-bottom: 10px;
    }
</style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="login-box">
            <asp:Label ID="lblError" runat="server" CssClass="error-message"></asp:Label>

            <label for="txtUsername">Username:</label>
            <asp:TextBox ID="txtUsername" runat="server" />

            <label for="txtPassword">Password:</label>
            <asp:TextBox ID="txtPassword" runat="server" TextMode="Password" />

            <asp:Button ID="btnLogin" runat="server" Text="Login" OnClick="btnLogin_Click" />

            <div class="hint">
                <strong>Demo Credentials:</strong><br />
                Survey Builder: builder1 / pass<br />
                Surveyor: surveyor1 / pass<br />
                Survey Administrator: admin1 / pass
            </div>
        </div>
    </form>
</body>
</html>

```

Imports System

Partial Class Login

```
Inherits System.Web.UI.Page

' User roles enum
Public Enum UserRole
    SurveyBuilder
    Surveyor
    SurveyAdmin
End Enum

' User class
Public Class User
    Public Property Username As String
    Public Property Password As String
    Public Property Role As UserRole
End Class

' Hardcoded users
Private Shared ReadOnly Users As New List(Of User) From {
    New User() With {.Username = "builder1", .Password = "pass", .Role =
UserRole.SurveyBuilder},
    New User() With {.Username = "surveyor1", .Password = "pass", .Role =
UserRole.Surveyor},
    New User() With {.Username = "admin1", .Password = "pass", .Role =
UserRole.SurveyAdmin}
}

Protected Sub btnLogin_Click(sender As Object, e As EventArgs)
    Dim username As String = txtUsername.Text.Trim()
    Dim password As String = txtPassword.Text.Trim()

    Dim user = Users.Find(Function(u) u.Username.Equals(username,
StringComparison.OrdinalIgnoreCase) AndAlso u.Password = password)

    If user IsNot Nothing Then
        ' Store user info in session
        Session("Username") = user.Username
        Session("UserRole") = user.Role

        ' Redirect to Dashboard (you can create Dashboard.aspx to handle role-based
redirect)
        Response.Redirect("Dashboard.aspx")
    Else
        lblError.Text = "Invalid username or password."
    End If
End Sub

End Class
```

SurveyBuilder.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" CodeFile="SurveyBuilder.aspx.vb"
Inherits="SurveyBuilder" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Survey Builder</title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="width:400px; margin:auto; padding:20px; border:1px solid #ccc;">
            <h2>Create Survey Questions</h2>
        </div>
    </form>
</body>
</html>
```

```

        <asp:Label ID="lblQuestion" runat="server" Text="Question:"></asp:Label><br />
        <asp:TextBox ID="txtQuestion" runat="server" Width="350px"></asp:TextBox><br /><br />
    />

    <asp:Label ID="lblType" runat="server" Text="Question Type:"></asp:Label><br />
    <asp:DropDownList ID="ddlType" runat="server" Width="150px">
        <asp:ListItem Text="Multiple Choice" Value="MCQ"></asp:ListItem>
        <asp:ListItem Text="True / False" Value="TF"></asp:ListItem>
    </asp:DropDownList><br /><br />

    <asp:Label ID="lblOptions" runat="server" Text="Options (comma separated for
MCQ):"></asp:Label><br />
    <asp:TextBox ID="txtOptions" runat="server" Width="350px"></asp:TextBox><br /><br />
/>

    <asp:Button ID="btnAdd" runat="server" Text="Add Question" OnClick="btnAdd_Click"
/><br /><br />

    <asp:BulletedList ID="bltSurvey" runat="server" DisplayMode="Text"
Width="380px"></asp:BulletedList>
</div>
</form>
</body>
</html>

```

```

Imports System
Imports System.Collections.Generic

Partial Class SurveyBuilder
    Inherits System.Web.UI.Page

    <Serializable()>
    Public Class QuestionItem
        Public Property QuestionText As String
        Public Property QuestionType As String
        Public Property Options As String
    End Class

    Private Property SurveyQuestions As List(Of QuestionItem)
    Get
        If ViewState("SurveyQuestions") Is Nothing Then
            ViewState("SurveyQuestions") = New List(Of QuestionItem)()
        End If
        Return CType(ViewState("SurveyQuestions"), List(Of QuestionItem))
    End Get
    Set(value As List(Of QuestionItem))
        ViewState("SurveyQuestions") = value
    End Set
End Property

    Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As EventArgs)
        Dim questionText As String = txtQuestion.Text.Trim()
        Dim questionType As String = ddlType.SelectedValue
        Dim options As String = txtOptions.Text.Trim()

        If questionText = "" Then
            ' Optionally, add validation message here
            Exit Sub
        End If

        ' For True/False, options can be ignored or set automatically
        If questionType = "TF" Then
            options = "True,False"
        End If
    End Sub

```

```

        Dim q As New QuestionItem()
        q.QuestionText = questionText
        q.QuestionType = questionType
        q.Options = options

        SurveyQuestions.Add(q)
        UpdateList()
        ClearInputs()
    End Sub

    Private Sub UpdateList()
        bltSurvey.Items.Clear()
        For Each q In SurveyQuestions
            bltSurvey.Items.Add("Q: " & q.QuestionText & " | Type: " & q.QuestionType & " | Options: " & q.Options)
        Next
    End Sub

    Private Sub ClearInputs()
        txtQuestion.Text = ""
        txtOptions.Text = ""
    End Sub
End Class

```

Surveyor.aspx

```

<%@ Page Language="vb" AutoEventWireup="false" CodeFile="Surveyor.aspx.vb" Inherits="Surveyor"
%>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Take Survey</title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="width:450px; margin:auto; padding:20px; border:1px solid #ccc;">
            <h2>Survey</h2>
            <asp:Placeholder ID="phSurvey" runat="server"></asp:Placeholder>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit Survey"
OnClick="btnSubmit_Click" /><br /><br />
            <asp:Label ID="lblResult" runat="server" Text="" />
        </div>
    </form>
</body>
</html>

Imports System
Imports System.Collections.Generic
Imports System.Web.UI.WebControls

Partial Class Surveyor
    Inherits System.Web.UI.Page

    <Serializable()>
    Public Class QuestionItem
        Public Property QuestionText As String
        Public Property QuestionType As String
        Public Property Options As String
    End Class

    ' Get questions from ViewState or Session
    Protected Property SurveyQuestions As List(Of QuestionItem)

```

```

Get
    ' For demo, trying ViewState first, but you can also use Session
    If ViewState("SurveyQuestions") Is Nothing Then
        ' You could retrieve questions from DB or Session if needed
        ' For now, create sample questions to avoid error:
        Dim questions As New List(Of QuestionItem) From {
            New QuestionItem With {.QuestionText = "What is your favorite language?",
.QuestionType = "MCQ", .Options = "Python,C#,Java"},
            New QuestionItem With {.QuestionText = "Do you like programming?",
.QuestionType = "TF", .Options = "True,False"}
        }
        ViewState("SurveyQuestions") = questions
    End If
    Return CType(ViewState("SurveyQuestions"), List(Of QuestionItem))
End Get
Set(value As List(Of QuestionItem))
    ViewState("SurveyQuestions") = value
End Set
End Property

Protected Overrides Sub OnInit(e As EventArgs)
    MyBase.OnInit(e)
    GenerateSurveyControls()
End Sub

Private Sub GenerateSurveyControls()
    phSurvey.Controls.Clear()

    For i As Integer = 0 To SurveyQuestions.Count - 1
        Dim q As QuestionItem = SurveyQuestions(i)

        Dim lbl As New Label()
        lbl.Text = "<b>" & (i + 1).ToString() & ". " & q.QuestionText & "</b><br />"
        phSurvey.Controls.Add(lbl)

        Dim rbl As New RadioButtonList()
        rbl.ID = "rbl" & i.ToString()

        If q.QuestionType = "MCQ" Then
            Dim options As String() = q.Options.Split(",c")
            For Each opt As String In options
                rbl.Items.Add(New ListItem(opt.Trim()))
            Next
        ElseIf q.QuestionType = "TF" Then
            rbl.Items.Add(New ListItem("True"))
            rbl.Items.Add(New ListItem("False"))
        End If

        phSurvey.Controls.Add(rbl)
        phSurvey.Controls.Add(New LiteralControl("<br /><br />"))
    Next
End Sub

Protected Sub btnSubmit_Click(sender As Object, e As EventArgs)
    Dim answers As New List(Of String)()

    For i As Integer = 0 To SurveyQuestions.Count - 1
        Dim rbl As RadioButtonList = CType(phSurvey.FindControl("rbl" & i.ToString()),
RadioButtonList)
        If rbl IsNot Nothing AndAlso rbl.SelectedIndex >= 0 Then
            answers.Add(SurveyQuestions(i).QuestionText & " - " & rbl.SelectedValue)
        Else
            answers.Add(SurveyQuestions(i).QuestionText & " - Not Answered")
        End If
    Next

```

```

        lblResult.Text = "<h3>Survey Submitted Successfully!</h3><br />Your Answers:<br />" &
String.Join("<br />", answers)
    End Sub

End Class

```

SurveyResults

```

<%@ Page Language="vb" AutoEventWireup="false" CodeFile="SurveyResults.aspx.vb"
Inherits="SurveyResults" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Survey Results Analysis</title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="width:700px; margin:auto; padding:20px; border:1px solid #ccc;">
            <h2>Survey Results Analysis</h2>

            <asp:Repeater ID="rptQuestions" runat="server">
                <ItemTemplate>
                    <h4>Q<%# Container.ItemIndex + 1 %>: <%# Eval("QuestionText") %> (<%#
Eval("QuestionType") %>)</h4>
                    <asp:BulletedList ID="bltAnswerSummary" runat="server" />
                    <hr />
                </ItemTemplate>
            </asp:Repeater>

            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" />
        </div>
    </form>
</body>
</html>

```

```

Imports System
Imports System.Collections.Generic
Imports System.Web.UI.WebControls

```

```

Partial Class SurveyResults
    Inherits System.Web.UI.Page

    <Serializable()>
    Public Class SurveyQuestion
        Public Property QuestionText As String
        Public Property QuestionType As String ' "Multiple Choice" or "True/False"
        Public Property Options As List(Of String) ' List of options for MCQ
    End Class

    <Serializable()>
    Public Class SurveyResult
        Public Property UserName As String
        Public Property Answers As List(Of String) ' Answer per question in order
    End Class

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Not IsPostBack Then
            LoadSurveyResults()
        End If
    End Sub

```

```

Private Sub LoadSurveyResults()
    Dim questions As List(Of SurveyQuestion) = TryCast(Session("SurveyQuestions"), List(Of
SurveyQuestion))
    Dim results As List(Of SurveyResult) = TryCast(Session("SurveyResults"), List(Of
SurveyResult))
    Dim isAnonymous As Boolean = False
    If Session("IsAnonymous") IsNot Nothing Then
        isAnonymous = CBool(Session("IsAnonymous"))
    End If

    If questions Is Nothing OrElse questions.Count = 0 Then
        lblMessage.Text = "No survey questions found."
        rptQuestions.Visible = False
        Return
    End If

    If results Is Nothing OrElse results.Count = 0 Then
        lblMessage.Text = "No survey responses found."
        rptQuestions.Visible = False
        Return
    End If

    lblMessage.Text = ""
    rptQuestions.Visible = True

    ' Prepare aggregated data for each question
    Dim aggregatedAnswers As New List(Of Dictionary(Of String, Integer)) ' List of
dictionaries per question

    For i As Integer = 0 To questions.Count - 1
        Dim answerCount As New Dictionary(Of String, Integer)

        ' Initialize counts for each option to zero
        For Each opt In questions(i).Options
            answerCount(opt) = 0
        Next

        aggregatedAnswers.Add(answerCount)
    Next

    ' Count answers from all results
    For Each res In results
        For i As Integer = 0 To questions.Count - 1
            Dim answer As String = ""
            If i < res.Answers.Count Then
                answer = res.Answers(i)
            End If

            ' Only count if answer is in options
            If aggregatedAnswers(i).ContainsKey(answer) Then
                aggregatedAnswers(i)(answer) += 1
            Else
                ' For unexpected answers (like blank), ignore or add
                If answer <> "" Then
                    aggregatedAnswers(i)(answer) = 1
                End If
            End If
        Next
    Next

    ' Store aggregated answers in ViewState to bind in repeater ItemDataBound
    ViewState("AggregatedAnswers") = aggregatedAnswers

    rptQuestions.DataSource = questions
    rptQuestions.DataBind()
End Sub

```



```

Protected Sub rptQuestions_ItemDataBound(sender As Object, e As RepeaterItemEventArgs)
Handles rptQuestions.ItemDataBound
    If e.Item.ItemType = ListItemType.Item Or e.Item.ItemType =
ListItemType.AlternatingItem Then
        Dim question As SurveyQuestion = CType(e.Item.DataItem, SurveyQuestion)
        Dim bltAnswerSummary As BulletedList =
CType(e.Item.FindControl("bltAnswerSummary"), BulletedList)

        Dim aggregatedAnswers As List(Of Dictionary(Of String, Integer)) =
TryCast(ViewState("AggregatedAnswers"), List(Of Dictionary(Of String, Integer)))

        If aggregatedAnswers IsNot Nothing Then
            Dim idx As Integer = e.Item.ItemIndex
            Dim counts As Dictionary(Of String, Integer) = aggregatedAnswers(idx)

            bltAnswerSummary.Items.Clear()

            ' Assuming counts is a Dictionary(Of String, Integer)
            For Each kvp As KeyValuePair(Of String, Integer) In counts
                bltAnswerSummary.Items.Add(kvp.Key & ": " & kvp.Value.ToString() & "
responses")
            Next

        End If
    End If
End Sub

End Class

```

SURVEY ADMIN

```

<%@ Page Language="vb" AutoEventWireup="false" CodeFile="SurveyAdmin.aspx.vb"
Inherits="Administrator" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Survey Administrator</title>
</head>
<body>
    <form id="form1" runat="server">
        <div style="width:600px; margin: auto; padding:20px; border:1px solid #ccc;">
            <h2>Survey Administrator Panel</h2>

            <asp:Label ID="lblInfo" runat="server" ForeColor="Blue" Font-
Bold="True"></asp:Label>
            <br /><br />

            <asp:Label ID="lblAnonymity" runat="server" Text="Surveyor
Anonymity:"></asp:Label>
            <asp:DropDownList ID="ddlAnonymity" runat="server">
                <asp:ListItem Text="Anonymous" Value="Anonymous" />
                <asp:ListItem Text="Identified" Value="Identified" />
            </asp:DropDownList>
            <br /><br />

            <asp:Button ID="btnSaveSettings" runat="server" Text="Save Settings"
OnClick="btnSaveSettings_Click" />
            <br /><br />

            <asp:Label ID="lblStatus" runat="server" ForeColor="Green"></asp:Label>
        </div>
    </form>
</body>

```

```

</html>
Imports System

Partial Class Administrator
    Inherits System.Web.UI.Page

    Private Const AnonymitySettingKey As String = "SurveyAnonymitySetting"

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Not IsPostBack Then
            Dim setting As String = If(Application(AnonymitySettingKey), "Anonymous")
            ddlAnonymity.SelectedValue = setting
            lblInfo.Text = "Set whether surveyors' identities are shown or anonymous."
        End If
    End Sub

    Protected Sub btnSaveSettings_Click(sender As Object, e As EventArgs)
        Application.Lock()
        Application(AnonymitySettingKey) = ddlAnonymity.SelectedValue
        Application.Unlock()

        lblStatus.Text = "Settings saved successfully!"
    End Sub

    Public Shared Function GetAnonymitySetting() As String
        Dim setting = CType(HttpContext.Current.Application(AnonymitySettingKey), String)
        Return If(String.IsNullOrEmpty(setting), "Anonymous", setting)
    End Function
End Class

```

DASHBOARD

```

<%@ Page Language="vb" AutoEventWireup="false" CodeFile="Dashboard.aspx.vb"
Inherits="Dashboard" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Dashboard</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 30px; }
        h2 { color: #0078d7; }
        .role-section { border: 1px solid #ccc; padding: 20px; margin-top: 20px; }
        .logout-btn {
            background-color: #e74c3c;
            color: white;
            border: none;
            padding: 10px 20px;
            cursor: pointer;
            font-size: 14px;
            margin-top: 10px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <h1>Welcome, <asp:Label ID="lblUsername" runat="server" /></h1>
        <asp:Button ID="btnLogout" runat="server" Text="Logout" CssClass="logout-btn"
OnClick="btnLogout_Click" />

        <asp:Panel ID="pnlSurveyBuilder" runat="server" CssClass="role-section"
Visible="False">
            <h2>Survey Builder Panel</h2>
            <p>Here you can create and manage surveys.</p>
            <!-- You can add Survey Builder controls here or link to survey builder page -->

```

```

        <asp:HyperLink ID="lnkBuilder" runat="server" NavigateUrl="SurveyBuilder.aspx">Go
to Survey Builder</asp:HyperLink>
    </asp:Panel>

    <asp:Panel ID="pnlSurveyor" runat="server" CssClass="role-section" Visible="False">
        <h2>Surveyor Panel</h2>
        <p>Here you can participate in surveys.</p>
        <!-- Add survey participation controls or link -->
        <asp:HyperLink ID="lnkSurveyor" runat="server" NavigateUrl="Surveyor.aspx">Go to
Surveys</asp:HyperLink>
    </asp:Panel>

    <asp:Panel ID="pnlSurveyAdmin" runat="server" CssClass="role-section" Visible="False">
        <h2>Survey Administrator Panel</h2>
        <p>Manage survey settings and analyze results.</p>
        <!-- Add admin controls or link -->
        <asp:HyperLink ID="lnkAdmin" runat="server" NavigateUrl="SurveyAdmin.aspx">Go to
Survey Admin</asp:HyperLink>
    </asp:Panel>

</form>
</body>
</html>

```

Imports System

Partial Class Dashboard

Inherits System.Web.UI.Page

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

If Not IsPostBack Then

' Check if user is logged in

If Session("Username") Is Nothing OrElse Session("UserRole") Is Nothing Then

' Not logged in, redirect to login

Response.Redirect("Login.aspx")

Return

End If

lblUsername.Text = Session("Username").ToString()

' Show controls based on role

Dim role = CType(Session("UserRole"), Object)

' You can define an enum or check role string, depending on how you store it

' If you stored enum, you can cast to string and compare:

Dim roleName As String = role.ToString().ToLower()

Select Case roleName

Case "surveybuilder"

pnlSurveyBuilder.Visible = True

Case "surveyor"

pnlSurveyor.Visible = True

Case "surveyadmin"

pnlSurveyAdmin.Visible = True

Case Else

' Unknown role, logout

Session.Clear()

Response.Redirect("Login.aspx")

End Select

End If

End Sub

Protected Sub btnLogout_Click(sender As Object, e As EventArgs)

Session.Clear()

Response.Redirect("Login.aspx")

End Sub

End Class