

**Name: Shahzaib**

**Project Name: Teacher Attendance Application**

### **Project Details:**

1. **Security:**

- All pages check for authentication and proper role
- Password hashing is used throughout
- SQL parameters prevent injection

2. **Database Connection:**

- Uses your Somee.com connection string from Web.config
- All connections are properly closed/disposed

3. **Teacher Features:**

- Dashboard with statistics
- Class management
- Attendance recording
- Attendance reporting
- Profile management

4. **Student Features:**

- View class schedule
- View attendance records
- View attendance statistics
- Profile management

5. **Shared Components:**

- DatabaseHelper for connection management
- UserManager for authentication
- TeacherManager for teacher-specific operations

### **Files Hierarchy:**

SchoolManagementSystem/

|

| └─ App\_Code/

| | └─ DatabaseHelper.vb (Database connection and operations)

| | └─ UserManager.vb (User authentication and role management)

| | └─ TeacherManager.vb (Teacher-specific operations)

| | └─ StudentManager.vb (Student-specific operations)

|

| -----└─ Default.aspx (Landing page)

| -----└─ Login.aspx (Login page)

| -----└─ Register.aspx (Registration page)

|

| └─ Student/

| | └─ Dashboard.aspx (Student main page)

| | └─ MyClasses.aspx

| | └─ Attendance.aspx

|

| |

| └─ Teacher/

| | └─ Dashboard.aspx (Teacher main page)

| | └─ MyClasses.aspx

| | └─ TakeAttendance.aspx

| | └─ ViewStudents.aspx

| └─ Web.config (Configuration file)

## Default.aspx.vb:

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

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If User.Identity.IsAuthenticated Then
            lblMessage.Text = "Welcome back, " & User.Identity.Name & "!"
            lnkLogin.Visible = False
            lnkRegister.Visible = False
        End If
    End Sub
End Class
```

## Login.aspx.vb:

```
Imports System.Data.SqlClient
Imports System.Web.Security

Partial Class Login
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Request.QueryString("registered") = "true" Then
            lblError.Text = "Registration successful! Please login."
            lblError.ForeColor = Drawing.Color.Green
            lblError.Visible = True
        End If
    End Sub

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

        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim query As String = "SELECT user_id, username, password_hash, role FROM
Users WHERE (username = @username OR email = @username) AND status = 'active'"

        Try
            conn.Open()
            Dim cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@username", username)

            Dim reader As SqlDataReader = cmd.ExecuteReader()
            If reader.Read() Then
                Dim storedHash As String = reader("password_hash").ToString()
                If UserManager.VerifyPassword(password, storedHash) Then
                    ' Create authentication ticket

FormsAuthentication.SetAuthCookie(reader("username").ToString(),
chkRemember.Checked)

                    ' Store user info in session
                    Session("UserID") = reader("user_id")
                    Session("Role") = reader("role")
                End If
            End If
        Catch ex As Exception
            ' Handle exception
        End Try
    End Sub
End Class
```

```

        ' Redirect based on role
        Dim redirectUrl As String = If(reader("role") = "student",
"Student/Dashboard.aspx", "Teacher/Dashboard.aspx")
        Response.Redirect(redirectUrl)
    Else
        ShowError("Invalid password")
    End If
Else
    ShowError("User not found or inactive.")
End If

Catch ex As Exception
    ShowError("An error occurred: " & ex.Message)
Finally
    conn.Close()
End Try
End Sub

Private Sub ShowError(message As String)
    lblError.Text = message
    lblError.ForeColor = Drawing.Color.Red
    lblError.Visible = True
End Sub

End Class

```

## Register.aspx.vb

```

Imports System.Data.SqlClient
Imports System.Web.Security

Partial Class Register
    Inherits System.Web.UI.Page

    Protected Sub ddlRole_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ddlRole.SelectedIndexChanged
        studentFields.Visible = (ddlRole.SelectedValue = "student")
        teacherFields.Visible = (ddlRole.SelectedValue = "teacher")
    End Sub

    Protected Sub btnRegister_Click(sender As Object, e As EventArgs) Handles
btnRegister.Click
        ' Validate passwords match
        If txtPassword.Text <> txtConfirmPassword.Text Then
            ShowError("Passwords do not match")
            Return
        End If

        ' Validate role selected
        If String.IsNullOrEmpty(ddlRole.SelectedValue) Then
            ShowError("Please select a role")
            Return
        End If

        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim transaction As SqlTransaction = Nothing
        Dim userId As Integer = 0

        Try

```

```

conn.Open()
transaction = conn.BeginTransaction()

' Check if username or email already exists
Dim checkSql As String = "SELECT COUNT(*) FROM Users WHERE username =
@username OR email = @email"
Dim checkCmd As New SqlCommand(checkSql, conn, transaction)
checkCmd.Parameters.AddWithValue("@username", txtUsername.Text)
checkCmd.Parameters.AddWithValue("@email", txtEmail.Text)

Dim exists As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())
If exists > 0 Then
    ShowError("Username or email already exists")
    Return
End If

' Insert into Users table
Dim userSql As String = "INSERT INTO Users (username, email,
password_hash, role, status) " &
"VALUES (@username, @email, @password, @role, 'active');"
" &
"SELECT SCOPE_IDENTITY();"

Dim userCmd As New SqlCommand(userSql, conn, transaction)
userCmd.Parameters.AddWithValue("@username", txtUsername.Text)
userCmd.Parameters.AddWithValue("@email", txtEmail.Text)
userCmd.Parameters.AddWithValue("@password",
UserManager.HashPassword(txtPassword.Text))
userCmd.Parameters.AddWithValue("@role", ddlRole.SelectedValue)

userId = Convert.ToInt32(userCmd.ExecuteScalar())

' Insert into role-specific table
' Insert into role-specific table
If ddlRole.SelectedValue = "student" Then
    Dim studentSql As String = "INSERT INTO Students (user_id,
full_name, roll_number, date_of_birth) " &
"VALUES (@user_id, @full_name, @roll_number, @dob)"

    Dim studentCmd As New SqlCommand(studentSql, conn, transaction)
    studentCmd.Parameters.AddWithValue("@user_id", userId)
    studentCmd.Parameters.AddWithValue("@full_name",
txtFullName.Text)
    studentCmd.Parameters.AddWithValue("@roll_number",
txtRollNumber.Text)
    studentCmd.Parameters.AddWithValue("@dob",
If(String.IsNullOrEmpty(txtDOB.Text), DBNull.Value,
Convert.ToDateTime(txtDOB.Text)))
    studentCmd.ExecuteNonQuery()
Else
    Dim teacherSql As String = "INSERT INTO Teachers (user_id,
full_name, department, phone) " &
"VALUES (@user_id, @full_name, @department, @phone)"

    Dim teacherCmd As New SqlCommand(teacherSql, conn, transaction)
    teacherCmd.Parameters.AddWithValue("@user_id", userId)
    teacherCmd.Parameters.AddWithValue("@full_name",
txtFullName.Text)
    teacherCmd.Parameters.AddWithValue("@department",
txtDepartment.Text)
    teacherCmd.Parameters.AddWithValue("@phone",
If(String.IsNullOrEmpty(txtPhone.Text), DBNull.Value, txtPhone.Text))

```

```

        teacherCmd.ExecuteNonQuery()
    End If

    transaction.Commit()
    lblSuccess.Text = "Registration successful! Please login."
    lblSuccess.Visible = True
    lblError.Visible = False

    ' Clear form
    txtUsername.Text = ""
    txtEmail.Text = ""
    txtFullName.Text = ""
    If ddlRole.SelectedValue = "student" Then
        txtRollNumber.Text = ""
        txtDOB.Text = ""
    Else
        txtDepartment.Text = ""
        txtPhone.Text = ""
    End If
End If

Catch ex As Exception
    If transaction IsNot Nothing Then
        transaction.Rollback()
    End If
    ShowError("Registration failed: " & ex.Message)
Finally
    conn.Close()
End Try
End Sub

Private Sub ShowError(message As String)
    lblError.Text = message
    lblError.Visible = True
    lblSuccess.Visible = False
End Sub
End Class

```

## Student:

### Dashboard.aspx.vb:

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class Student_Dashboard
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing Then
                Response.Redirect("~/Login.aspx")
            End If

            LoadStudentData()
            LoadDashboardStats()
            LoadTodayClasses()
        End If
    End Sub
End Class

```

```

End Sub

Private Sub LoadStudentData()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()
    Dim query As String = "SELECT s.full_name FROM Students s INNER JOIN
Users u ON s.user_id = u.user_id WHERE s.user_id = @user_id"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

        Dim reader As SqlDataReader = cmd.ExecuteReader()
        If reader.Read() Then
            lblStudentName.Text = reader("full_name").ToString()
        End If
        reader.Close()
    Catch ex As Exception
        ' Handle error
    Finally
        conn.Close()
    End Try
End Sub

Private Sub LoadDashboardStats()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()

    ' Get total classes
    Dim classQuery As String = "SELECT COUNT(*) FROM StudentClasses WHERE
student_id = (SELECT student_id FROM Students WHERE user_id = @user_id)"

    ' Get attendance percentage (simplified)
    Dim attendanceQuery As String = "SELECT
(COUNT(CASE WHEN status = 'Present' THEN
1 END) * 100.0 /
NULLIF(COUNT(*), 0)) AS
attendance_percent
FROM Attendance
WHERE student_id = (SELECT student_id
FROM Students WHERE user_id = @user_id)"

    ' Get upcoming classes (next 7 days)
    Dim upcomingQuery As String = "SELECT COUNT(*) FROM Classes c
INNER JOIN StudentClasses sc ON c.class_id =
sc.class_id
WHERE sc.student_id = (SELECT student_id
FROM Students WHERE user_id = @user_id)
AND c.schedule_day IN (SELECT
DATENAME(WEEKDAY, DATEADD(DAY, number, GETDATE())))
FROM master..spt_values
WHERE type = 'P' AND
number BETWEEN 0 AND 6)"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(classQuery, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))
        lblTotalClasses.Text = cmd.ExecuteScalar().ToString()

        cmd.CommandText = attendanceQuery
        Dim percentObj = cmd.ExecuteScalar()
        If percentObj IsNot DBNull.Value Then

```

```

        lblAttendancePercent.Text =
Math.Round(Convert.ToDouble(percentObj), 1) & "%"
    Else
        lblAttendancePercent.Text = "N/A"
    End If

    cmd.CommandText = upcomingQuery
    lblUpcomingClasses.Text = cmd.ExecuteScalar().ToString()
Catch ex As Exception
    ' Handle error
Finally
    conn.Close()
End Try
End Sub

Private Sub LoadTodayClasses()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()
    Dim query As String = "SELECT c.class_name, s.subject_name, c.start_time,
c.end_time, c.room_number
FROM Classes c
INNER JOIN Subjects s ON c.subject_id = s.subject_id
INNER JOIN StudentClasses sc ON c.class_id =
sc.class_id
WHERE sc.student_id = (SELECT student_id FROM
Students WHERE user_id = @user_id)
AND c.schedule_day = DATENAME(WEEKDAY, GETDATE())
ORDER BY c.start_time"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

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

        If dt.Rows.Count > 0 Then
            rptTodayClasses.DataSource = dt
            rptTodayClasses.DataBind()
        Else
            lblNoClassesToday.Visible = True
        End If
    Catch ex As Exception
        ' Handle error
    Finally
        conn.Close()
    End Try
End Sub

Protected Sub lnkLogout_Click(ByVal sender As Object, ByVal e As EventArgs)
    FormsAuthentication.SignOut()
    Session.Abandon()
    Response.Redirect("~/Login.aspx")
End Sub
End Class

```

## MyClasses.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

```



```

Partial Class Student_MyClasses
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing Then
                Response.Redirect("~/Login.aspx")
            End If
            LoadClasses()
        End If
    End Sub

    Private Sub LoadClasses()
        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim query As String = "SELECT c.class_name, s.subject_name,
c.schedule_day, c.start_time, c.end_time,
                                c.room_number, t.full_name AS teacher_name
FROM Classes c
INNER JOIN Subjects s ON c.subject_id = s.subject_id
INNER JOIN Teachers t ON c.teacher_id = t.teacher_id
INNER JOIN StudentClasses sc ON c.class_id =
sc.class_id
                                WHERE sc.student_id = (SELECT student_id FROM
Students WHERE user_id = @user_id)
                                ORDER BY
                                CASE c.schedule_day
                                    WHEN 'Monday' THEN 1
                                    WHEN 'Tuesday' THEN 2
                                    WHEN 'Wednesday' THEN 3
                                    WHEN 'Thursday' THEN 4
                                    WHEN 'Friday' THEN 5
                                    WHEN 'Saturday' THEN 6
                                    WHEN 'Sunday' THEN 7
                                END, c.start_time"

        Try
            conn.Open()
            Dim cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

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

            gvClasses.DataSource = dt
            gvClasses.DataBind()
        Catch ex As Exception
            ' Handle error
        Finally
            conn.Close()
        End Try
    End Sub

    Protected Sub lnkLogout_Click(ByVal sender As Object, ByVal e As EventArgs)
        FormsAuthentication.SignOut()
        Session.Abandon()
        Response.Redirect("~/Login.aspx")
    End Sub
End Class

```

## Attendance.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class Student_Attendance
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing Then
                Response.Redirect("~/Login.aspx")
            End If

            LoadClassDropdown()
            LoadAttendance()
            LoadAttendanceSummary()
        End If
    End Sub

    Private Sub LoadClassDropdown()
        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim query As String = "SELECT c.class_id, c.class_name
                                FROM Classes c
                                INNER JOIN StudentClasses sc ON c.class_id =
sc.class_id
                                WHERE sc.student_id = (SELECT student_id FROM
Students WHERE user_id = @user_id)
                                ORDER BY c.class_name"

        Try
            conn.Open()
            Dim cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

            Dim reader As SqlDataReader = cmd.ExecuteReader()
            ddlClasses.DataSource = reader
            ddlClasses.DataTextField = "class_name"
            ddlClasses.DataValueField = "class_id"
            ddlClasses.DataBind()
            reader.Close()
        Catch ex As Exception
            ' Handle error
        Finally
            conn.Close()
        End Try
    End Sub

    Private Sub LoadAttendance()
        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim query As String = "SELECT c.class_name, a.date, a.status, a.notes
                                FROM Attendance a
                                INNER JOIN Classes c ON a.class_id = c.class_id
                                WHERE a.student_id = (SELECT student_id FROM
Students WHERE user_id = @user_id)"

        ' Apply filters
        If ddlClasses.SelectedValue <> "0" Then
            query += " AND a.class_id = @class_id"
        End If

        If ddlMonths.SelectedValue <> "0" Then
            query += " AND MONTH(a.date) = @month"
        End If
    End Sub

```

```

query += " ORDER BY a.date DESC"

Try
    conn.Open()
    Dim cmd As New SqlCommand(query, conn)
    cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

    If ddlClasses.SelectedValue <> "0" Then
        cmd.Parameters.AddWithValue("@class_id",
ddlClasses.SelectedValue)
    End If

    If ddlMonths.SelectedValue <> "0" Then
        cmd.Parameters.AddWithValue("@month", ddlMonths.SelectedValue)
    End If

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

    gvAttendance.DataSource = dt
    gvAttendance.DataBind()
Catch ex As Exception
    ' Handle error
Finally
    conn.Close()
End Try
End Sub

Private Sub LoadAttendanceSummary()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()
    Dim query As String = "SELECT
                                COUNT(*) AS total,
                                COUNT(CASE WHEN status = 'Present' THEN 1 END) AS
present,
                                COUNT(CASE WHEN status = 'Absent' THEN 1 END) AS
absent,
                                COUNT(CASE WHEN status = 'Late' THEN 1 END) AS late,
                                COUNT(CASE WHEN status = 'Excused' THEN 1 END) AS
excused
                                FROM Attendance
                                WHERE student_id = (SELECT student_id FROM Students
WHERE user_id = @user_id)"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

        Dim reader As SqlDataReader = cmd.ExecuteReader()
        If reader.Read() Then
            Dim total As Integer = Convert.ToInt32(reader("total"))
            Dim present As Integer = Convert.ToInt32(reader("present"))
            Dim absent As Integer = Convert.ToInt32(reader("absent"))
            Dim late As Integer = Convert.ToInt32(reader("late"))
            Dim excused As Integer = Convert.ToInt32(reader("excused"))

            Dim percent As Double = If(total > 0, Math.Round((present /
total) * 100, 1), 0)

            lblSummary.Text = String.Format("Total Records: {0} | Present:
{1} ({2}%) | Absent: {3} | Late: {4} | Excused: {5}",

```

```

late, excused)
    End If
    reader.Close()
Catch ex As Exception
    ' Handle error
Finally
    conn.Close()
End Try
End Sub

Protected Sub ddlClasses_SelectedIndexChanged(sender As Object, e As
EventArgs)
    LoadAttendance()
    LoadAttendanceSummary()
End Sub

Protected Sub ddlMonths_SelectedIndexChanged(sender As Object, e As
EventArgs)
    LoadAttendance()
    LoadAttendanceSummary()
End Sub

Protected Sub lnkLogout_Click(ByVal sender As Object, ByVal e As EventArgs)
    FormsAuthentication.SignOut()
    Session.Abandon()
    Response.Redirect("~/Login.aspx")
End Sub
End Class

```

**Teacher:**

## Dashboard.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class Teacher_Dashboard
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing Then
                Response.Redirect("~/Login.aspx")
            End If

            LoadTeacherData()
            LoadDashboardStats()
            LoadTodaysClasses()
        End If
    End Sub

    Private Sub LoadTeacherData()
        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim query As String = "SELECT full_name FROM Teachers WHERE user_id =
@user_id"

        Try

```

```

        conn.Open()
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

        Dim reader As SqlDataReader = cmd.ExecuteReader()
        If reader.Read() Then
            lblTeacherName.Text = reader("full_name").ToString()
        End If
        reader.Close()
    Catch ex As Exception
        ' Handle error
    Finally
        conn.Close()
    End Try
End Sub

Private Sub LoadDashboardStats()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()

    ' Get total classes
    Dim classQuery As String = "SELECT COUNT(*) FROM Classes WHERE teacher_id
= (SELECT teacher_id FROM Teachers WHERE user_id = @user_id)"

    ' Get total students
    Dim studentQuery As String = "SELECT COUNT(DISTINCT sc.student_id)
                                FROM StudentClasses sc
                                INNER JOIN Classes c ON sc.class_id =
c.class_id
                                WHERE c.teacher_id = (SELECT teacher_id FROM
Teachers WHERE user_id = @user_id)"

    ' Get today's classes count
    Dim todayQuery As String = "SELECT COUNT(*) FROM Classes
                                WHERE teacher_id = (SELECT teacher_id FROM
Teachers WHERE user_id = @user_id)
                                AND schedule_day = DATENAME(WEEKDAY,
GETDATE())"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(classQuery, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))
        lblTotalClasses.Text = cmd.ExecuteScalar().ToString()

        cmd.CommandText = studentQuery
        lblTotalStudents.Text = cmd.ExecuteScalar().ToString()

        cmd.CommandText = todayQuery
        lblTodaysClasses.Text = cmd.ExecuteScalar().ToString()
    Catch ex As Exception
        ' Handle error
    Finally
        conn.Close()
    End Try
End Sub

Private Sub LoadTodaysClasses()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()
    Dim query As String = "SELECT c.class_id, c.class_name, s.subject_name,
c.start_time, c.end_time, c.room_number,
                        (SELECT COUNT(*) FROM StudentClasses WHERE class_id
= c.class_id) AS student_count
                        FROM Classes c"

```

```

                INNER JOIN Subjects s ON c.subject_id = s.subject_id
                WHERE c.teacher_id = (SELECT teacher_id FROM
Teachers WHERE user_id = @user_id)
                AND c.schedule_day = DATENAME(WEEKDAY, GETDATE())
                ORDER BY c.start_time"

Try
    conn.Open()
    Dim cmd As New SqlCommand(query, conn)
    cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

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

    If dt.Rows.Count > 0 Then
        rptTodaysClasses.DataSource = dt
        rptTodaysClasses.DataBind()
    Else
        lblNoClassesToday.Visible = True
    End If
Catch ex As Exception
    ' Handle error
Finally
    conn.Close()
End Try
End Sub

Protected Sub lnkLogout_Click(ByVal sender As Object, ByVal e As EventArgs)
    FormsAuthentication.SignOut()
    Session.Abandon()
    Response.Redirect("~/Login.aspx")
End Sub
End Class

```

## MyClasses.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class Teacher_MyClasses
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing Then
                Response.Redirect("~/Login.aspx")
            End If
            LoadClasses()
        End If
    End Sub

    Private Sub LoadClasses()
        Dim conn As SqlConnection = DatabaseHelper.GetConnection()
        Dim query As String = "SELECT c.class_id, c.class_name, s.subject_name,
c.schedule_day, c.start_time, c.end_time,
c.room_number, (SELECT COUNT(*) FROM StudentClasses
WHERE class_id = c.class_id) AS student_count
FROM Classes c
INNER JOIN Subjects s ON c.subject_id = s.subject_id"

```

```

                                WHERE c.teacher_id = (SELECT teacher_id FROM
Teachers WHERE user_id = @user_id)
                                ORDER BY
                                CASE c.schedule_day
                                    WHEN 'Monday' THEN 1
                                    WHEN 'Tuesday' THEN 2
                                    WHEN 'Wednesday' THEN 3
                                    WHEN 'Thursday' THEN 4
                                    WHEN 'Friday' THEN 5
                                    WHEN 'Saturday' THEN 6
                                    WHEN 'Sunday' THEN 7
                                END, c.start_time"

```

```

Try
    conn.Open()
    Dim cmd As New SqlCommand(query, conn)
    cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

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

    gvClasses.DataSource = dt
    gvClasses.DataBind()
Catch ex As Exception
    ' Handle error
Finally
    conn.Close()
End Try
End Sub

Protected Sub lnkLogout_Click(ByVal sender As Object, ByVal e As EventArgs)
    FormsAuthentication.SignOut()
    Session.Abandon()
    Response.Redirect("~/Login.aspx")
End Sub
End Class

```

## ViewAttendance.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient
Imports System.IO
Imports System.Web.UI

Partial Class Teacher_ViewAttendance
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing Then
                Response.Redirect("~/Login.aspx")
            End If

            LoadClassDropdown()
            txtDateFrom.Text = DateTime.Today.AddDays(-7).ToString("yyyy-MM-dd")
            txtDateTo.Text = DateTime.Today.ToString("yyyy-MM-dd")
            LoadAttendance()
        End If
    End Sub
End Class

```

```

Private Sub LoadClassDropdown()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()
    Dim query As String = "SELECT c.class_id, c.class_name
                           FROM Classes c
                           WHERE c.teacher_id = (SELECT teacher_id FROM
Teachers WHERE user_id = @user_id)
                           ORDER BY c.class_name"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

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

        ddlClasses.DataSource = dt
        ddlClasses.DataTextField = "class_name"
        ddlClasses.DataValueField = "class_id"
        ddlClasses.DataBind()
    Catch ex As Exception
        ' Handle error
    Finally
        conn.Close()
    End Try
End Sub

Private Sub LoadAttendance()
    Dim conn As SqlConnection = DatabaseHelper.GetConnection()
    Dim query As String = "SELECT c.class_name, s.full_name, s.roll_number,
a.date, a.status, a.notes
                           FROM Attendance a
                           INNER JOIN Classes c ON a.class_id = c.class_id
                           INNER JOIN Students s ON a.student_id = s.student_id
                           WHERE c.teacher_id = (SELECT teacher_id FROM
Teachers WHERE user_id = @user_id)"

    ' Apply filters
    If ddlClasses.SelectedValue <> "0" Then
        query += " AND a.class_id = @class_id"
    End If

    If Not String.IsNullOrEmpty(txtDateFrom.Text) Then
        query += " AND a.date >= @date_from"
    End If

    If Not String.IsNullOrEmpty(txtDateTo.Text) Then
        query += " AND a.date <= @date_to"
    End If

    query += " ORDER BY a.date DESC, c.class_name, s.full_name"

    Try
        conn.Open()
        Dim cmd As New SqlCommand(query, conn)
        cmd.Parameters.AddWithValue("@user_id", Session("UserID"))

        If ddlClasses.SelectedValue <> "0" Then
            cmd.Parameters.AddWithValue("@class_id",
ddlClasses.SelectedValue)
        End If
    End Try

```



```

        If Not String.IsNullOrEmpty(txtDateFrom.Text) Then
            cmd.Parameters.AddWithValue("@date_from",
DateTime.Parse(txtDateFrom.Text))
        End If

        If Not String.IsNullOrEmpty(txtDateTo.Text) Then
            cmd.Parameters.AddWithValue("@date_to",
DateTime.Parse(txtDateTo.Text))
        End If

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

        gvAttendance.DataSource = dt
        gvAttendance.DataBind()
    Catch ex As Exception
        ' Handle error
    Finally
        conn.Close()
    End Try
End Sub

Protected Sub ddlClasses_SelectedIndexChanged(ByVal sender As Object, ByVal e
As EventArgs)
    LoadAttendance()
End Sub

Protected Sub btnFilter_Click(ByVal sender As Object, ByVal e As EventArgs)
    LoadAttendance()
End Sub

Protected Sub btnExport_Click(ByVal sender As Object, ByVal e As EventArgs)
    Response.Clear()
    Response.Buffer = True
    Response.AddHeader("content-disposition",
"attachment;filename=AttendanceReport_" & DateTime.Now.ToString("yyyyMMdd") &
".xls")
    Response.Charset = ""
    Response.ContentType = "application/vnd.ms-excel"

    Dim sw As New StringWriter()
    Dim hw As New HtmlTextWriter(sw)

    ' Create a form to contain the grid
    Dim form As New HtmlForm()
    gvAttendance.Parent.Controls.Add(form)
    form.Controls.Add(gvAttendance)

    ' Render the grid to the HtmlTextWriter
    form.RenderControl(hw)

    ' Style to format columns
    Dim style As String = "<style> .textmode { mso-number-format:\@; }
</style>"
    Response.Write(style)
    Response.Output.Write(sw.ToString())
    Response.Flush()
    Response.End()
End Sub

Protected Sub lnkLogout_Click(ByVal sender As Object, ByVal e As EventArgs)
    FormsAuthentication.SignOut()

```

```
        Session.Abandon()  
        Response.Redirect("~/Login.aspx")  
    End Sub  
End Class
```