

Code Implementation of Quiz Portal:

Login Page:

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Login</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Login</h2>
        <p>Username: <asp:TextBox ID="txtUsername" runat="server"></asp:TextBox></p>
        <p>Password: <asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox></p>
        <asp:Button ID="btnLogin" runat="server" Text="Login"
OnClick="btnLogin_Click" />
        <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label>
    </form>
</body>
</html>
```

```
Imports System.Data.SqlClient
```

```
Partial Class Login
```

```
    Inherits System.Web.UI.Page
```

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

        Dim conn As New SqlConnection("Data Source=Eman\SQLEXPRESS;Initial
Catalog=OnlineTest;Integrated Security=True;")
        Dim cmd As New SqlCommand("SELECT * FROM Users WHERE Username=@user AND
Password=@pass", conn)
        cmd.Parameters.AddWithValue("@user", username)
        cmd.Parameters.AddWithValue("@pass", password)

        conn.Open()
        Dim reader As SqlDataReader = cmd.ExecuteReader()

        If reader.Read() Then
            Session("UserID") = reader("UserID")
            Session("Role") = reader("Role")

            If reader("Role") = "Teacher" Then
                Response.Redirect("TeacherDashboard.aspx")
            ElseIf reader("Role") = "Student" Then
                Response.Redirect("StudentDashboard.aspx")
            Else
                Response.Redirect("AdminDashboard.aspx")
            End If
        Else
```

```

        lblMessage.Text = "Invalid login"
    End If

    conn.Close()
End Sub
End Class

```

Admin Dashboard:

```

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Admin Dashboard</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Admin Dashboard</h2>

        <p>
            <asp:Button ID="btnCreateUser" runat="server" Text="Add User"
OnClick="btnCreateUser_Click" />
            <asp:Button ID="btnLogout" runat="server" Text="Logout"
OnClick="btnLogout_Click" />
        </p>

        <h3>Users</h3>
        <asp:GridView ID="gvUsers" runat="server" AutoGenerateColumns="false"
OnRowDeleting="gvUsers_RowDeleting" DataKeyNames="UserID" EnableViewState="true">
            <Columns>
                <asp:BoundField DataField="UserID" HeaderText="ID" />
                <asp:BoundField DataField="Username" HeaderText="Username" />
                <asp:BoundField DataField="FullName" HeaderText="Name" />
                <asp:BoundField DataField="Role" HeaderText="Role" />
                <asp:CommandField ShowDeleteButton="true" />
            </Columns>
        </asp:GridView>

        <h3>Quizzes</h3>
        <asp:GridView ID="gvQuizzes" runat="server" AutoGenerateColumns="false"
OnRowDeleting="gvQuizzes_RowDeleting" DataKeyNames="QuizID"
EnableViewState="true">
            <Columns>
                <asp:BoundField DataField="QuizID" HeaderText="ID" />
                <asp:BoundField DataField="Title" HeaderText="Title" />
                <asp:BoundField DataField="FullName" HeaderText="Created By" />
                <asp:CommandField ShowDeleteButton="true" />
            </Columns>
        </asp:GridView>

        <p>

```

```

        Total Users: <asp:Label ID="lblUserCount" runat="server"></asp:Label><br />
        Total Quizzes: <asp:Label ID="lblQuizCount" runat="server"></asp:Label>
    </p>
</form>
</body>
</html>

```

```

Imports System.Data
Imports System.Data.SqlClient

```

```

Partial Class AdminDashboard
    Inherits System.Web.UI.Page

```

```

    Dim connString As String =
    ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString

```

```

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing OrElse Session("Role").ToString() <>
                "Administrator" Then
                    Response.Redirect("Login.aspx")
                End If
            LoadUsers() ' Only bind data on initial load, not postbacks
        End If
    End Sub

```

```

    Sub LoadUsers()
        Dim conn As New SqlConnection(connString)
        Dim cmd As New SqlCommand("SELECT UserID, Username, FullName, Role FROM
Users", conn)

```

```

        Dim adapter As New SqlDataAdapter(cmd)
        Dim dt As New DataTable()
        adapter.Fill(dt)

```

```

        gvUsers.DataSource = dt
        gvUsers.DataBind()

```

```

        conn.Close()
    End Sub

```

```

    Sub LoadQuizzes()
        Dim conn As New SqlConnection(connString)
        Dim cmd As New SqlCommand("SELECT q.QuizID, q.Title, u.FullName FROM Quizzes
q JOIN Users u ON q.CreatedBy = u.UserID", conn)

```

```

        Dim adapter As New SqlDataAdapter(cmd)
        Dim dt As New DataTable()
        adapter.Fill(dt)

```

```

        gvQuizzes.DataSource = dt
        gvQuizzes.DataBind()

```

```

        conn.Close()
    End Sub

```

```

Sub LoadCounts()
    Dim conn As New SqlConnection(connString)
    conn.Open()

    Dim cmd1 As New SqlCommand("SELECT COUNT(*) FROM Users", conn)
    lblUserCount.Text = cmd1.ExecuteScalar().ToString()

    Dim cmd2 As New SqlCommand("SELECT COUNT(*) FROM Quizzes", conn)
    lblQuizCount.Text = cmd2.ExecuteScalar().ToString()

    conn.Close()
End Sub

Protected Sub gvUsers_RowDeleting(sender As Object, e As
GridViewDeleteEventArgs)
    ' First check if GridView has data
    If gvUsers.Rows.Count = 0 Then
        ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('No
users to delete.');" , True)
        Return
    End If

    ' Check row index is valid
    If e.RowIndex < 0 OrElse e.RowIndex >= gvUsers.Rows.Count Then
        ClientScript.RegisterStartupScript(Me.GetType(), "alert",
"alert('Invalid row selection.');" , True)
        Return
    End If

    ' Safely get UserID
    If gvUsers.DataKeys.Count <= e.RowIndex Then
        LoadUsers() ' Refresh data if keys are out of sync
        ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('Please
try deleting again.');" , True)
        Return
    End If

    Dim userID As Integer = Convert.ToInt32(gvUsers.DataKeys(e.RowIndex).Value)

    ' Check if admin is trying to delete themselves
    Dim currentUserID As Integer = Convert.ToInt32(Session("UserID"))
    If userID = currentUserID Then
        ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('You
cannot delete your own account!');" , True)
        Return
    End If

    ' Delete from database
    Dim conn As New SqlConnection(connString)
    Try
        conn.Open()

        ' First delete dependent records
        Dim deleteDependencies As String = "DELETE FROM StudentAnswers WHERE
StudentID = @id; " &
                                           "DELETE FROM QuizResults WHERE StudentID =
@id; " &

```

```

                                "UPDATE Quizzes SET CreatedBy = NULL WHERE
CreatedBy = @id;"
    Using depCmd As New SqlCommand(deleteDependencies, conn)
        depCmd.Parameters.AddWithValue("@id", userID)
        depCmd.ExecuteNonQuery()
    End Using

    ' Then delete the user
    Using userCmd As New SqlCommand("DELETE FROM Users WHERE UserID = @id",
conn)
        userCmd.Parameters.AddWithValue("@id", userID)
        userCmd.ExecuteNonQuery()
    End Using

    ' Refresh the data
    LoadUsers()
    LoadCounts()

    ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('User
deleted successfully.');" , True)
    Catch ex As Exception
        ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('Error
deleting user: " & ex.Message.Replace("'", "\"") & "');" , True)
    Finally
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
    End Try
End Sub

Protected Sub gvQuizzes_RowDeleting(sender As Object, e As
GridViewDeleteEventArgs)
    Dim quizID As Integer =
Convert.ToInt32(gvQuizzes.DataKeys(e.RowIndex).Value)

    Dim conn As New SqlConnection(connString)
    Try
        conn.Open()

        ' Delete questions first
    Using cmd1 As New SqlCommand("DELETE FROM Questions WHERE QuizID = @id",
conn)
        cmd1.Parameters.AddWithValue("@id", quizID)
        cmd1.ExecuteNonQuery()
    End Using

        ' Then delete quiz
    Using cmd2 As New SqlCommand("DELETE FROM Quizzes WHERE QuizID = @id",
conn)
        cmd2.Parameters.AddWithValue("@id", quizID)
        cmd2.ExecuteNonQuery()
    End Using

        ' Refresh the data
    LoadQuizzes()
    LoadCounts()

```

```

        ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('Quiz
deleted successfully.');" , True)
    Catch ex As Exception
        ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('Error
deleting quiz: " & ex.Message.Replace("'", "\'") & "');" , True)
    Finally
        If conn.State = ConnectionState.Open Then
            conn.Close()
        End If
    End Try
End Sub

Protected Sub btnCreateUser_Click(sender As Object, e As EventArgs)
    Response.Redirect("CreateUser.aspx")
End Sub

Protected Sub btnLogout_Click(sender As Object, e As EventArgs)
    Session.Clear()
    Response.Redirect("Login.aspx")
End Sub
End Class

```

Create Users:

```

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Create New User</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Create New User</h2>
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label>

            <div>
                <asp:Label ID="lblUsername" runat="server" Text="Username:"
AssociatedControlID="txtUsername"></asp:Label>
                <asp:TextBox ID="txtUsername" runat="server"
required="true"></asp:TextBox>
            </div>

            <div>
                <asp:Label ID="lblPassword" runat="server" Text="Password:"
AssociatedControlID="txtPassword"></asp:Label>
                <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
required="true"></asp:TextBox>
            </div>

            <div>
                <asp:Label ID="lblFullName" runat="server" Text="Full Name:"
AssociatedControlID="txtFullName"></asp:Label>

```

```

        <asp:TextBox ID="txtFullName" runat="server"
required="true"></asp:TextBox>
    </div>

    <div>
        <asp:Label ID="lblRole" runat="server" Text="Role:"
AssociatedControlID="ddlRole"></asp:Label>
        <asp:DropDownList ID="ddlRole" runat="server" required="true">
            <asp:ListItem Value="">Select Role</asp:ListItem>
            <asp:ListItem Value="Administrator">Administrator</asp:ListItem>
            <asp:ListItem Value="Teacher">Teacher</asp:ListItem>
            <asp:ListItem Value="Student">Student</asp:ListItem>
        </asp:DropDownList>
    </div>

    <div>
        <asp:Button ID="btnCreate" runat="server" Text="Create User"
OnClick="btnCreate_Click" />
        <asp:Button ID="btnCancel" runat="server" Text="Cancel"
OnClick="btnCancel_Click" CausesValidation="false" />
    </div>
</div>
</form>
</body>
</html>

```

```
Imports System.Data.SqlClient
```

```
Partial Class CreateUser
    Inherits System.Web.UI.Page
```

```
    Private ReadOnly connectionString As String =
ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString
```

```
    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Session("UserID") Is Nothing OrElse Session("Role").ToString() <>
"Administrator" Then
            Response.Redirect("Login.aspx")
        End If
    End Sub
```

```
    Protected Sub btnCreate_Click(sender As Object, e As EventArgs)
        Dim username As String = txtUsername.Text.Trim()
        Dim password As String = txtPassword.Text
        Dim fullName As String = txtFullName.Text.Trim()
        Dim role As String = ddlRole.SelectedValue

        ' Validate input
        If String.IsNullOrEmpty(username) OrElse String.IsNullOrEmpty(password)
OrElse
        String.IsNullOrEmpty(fullName) OrElse String.IsNullOrEmpty(role) Then
            lblMessage.Text = "All fields are required."
            Return
        End If

        Using conn As New SqlConnection(connectionString)
            ' Check if username already exists

```

```

Dim checkQuery As String = "SELECT COUNT(*) FROM Users WHERE Username =
@Username"
Dim checkCmd As New SqlCommand(checkQuery, conn)
checkCmd.Parameters.AddWithValue("@Username", username)

conn.Open()
Dim exists As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())

If exists > 0 Then
    lblMessage.Text = "Username already exists. Please choose a
different username."
    Return
End If

' Create new user
Dim insertQuery As String = "INSERT INTO Users (Username, Password,
FullName, Role) " &
"VALUES (@Username, @Password, @FullName,
@Role)"

Dim insertCmd As New SqlCommand(insertQuery, conn)
insertCmd.Parameters.AddWithValue("@Username", username)
insertCmd.Parameters.AddWithValue("@Password", password)
insertCmd.Parameters.AddWithValue("@FullName", fullName)
insertCmd.Parameters.AddWithValue("@Role", role)

insertCmd.ExecuteNonQuery()
End Using

Response.Redirect("AdminDashboard.aspx")
End Sub

Protected Sub btnCancel_Click(sender As Object, e As EventArgs)
    Response.Redirect("AdminDashboard.aspx")
End Sub
End Class

```

Student Dashboard:

```

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Student Dashboard</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Student Dashboard</h2>
        <h3>Available Quizzes</h3>
        <asp:GridView ID="gvQuizzes" runat="server" AutoGenerateColumns="False">
            <Columns>
                <asp:BoundField DataField="QuizID" HeaderText="ID" />
                <asp:BoundField DataField="Title" HeaderText="Title" />
                <asp:BoundField DataField="Description" HeaderText="Description" />
                <asp:TemplateField>

```

```

        <ItemTemplate>
            <asp:Button ID="btnTake" runat="server" Text="Take Quiz"
                CommandArgument='<%# Eval("QuizID") %>' OnClick="btnTake_Click"
            />
        </ItemTemplate>
    </asp:TemplateField>
</Columns>
<EmptyDataTemplate>
    No active quizzes available.
</EmptyDataTemplate>
</asp:GridView>
</form>
</body>
</html>

```

```

Imports System.Data
Imports System.Data.SqlClient

```

```

Partial Class StudentDashboard
    Inherits System.Web.UI.Page

```

```

    Private ReadOnly connectionString As String =
        ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString

```

```

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing OrElse Session("Role").ToString() <>
                "Student" Then
                    Response.Redirect("Login.aspx")
                End If
            LoadQuizzes()
        End If
    End Sub

```

```

    Private Sub LoadQuizzes()
        Try
            Using conn As New SqlConnection(connectionString)
                Dim query As String = "SELECT QuizID, Title, Description FROM
Quizzes WHERE IsActive = 1"
                Dim cmd As New SqlCommand(query, conn)

                Dim adapter As New SqlDataAdapter(cmd)
                Dim dt As New DataTable()
                adapter.Fill(dt)

                If dt.Rows.Count > 0 Then
                    gvQuizzes.DataSource = dt
                    gvQuizzes.DataBind()
                Else
                    ' Add empty row to show message
                    dt.Rows.Add(dt.NewRow())
                    gvQuizzes.DataSource = dt
                    gvQuizzes.DataBind()
                    gvQuizzes.Rows(0).Cells.Clear()
                    gvQuizzes.Rows(0).Cells.Add(New TableCell())
                    gvQuizzes.Rows(0).Cells(0).ColumnSpan = 4
                End If
            End Using
        Catch ex As Exception
            ' Handle exception
        End Try
    End Sub

```

```

        gvQuizzes.Rows(0).Cells(0).Text = "No active quizzes available"
        gvQuizzes.Rows(0).Cells(0).HorizontalAlign =
HorizontalAlign.Center
        End If
    End Using
    Catch ex As Exception
        ' Log error and show message
        Response.Write("<script>alert('Error loading quizzes: " &
ex.Message.Replace("'", "\"") & "');</script>")
    End Try
End Sub

Protected Sub btnTake_Click(sender As Object, e As EventArgs)
    Dim btn As Button = CType(sender, Button)
    Response.Redirect("AttemptQuiz.aspx?QuizID=" & btn.CommandArgument)
End Sub
End Class

```

Attempt Quiz:

```

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Attempt Quiz</title>
    <style>
        .question-container {
            margin-bottom: 20px;
            padding: 15px;
            border: 1px solid #ddd;
            border-radius: 5px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2><asp:Label ID="lblQuizTitle" runat="server"
Text=""></asp:Label></h2>
            <asp:Label ID="lblQuizDescription" runat="server" Text=""></asp:Label>
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"
Visible="false"></asp:Label>

            <asp:Panel ID="pnlQuestions" runat="server">
                <asp:Repeater ID="rptQuestions" runat="server"
OnItemDataBound="rptQuestions_ItemDataBound">
                    <ItemTemplate>
                        <div class="question-container">
                            <asp:HiddenField ID="hfQuestionID" runat="server"
Value='<%# Eval("QuestionID") %>' />
                            <h4><%# Container.ItemIndex + 1 %>. <%#
Eval("QuestionText") %></h4>

```

```

                <asp:RadioButtonList ID="rblOptions" runat="server">
                </asp:RadioButtonList>
            </div>
        </ItemTemplate>
    </asp:Repeater>

    <asp:Button ID="btnSubmit" runat="server" Text="Submit Quiz"
OnClick="btnSubmit_Click" />
    </asp:Panel>
</div>
</form>
</body>
</html>

```

```

Imports System.Data
Imports System.Data.SqlClient

```

```

Partial Class AttemptQuiz
    Inherits System.Web.UI.Page

```

```

    Private ReadOnly connectionString As String =
ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString
    Private quizID As Integer
    Private studentID As Integer

```

```

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Session("UserID") Is Nothing OrElse Session("Role").ToString() <>
"Student" Then
            Response.Redirect("Login.aspx")
        End If

        studentID = Convert.ToInt32(Session("UserID"))

        If Not IsPostBack Then
            If Not Integer.TryParse(Request.QueryString("QuizID"), quizID) Then
                Response.Redirect("StudentDashboard.aspx")
            End If

            ' Store QuizID in session for verification
            Session("CurrentQuizID") = quizID

            LoadQuizDetails()
            LoadQuestions()
        End If
    End Sub

```

```

    Private Sub LoadQuizDetails()
        Using conn As New SqlConnection(connectionString)
            Dim query As String = "SELECT Title, Description FROM Quizzes WHERE
QuizID = @QuizID AND IsActive = 1"
            Dim cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@QuizID", quizID)

            conn.Open()
            Using reader As SqlDataReader = cmd.ExecuteReader()
                If reader.Read() Then
                    lblQuizTitle.Text = reader("Title").ToString()

```

```

        lblQuizDescription.Text = reader("Description").ToString()
    Else
        Response.Redirect("StudentDashboard.aspx")
    End If
End Using
End Using
End Sub

Private Sub LoadQuestions()
    Using conn As New SqlConnection(connectionString)
        ' Get questions in random order
        Dim query As String = "SELECT QuestionID, QuestionText, Option1,
Option2, Option3, Option4 " &
                                "FROM Questions WHERE QuizID = @QuizID ORDER BY
NEWID()"
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@QuizID", quizID)

        conn.Open()
        Using reader As SqlDataReader = cmd.ExecuteReader()
            If reader.HasRows Then
                rptQuestions.DataSource = reader
                rptQuestions.DataBind()
            Else
                pnlQuestions.Visible = False
                lblMessage.Text = "No questions available for this quiz."
                lblMessage.Visible = True
            End If
        End Using
    End Using
End Sub

Protected Sub rptQuestions_ItemDataBound(sender As Object, e As
RepeaterItemEventArgs)
    If e.Item.ItemType = ListItemType.Item OrElse e.Item.ItemType =
ListItemType.AlternatingItem Then
        Dim rblOptions As RadioButtonList =
CType(e.Item.FindControl("rblOptions"), RadioButtonList)
        Dim record As System.Data.Common.DbDataRecord = CType(e.Item.DataItem,
System.Data.Common.DbDataRecord)

        ' Clear any existing items
        rblOptions.Items.Clear()

        ' Add options to RadioButtonList
        rblOptions.Items.Add(New ListItem(record("Option1").ToString(), "1"))
        rblOptions.Items.Add(New ListItem(record("Option2").ToString(), "2"))

        If Not record.IsDBNull(record.GetOrdinal("Option3")) AndAlso Not
String.IsNullOrEmpty(record("Option3").ToString()) Then
            rblOptions.Items.Add(New ListItem(record("Option3").ToString(),
"3"))
        End If

        If Not record.IsDBNull(record.GetOrdinal("Option4")) AndAlso Not
String.IsNullOrEmpty(record("Option4").ToString()) Then
            rblOptions.Items.Add(New ListItem(record("Option4").ToString(),
"4"))
        End If
    End If
End Sub

```

```

        End If
    End If
End Sub

Protected Sub btnSubmit_Click(sender As Object, e As EventArgs)
    ' Save answers and calculate score
    Dim score As Integer = 0
    Dim totalQuestions As Integer = 0

    Using conn As New SqlConnection(connectionString)
        conn.Open()
        Dim transaction As SqlTransaction = conn.BeginTransaction()
        Dim isTransactionCommitted As Boolean = False

        Try
            ' Verify quiz exists and is active
            Using verifyCmd As New SqlCommand("SELECT Title, IsActive FROM
Quizzes WHERE QuizID = @QuizID", conn, transaction)
                verifyCmd.Parameters.AddWithValue("@QuizID", quizID)
                Using reader As SqlDataReader = verifyCmd.ExecuteReader()
                    If Not reader.Read() Then
                        Throw New Exception("This quiz has been removed from the
system.")
                    ElseIf Not Convert.ToBoolean(reader("IsActive")) Then
                        Throw New Exception("This quiz has been deactivated by
your instructor.")
                    End If
                End Using
            End Using

            ' Process each question
            For Each item As RepeaterItem In rptQuestions.Items
                If item.ItemType = ListItemType.Item OrElse item.ItemType =
ListItemType.AlternatingItem Then
                    totalQuestions += 1

                    Dim hfQuestionID As HiddenField =
CType(item.FindControl("hfQuestionID"), HiddenField)
                    Dim rblOptions As RadioButtonList =
CType(item.FindControl("rblOptions"), RadioButtonList)

                    If hfQuestionID Is Nothing OrElse rblOptions Is Nothing Then
                        Continue For
                    End If

                    Dim questionID As Integer =
Convert.ToInt32(hfQuestionID.Value)
                    Dim selectedAnswer As Integer = If(rblOptions.SelectedIndex
>= 0, rblOptions.SelectedIndex + 1, 0)

                    ' Save student answer
                    Using insertCmd As New SqlCommand(
                        "INSERT INTO StudentAnswers (StudentID, QuestionID,
SelectedAnswer) " &
                        "VALUES (@StudentID, @QuestionID, @SelectedAnswer)",
conn, transaction)

```

```

        insertCmd.Parameters.AddWithValue("@StudentID",
studentID)
        insertCmd.Parameters.AddWithValue("@QuestionID",
questionID)
        insertCmd.Parameters.AddWithValue("@SelectedAnswer",
selectedAnswer)
        insertCmd.ExecuteNonQuery()
    End Using

    ' Check if answer is correct
    Using correctAnswerCmd As New SqlCommand(
        "SELECT CorrectAnswer FROM Questions WHERE QuestionID =
@QuestionID", conn, transaction)

        correctAnswerCmd.Parameters.AddWithValue("@QuestionID",
questionID)

        Dim correctAnswer As Integer =
Convert.ToInt32(correctAnswerCmd.ExecuteScalar())

        If selectedAnswer = correctAnswer Then
            score += 1
        End If
    End Using
End If
Next

' Save quiz result
Using resultCmd As New SqlCommand(
    "INSERT INTO QuizResults (StudentID, QuizID, Score,
TotalQuestions) " &
    "VALUES (@StudentID, @QuizID, @Score, @TotalQuestions)", conn,
transaction)

    resultCmd.Parameters.AddWithValue("@StudentID", studentID)
    resultCmd.Parameters.AddWithValue("@QuizID", quizID)
    resultCmd.Parameters.AddWithValue("@Score", score)
    resultCmd.Parameters.AddWithValue("@TotalQuestions",
totalQuestions)

    resultCmd.ExecuteNonQuery()
End Using

transaction.Commit()
isTransactionCommitted = True
Response.Redirect("QuizResult.aspx?QuizID=" & quizID.ToString())
Catch ex As Exception
    If Not isTransactionCommitted Then
        Try
            transaction.Rollback()
        Catch rollbackEx As Exception
            ' Log rollback error if needed
            System.Diagnostics.Debug.WriteLine("Rollback failed: " &
rollbackEx.Message)
        End Try
    End If

    ' More detailed error handling
    Dim errorMessage As String = "Error saving quiz results: " &
ex.Message

```

```

        If ex.Message.Contains("removed") OrElse
ex.Message.Contains("deactivated") Then
            errorMessage = ex.Message & " Please contact your instructor."
        End If

        ClientScript.RegisterStartupScript(Me.GetType(), "alert",
            "alert('" & errorMessage.Replace("'", "\"") & "');", True)
    End Try
End Using
End Sub
End Class

```

Teacher Dashboard:

```

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Student Dashboard</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Student Dashboard</h2>
        <h3>Available Quizzes</h3>
        <asp:GridView ID="gvQuizzes" runat="server" AutoGenerateColumns="False">
            <Columns>
                <asp:BoundField DataField="QuizID" HeaderText="ID" />
                <asp:BoundField DataField="Title" HeaderText="Title" />
                <asp:BoundField DataField="Description" HeaderText="Description" />
                <asp:TemplateField>
                    <ItemTemplate>
                        <asp:Button ID="btnTake" runat="server" Text="Take Quiz"
                            CommandArgument='<%# Eval("QuizID") %>' OnClick="btnTake_Click"
                        />
                    </ItemTemplate>
                </asp:TemplateField>
            </Columns>
            <EmptyDataTemplate>
                No active quizzes available.
            </EmptyDataTemplate>
        </asp:GridView>
    </form>
</body>
</html>

```

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class StudentDashboard
    Inherits System.Web.UI.Page

```

```

Private ReadOnly connectionString As String =
ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString

Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
    If Not IsPostBack Then
        If Session("UserID") Is Nothing OrElse Session("Role").ToString() <>
"Student" Then
            Response.Redirect("Login.aspx")
        End If
        LoadQuizzes()
    End If
End Sub

Private Sub LoadQuizzes()
    Try
        Using conn As New SqlConnection(connectionString)
            Dim query As String = "SELECT QuizID, Title, Description FROM
Quizzes WHERE IsActive = 1"
            Dim cmd As New SqlCommand(query, conn)

            Dim adapter As New SqlDataAdapter(cmd)
            Dim dt As New DataTable()
            adapter.Fill(dt)

            If dt.Rows.Count > 0 Then
                gvQuizzes.DataSource = dt
                gvQuizzes.DataBind()
            Else
                ' Add empty row to show message
                dt.Rows.Add(dt.NewRow())
                gvQuizzes.DataSource = dt
                gvQuizzes.DataBind()
                gvQuizzes.Rows(0).Cells.Clear()
                gvQuizzes.Rows(0).Cells.Add(New TableCell())
                gvQuizzes.Rows(0).Cells(0).ColumnSpan = 4
                gvQuizzes.Rows(0).Cells(0).Text = "No active quizzes available"
                gvQuizzes.Rows(0).Cells(0).HorizontalAlign =
HorizontalAlign.Center
            End If
        End Using
        Catch ex As Exception
            ' Log error and show message
            Response.Write("<script>alert('Error loading quizzes: " &
ex.Message.Replace("'", "\"") & "');</script>")
        End Try
    End Sub

Protected Sub btnTake_Click(sender As Object, e As EventArgs)
    Dim btn As Button = CType(sender, Button)
    Response.Redirect("AttemptQuiz.aspx?QuizID=" & btn.CommandArgument)
End Sub
End Class

```

Create Quiz:

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Create New User</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Create New User</h2>
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label>

            <div>
                <asp:Label ID="lblUsername" runat="server" Text="Username:"
AssociatedControlID="txtUsername"></asp:Label>
                <asp:TextBox ID="txtUsername" runat="server"
required="true"></asp:TextBox>
            </div>

            <div>
                <asp:Label ID="lblPassword" runat="server" Text="Password:"
AssociatedControlID="txtPassword"></asp:Label>
                <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
required="true"></asp:TextBox>
            </div>

            <div>
                <asp:Label ID="lblFullName" runat="server" Text="Full Name:"
AssociatedControlID="txtFullName"></asp:Label>
                <asp:TextBox ID="txtFullName" runat="server"
required="true"></asp:TextBox>
            </div>

            <div>
                <asp:Label ID="lblRole" runat="server" Text="Role:"
AssociatedControlID="ddlRole"></asp:Label>
                <asp:DropDownList ID="ddlRole" runat="server" required="true">
                    <asp:ListItem Value="">Select Role</asp:ListItem>
                    <asp:ListItem Value="Administrator">Administrator</asp:ListItem>
                    <asp:ListItem Value="Teacher">Teacher</asp:ListItem>
                    <asp:ListItem Value="Student">Student</asp:ListItem>
                </asp:DropDownList>
            </div>

            <div>
                <asp:Button ID="btnCreate" runat="server" Text="Create User"
OnClick="btnCreate_Click" />
                <asp:Button ID="btnCancel" runat="server" Text="Cancel"
OnClick="btnCancel_Click" CausesValidation="false" />
            </div>
        </div>
    </form>
</body>
```

```
</html>
```

```
Imports System.Data.SqlClient
```

```
Partial Class CreateUser
```

```
    Inherits System.Web.UI.Page
```

```
    Private ReadOnly connectionString As String =  
ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString
```

```
    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load  
        If Session("UserID") Is Nothing OrElse Session("Role").ToString() <>  
"Administrator" Then  
            Response.Redirect("Login.aspx")  
        End If  
    End Sub
```

```
    Protected Sub btnCreate_Click(sender As Object, e As EventArgs)  
        Dim username As String = txtUsername.Text.Trim()  
        Dim password As String = txtPassword.Text  
        Dim fullName As String = txtFullName.Text.Trim()  
        Dim role As String = ddlRole.SelectedValue  
  
        ' Validate input  
        If String.IsNullOrEmpty(username) OrElse String.IsNullOrEmpty(password)  
OrElse  
            String.IsNullOrEmpty(fullName) OrElse String.IsNullOrEmpty(role) Then  
                lblMessage.Text = "All fields are required."  
                Return  
            End If
```

```
        Using conn As New SqlConnection(connectionString)  
            ' Check if username already exists  
            Dim checkQuery As String = "SELECT COUNT(*) FROM Users WHERE Username =  
@Username"  
  
            Dim checkCmd As New SqlCommand(checkQuery, conn)  
            checkCmd.Parameters.AddWithValue("@Username", username)  
  
            conn.Open()  
            Dim exists As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())  
  
            If exists > 0 Then  
                lblMessage.Text = "Username already exists. Please choose a  
different username."  
                Return  
            End If  
  
            ' Create new user  
            Dim insertQuery As String = "INSERT INTO Users (Username, Password,  
FullName, Role) " &  
                "VALUES (@Username, @Password, @FullName,  
@Role)"  
  
            Dim insertCmd As New SqlCommand(insertQuery, conn)  
            insertCmd.Parameters.AddWithValue("@Username", username)  
            insertCmd.Parameters.AddWithValue("@Password", password)  
            insertCmd.Parameters.AddWithValue("@FullName", fullName)  
            insertCmd.Parameters.AddWithValue("@Role", role)
```

```

        insertCmd.ExecuteNonQuery()
    End Using

    Response.Redirect("AdminDashboard.aspx")
End Sub

Protected Sub btnCancel_Click(sender As Object, e As EventArgs)
    Response.Redirect("AdminDashboard.aspx")
End Sub
End Class

```

Manage Quiz:

```

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Manage Questions</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Manage Questions for Quiz ID: <asp:Label ID="lblQuizID"
runat="server"></asp:Label></h2>

            <asp:GridView ID="gvQuestions" runat="server"
AutoGenerateColumns="false" DataKeyNames="QuestionID"
OnRowCommand="gvQuestions_RowCommand" OnRowDeleting="gvQuestions_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="DisplayOrder" HeaderText="Order" />
                    <asp:BoundField DataField="QuestionText" HeaderText="Question" />
                    <asp:TemplateField>
                        <ItemTemplate>
                            <asp:Button ID="btnUp" runat="server" Text="↑" CommandName="MoveUp"
                                CommandArgument='<%# Eval("QuestionID") %>' />
                            <asp:Button ID="btnDown" runat="server" Text="↓"
CommandName="MoveDown"
                                CommandArgument='<%# Eval("QuestionID") %>' />
                        </ItemTemplate>
                    </asp:TemplateField>
                    <asp:CommandField ShowEditButton="true" ShowDeleteButton="true" />
                </Columns>
            </asp:GridView>

            <h3>Add New Question</h3>
            <asp:TextBox ID="txtNewQuestion" runat="server" TextMode="MultiLine" Rows="3"
Width="300"
                placeholder="Enter question text"></asp:TextBox><br />

            <asp:TextBox ID="txtOption1" runat="server" placeholder="Option 1
(required)"></asp:TextBox><br />
            <asp:TextBox ID="txtOption2" runat="server" placeholder="Option 2
(required)"></asp:TextBox><br />

```

```

<asp:TextBox ID="txtOption3" runat="server" placeholder="Option 3
(optional)"></asp:TextBox><br />
<asp:TextBox ID="txtOption4" runat="server" placeholder="Option 4
(optional)"></asp:TextBox><br />

<asp:DropDownList ID="ddlCorrectAnswer" runat="server">
    <asp:ListItem Value="1">Option 1 is correct</asp:ListItem>
    <asp:ListItem Value="2">Option 2 is correct</asp:ListItem>
    <asp:ListItem Value="3">Option 3 is correct</asp:ListItem>
    <asp:ListItem Value="4">Option 4 is correct</asp:ListItem>
</asp:DropDownList><br />

<asp:Button ID="btnAddQuestion" runat="server" Text="Add Question"
OnClick="btnAddQuestion_Click" />

    <asp:Button ID="btnBack" runat="server" Text="Back to Dashboard"
OnClick="btnBack_Click" />
    <asp:Label ID="lblMessage" runat="server" CssClass="message"
Visible="false"></asp:Label>
</div>
</form>
</body>
</html>

Imports System.Data
Imports System.Data.SqlClient

Partial Class ManageQuestions
    Inherits System.Web.UI.Page

    Private ReadOnly connectionString As String =
ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString
    Private quizID As Integer

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Not IsPostBack Then
            If Not Integer.TryParse(Request.QueryString("QuizID"), quizID) Then
                Response.Redirect("TeacherDashboard.aspx")
            End If

            ' Check for success message from CreateQuiz
            If Not String.IsNullOrEmpty(Request.QueryString("success")) Then
                lblMessage.Text = "Quiz created successfully!"
                lblMessage.Visible = True
            End If

            lblQuizID.Text = quizID.ToString()
            LoadQuestions()
        End If
    End Sub

    Protected Sub btnBack_Click(sender As Object, e As EventArgs)
        Response.Redirect("TeacherDashboard.aspx")
    End Sub

    Private Sub LoadQuestions()

```

```

        Using conn As New
SqlConnection(ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString)
        Dim query As String = "SELECT QuestionID, QuestionText, DisplayOrder
FROM Questions " &
                                "WHERE QuizID = @QuizID ORDER BY DisplayOrder"
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@QuizID", quizID)

        Dim adapter As New SqlDataAdapter(cmd)
        Dim dt As New DataTable()
        adapter.Fill(dt)

        gvQuestions.DataSource = dt
        gvQuestions.DataBind()
    End Using
End Sub

```

```

Protected Sub btnAddQuestion_Click(sender As Object, e As EventArgs)
    Dim questionText As String = txtNewQuestion.Text.Trim()
    Dim option1 As String = txtOption1.Text.Trim()
    Dim option2 As String = txtOption2.Text.Trim()
    Dim quizID As Integer = Integer.Parse(lblQuizID.Text)

    ' Calculate next display order
    Dim nextDisplayOrder As Integer = GetNextDisplayOrder(quizID)

    ' Validate required fields
    If String.IsNullOrEmpty(questionText) OrElse
String.IsNullOrEmpty(option1) OrElse
String.IsNullOrEmpty(option2) Then
        ClientScript.RegisterStartupScript(Me.GetType(), "alert",
"alert('Question text and at least 2 options are required.');" , True)
        Return
    End If

```

```

        Using conn As New
SqlConnection(ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString)
        Dim query As String = "INSERT INTO Questions (QuizID, QuestionText,
Option1, Option2, Option3, Option4, CorrectAnswer, DisplayOrder) " &
                                "VALUES (@QuizID, @QuestionText, @Option1, @Option2,
@Option3, @Option4, @CorrectAnswer, @DisplayOrder)"
        Dim cmd As New SqlCommand(query, conn)

        cmd.Parameters.AddWithValue("@QuizID", quizID)
        cmd.Parameters.AddWithValue("@QuestionText", questionText)
        cmd.Parameters.AddWithValue("@Option1", option1)
        cmd.Parameters.AddWithValue("@Option2", option2)
        cmd.Parameters.AddWithValue("@Option3",
If(String.IsNullOrEmpty(txtOption3.Text.Trim()), DBNull.Value,
txtOption3.Text.Trim()))
        cmd.Parameters.AddWithValue("@Option4",
If(String.IsNullOrEmpty(txtOption4.Text.Trim()), DBNull.Value,
txtOption4.Text.Trim()))
        cmd.Parameters.AddWithValue("@CorrectAnswer",
ddlCorrectAnswer.SelectedValue)
        cmd.Parameters.AddWithValue("@DisplayOrder", nextDisplayOrder)

        Try

```

```

        conn.Open()
        cmd.ExecuteNonQuery()

        ' Clear fields after successful insert
        ClearQuestionForm()
        LoadQuestions()

        ClientScript.RegisterStartupScript(Me.GetType(), "alert",
"alert('Question added successfully!');", True)
        Catch ex As Exception
            ClientScript.RegisterStartupScript(Me.GetType(), "alert",
"alert('Error adding question: " & ex.Message.Replace("'", "\"") & "');", True)
        End Try
    End Using
End Sub

Private Function GetNextDisplayOrder(quizID As Integer) As Integer
    Using conn As New
        SqlConnection(ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString)
        Dim query As String = "SELECT ISNULL(MAX(DisplayOrder), 0) + 1 FROM
Questions WHERE QuizID = @QuizID"
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@QuizID", quizID)

        conn.Open()
        Return Convert.ToInt32(cmd.ExecuteScalar())
    End Using
End Function

Protected Sub gvQuestions_RowCommand(sender As Object, e As
GridViewCommandEventArgs)
    If e.CommandName = "MoveUp" OrElse e.CommandName = "MoveDown" Then
        Dim questionID As Integer = Convert.ToInt32(e.CommandArgument)
        ReorderQuestion(questionID, e.CommandName = "MoveUp")
        LoadQuestions()
    End If
End Sub

Protected Sub gvQuestions_RowDeleting(sender As Object, e As
GridViewDeleteEventArgs)
    Dim questionID As Integer =
Convert.ToInt32(gvQuestions.DataKeys(e.RowIndex).Value)

    Using conn As New
        SqlConnection(ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString)
        Try
            conn.Open()

            Dim deleteAnswersCmd As New SqlCommand("DELETE FROM StudentAnswers
WHERE QuestionID = @QuestionID", conn)
            deleteAnswersCmd.Parameters.AddWithValue("@QuestionID", questionID)
            deleteAnswersCmd.ExecuteNonQuery()

            Dim deleteQuestionCmd As New SqlCommand("DELETE FROM Questions WHERE
QuestionID = @QuestionID", conn)
            deleteQuestionCmd.Parameters.AddWithValue("@QuestionID", questionID)
            deleteQuestionCmd.ExecuteNonQuery()

```

```

        ' Reorder remaining questions
        ReorderAfterDeletion(conn, questionID)

        ' Refresh the grid
        LoadQuestions()

        ClientScript.RegisterStartupScript(Me.GetType(), "alert",
"alert('Question deleted successfully.');" , True)
        Catch ex As Exception
            ClientScript.RegisterStartupScript(Me.GetType(), "alert",
"alert('Error adding question: " & ex.Message.Replace("'", "\"") & "');" , True)
        End Try
    End Using
End Sub

Private Sub ReorderAfterDeletion(conn As SqlConnection, deletedQuestionID As Integer)
    ' Get the display order of the deleted question
    Dim getOrderCmd As New SqlCommand("SELECT DisplayOrder FROM Questions WHERE
QuestionID = @QuestionID", conn)
    getOrderCmd.Parameters.AddWithValue("@QuestionID", deletedQuestionID)
    Dim deletedOrder As Object = getOrderCmd.ExecuteScalar()

    If deletedOrder IsNot Nothing Then
        ' Update display orders of questions that came after the deleted one
        Dim updateCmd As New SqlCommand("UPDATE Questions SET DisplayOrder =
DisplayOrder - 1 " &
"WHERE QuizID = @QuizID AND DisplayOrder >
@DeletedOrder", conn)
        updateCmd.Parameters.AddWithValue("@QuizID", quizID)
        updateCmd.Parameters.AddWithValue("@DeletedOrder", deletedOrder)
        updateCmd.ExecuteNonQuery()
    End If
End Sub

Private Sub ReorderQuestion(questionID As Integer, moveUp As Boolean)
    Using conn As New
SqlConnection(ConfigurationManager.ConnectionStrings("QuizDB").ConnectionString)
        conn.Open()

        ' Get current question's display order
        Dim currentOrder As Integer
        Dim getOrderCmd As New SqlCommand("SELECT DisplayOrder FROM Questions
WHERE QuestionID = @QuestionID", conn)
        getOrderCmd.Parameters.AddWithValue("@QuestionID", questionID)
        currentOrder = Convert.ToInt32(getOrderCmd.ExecuteScalar())

        ' Determine swap direction
        Dim swapOrder As Integer = If(moveUp, currentOrder - 1, currentOrder +
1)

        ' Find question to swap with
        Dim swapCmd As New SqlCommand("SELECT QuestionID FROM Questions WHERE
QuizID = @QuizID AND DisplayOrder = @DisplayOrder", conn)
        swapCmd.Parameters.AddWithValue("@QuizID", quizID)
        swapCmd.Parameters.AddWithValue("@DisplayOrder", swapOrder)
        Dim swapID As Object = swapCmd.ExecuteScalar()

```

```

        If swapID IsNot Nothing Then
            ' Perform the swap
            Dim updateCmd As New SqlCommand("UPDATE Questions SET DisplayOrder =
@NewOrder WHERE QuestionID = @QuestionID", conn)

            ' Update current question
            updateCmd.Parameters.AddWithValue("@QuestionID", questionID)
            updateCmd.Parameters.AddWithValue("@NewOrder", swapOrder)
            updateCmd.ExecuteNonQuery()

            ' Update swapped question
            updateCmd.Parameters.Clear()
            updateCmd.Parameters.AddWithValue("@QuestionID", swapID)
            updateCmd.Parameters.AddWithValue("@NewOrder", currentOrder)
            updateCmd.ExecuteNonQuery()
        End If
    End Using
End Sub
Private Sub ClearQuestionForm()
    txtNewQuestion.Text = ""
    txtOption1.Text = ""
    txtOption2.Text = ""
    txtOption3.Text = ""
    txtOption4.Text = ""
    ddlCorrectAnswer.SelectedIndex = 0
End Sub
End Class

```