**1.Authentication page code**

Imports System.Data.SqlClient

Imports System.Web.Configuration

Partial Class authentication

Inherits System.Web.UI.Page

Protected Sub btnLogin\_Click(sender As Object, e As EventArgs) Handles btnLogin.Click

Dim username As String = txtUsername.Text.Trim()

Dim password As String = txtPassword.Text.Trim()

Dim connString As String = WebConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Dim query As String = "SELECT Role FROM USER4 WHERE Username=@Username AND Password=@Password"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Username", username)

cmd.Parameters.AddWithValue("@Password", password)

Try

conn.Open()

Dim reader As SqlDataReader = cmd.ExecuteReader()

If reader.HasRows Then

ddlRoles.Items.Clear()

While reader.Read()

ddlRoles.Items.Add(reader("Role").ToString())

End While

ddlRoles.Visible = True

btnSelectRole.Visible = True

Session("Username") = username

Session("Password") = password

lblError.Text = "Select your role to continue."

Else

lblError.Text = "Invalid username or password."

End If

Catch ex As Exception

lblError.Text = "Error: " & ex.Message

End Try

End Using

End Using

End Sub

Protected Sub btnSelectRole\_Click(sender As Object, e As EventArgs) Handles btnSelectRole.Click

Dim selectedRole As String = ddlRoles.SelectedValue

Dim username As String = Session("Username").ToString()

Dim password As String = Session("Password").ToString()

Dim connString As String = WebConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Dim query As String = "SELECT UserID FROM USER4 WHERE Username=@Username AND Password=@Password AND Role=@Role"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Username", username)

cmd.Parameters.AddWithValue("@Password", password)

cmd.Parameters.AddWithValue("@Role", selectedRole)

Try

conn.Open()

Dim userID = cmd.ExecuteScalar()

If userID IsNot Nothing Then

Session("UserID") = userID.ToString()

Session("Role") = selectedRole.ToLower()

Select Case selectedRole.ToLower()

Case "admin"

Response.Redirect("Admin.aspx")

Case "student"

Response.Redirect("home.aspx")

Case "tutor"

Response.Redirect("updateattendance.aspx")

Case Else

lblError.Text = "Unknown role selected."

End Select

Else

lblError.Text = "User with selected role not found."

End If

Catch ex As Exception

lblError.Text = "Error: " & ex.Message

End Try

End Using

End Using

End Sub

End Class

**Code explanation:**

This page probably handles **user authentication checks**, such as verifying session tokens, checking user roles, and redirecting unauthorized users. It's commonly used for access control between public and restricted areas of the application.

**2.Admin page code**

Imports System.Data.SqlClient

Imports System.Web.Configuration

Partial Class AdminDashboard

Inherits System.Web.UI.Page

Dim connString As String = WebConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

If Not IsPostBack Then

BindStudents()

BindTutors()

End If

End Sub

Private Sub BindStudents()

Dim query As String = "SELECT UserID, Username, Role FROM USER4 WHERE Role='student'"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

conn.Open()

gvStudents.DataSource = cmd.ExecuteReader()

gvStudents.DataBind()

End Using

End Using

End Sub

Private Sub BindTutors()

Dim query As String = "SELECT UserID, Username, Role FROM USER4 WHERE Role='tutor'"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

conn.Open()

gvTutors.DataSource = cmd.ExecuteReader()

gvTutors.DataBind()

End Using

End Using

End Sub

Protected Sub gvStudents\_RowDeleting(sender As Object, e As GridViewDeleteEventArgs)

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

DeleteUser(userID)

lblMessage.Text = "Student deleted successfully."

BindStudents()

End Sub

Protected Sub gvTutors\_RowDeleting(sender As Object, e As GridViewDeleteEventArgs)

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

DeleteUser(userID)

lblMessage.Text = "Tutor deleted successfully."

BindTutors()

End Sub

Private Sub DeleteUser(userID As Integer)

Dim query As String = "DELETE FROM USER4 WHERE UserID=@UserID"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@UserID", userID)

conn.Open()

cmd.ExecuteNonQuery()

End Using

End Using

End Sub

End Class

**Code explanation:**

This page likely serves as the **administrator dashboard**. It is responsible for managing system-wide tasks like viewing registered users (students or tutors), managing quizzes, deleting records, and overseeing the entire system’s backend operations such as database monitoring and platform control.

**3.check attendance page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Data

Partial Class StudentAttendance

Inherits System.Web.UI.Page

Private connString As String = ConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Protected Sub btnViewAttendance\_Click(sender As Object, e As EventArgs) Handles btnViewAttendance.Click

Dim studentID As Integer

If Not Integer.TryParse(txtStudentID.Text.Trim(), studentID) Then

ShowAlert("Please enter a valid student ID.")

Return

End If

Using conn As New SqlConnection(connString)

Dim query As String = "SELECT " &

"c.course\_name, " &

"a.session\_date, " &

"a.session\_time, " &

"a.duration\_minutes, " &

"a.topic\_covered, " &

"sa.attendance\_status, " &

"sa.notes " &

"FROM StudentAttendance sa " &

"INNER JOIN Attendance a ON sa.session\_id = a.session\_id " &

"INNER JOIN COURSE9 c ON a.course\_id = c.course\_id " &

"WHERE sa.student\_id = @StudentID " &

"ORDER BY a.session\_date DESC, a.session\_time DESC"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@StudentID", studentID)

Dim adapter As New SqlDataAdapter(cmd)

Dim dt As New DataTable()

Try

conn.Open()

adapter.Fill(dt)

gvAttendance.DataSource = dt

gvAttendance.DataBind()

Catch ex As Exception

ShowAlert("Error loading attendance: " & ex.Message)

End Try

End Using

End Using

End Sub

Private Sub ShowAlert(msg As String)

Dim safeMsg As String = msg.Replace("'", "\'")

ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('" & safeMsg & "');", True)

End Sub

End Class

**Code explanation:**

This page is designed for **students or tutors to view attendance records**. Students may use it to check their session participation, and tutors may use it to track attendance records for each student in a given course or language session.

**4.Available courses page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Web.UI.WebControls

Imports System

Partial Class Courses

Inherits System.Web.UI.Page

Private ReadOnly connString As String = ConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load

If Not IsPostBack Then

LoadAvailableCourses()

End If

End Sub

Private Sub LoadAvailableCourses()

Try

Using conn As New SqlConnection(connString)

Dim query As String = "SELECT course\_id, course\_name, course\_description FROM COURSE9 ORDER BY course\_name"

Using cmd As New SqlCommand(query, conn)

conn.Open()

gvCourseDetails.DataSource = cmd.ExecuteReader()

gvCourseDetails.DataBind()

gvCourseDetails.Visible = True

End Using

End Using

Catch ex As Exception

lblError.Text = "Error loading courses: " & ex.Message

lblError.Visible = True

End Try

End Sub

Protected Sub btnSearch\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnSearch.Click

Try

Dim searchTerm As String = txtSearch.Text.Trim()

Using conn As New SqlConnection(connString)

Dim query As String = "SELECT course\_id,course\_name,course\_description FROM COURSE9 " &

"WHERE course\_name LIKE @SearchTerm OR course\_description LIKE @SearchTerm " &

"ORDER BY course\_name"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@SearchTerm", "%" & searchTerm & "%")

conn.Open()

Dim reader = cmd.ExecuteReader()

gvCourseDetails.DataSource = reader

gvCourseDetails.DataBind()

gvCourseDetails.Visible = True

If Not reader.HasRows Then

lblMessage.Text = "No courses found matching your search."

lblMessage.Visible = True

Else

lblMessage.Text = ""

lblMessage.Visible = False

End If

End Using

End Using

Catch ex As Exception

lblError.Text = "Search Error: " & ex.Message

lblError.Visible = True

End Try

End Sub

Protected Sub gvCourseDetails\_RowCommand(ByVal sender As Object, ByVal e As GridViewCommandEventArgs)

If e.CommandName = "Enroll" Then

If Session("student\_id") IsNot Nothing Then

Try

Dim rowIndex As Integer = Convert.ToInt32(e.CommandArgument)

Dim courseId As Integer = Convert.ToInt32(gvCourseDetails.DataKeys(rowIndex).Value)

Dim studentId As Integer = Convert.ToInt32(Session("student\_id"))

Using conn As New SqlConnection(connString)

Dim checkQuery As String = "SELECT COUNT(\*) FROM ENROLLED\_COURSES WHERE student\_id = @S\_ID AND course\_id = @C\_ID"

Using checkCmd As New SqlCommand(checkQuery, conn)

checkCmd.Parameters.AddWithValue("@S\_ID", studentId)

checkCmd.Parameters.AddWithValue("@C\_ID", courseId)

conn.Open()

Dim existingCount As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())

If existingCount > 0 Then

lblMessage.Text = "You are already enrolled in this course."

lblMessage.Visible = True

Return

End If

End Using

Dim insertQuery As String = "INSERT INTO ENROLLED\_COURSES (student\_id,course\_id, Enroll\_Date) VALUES (@S\_ID, @C\_ID, GETDATE())"

Using cmd As New SqlCommand(insertQuery, conn)

cmd.Parameters.AddWithValue("@S\_ID", studentId)

cmd.Parameters.AddWithValue("@C\_ID", courseId)

cmd.ExecuteNonQuery()

lblMessage.Text = "Successfully enrolled in course!"

lblMessage.Visible = True

End Using

End Using

Catch ex As Exception

lblError.Text = "Error enrolling in course: " & ex.Message

lblError.Visible = True

End Try

Else

lblError.Text = "Please log in to enroll in a course."

lblError.Visible = True

End If

End If

End Sub

End Class

**Code explanation:**

This is the **courses listing page**, where students and tutors can view available courses. Students might be able to enroll in a course, and tutors might manage the courses they are assigned to teach.

**5.Home page code**

Imports System.Data.SqlClient

Partial Class Home

Inherits System.Web.UI.Page

Dim connString As String = "iaddatabase"

Dim currentUserEmail As String

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As EventArgs)

If Not IsPostBack Then

If Session("Email") IsNot Nothing Then

currentUserEmail = Session("Email").ToString()

LoadUserData(currentUserEmail)

Else

End If

End If

End Sub

Private Sub LoadUserData(ByVal email As String)

Dim userType As String = GetUserType(email)

If userType = "Student" Then

LoadStudentCourses(email)

LoadStudentSessions(email)

studentContent.Visible = True

tutorContent.Visible = False

ElseIf userType = "Tutor" Then

LoadTutorCourses(email)

LoadTutorSessions(email)

studentContent.Visible = False

tutorContent.Visible = True

End If

End Sub

Private Function GetUserType(ByVal email As String) As String

Dim query As String = "SELECT 'Student' AS UserType FROM STUDENT4 WHERE Email = @Email " &

"UNION " &

"SELECT 'Tutor' AS UserType FROM TUTORS WHERE Email = @Email"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Email", email)

conn.Open()

Dim result = cmd.ExecuteScalar()

If result IsNot Nothing Then

Return result.ToString()

End If

End Using

End Using

Return String.Empty

End Function

Private Sub LoadStudentCourses(ByVal email As String)

Dim query As String = "SELECT C.CourseName FROM COURSE9 C " &

"JOIN STUDENT\_COURSE\_LANGUAGE SC ON C.CourseID = SC.CourseID " &

"WHERE SC.StudentID IN (SELECT StudentID FROM STUDENT4 WHERE Email = @Email)"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Email", email)

conn.Open()

Dim reader As SqlDataReader = cmd.ExecuteReader()

While reader.Read()

Dim courseName As String = reader("CourseName").ToString()

studentCourses.InnerHtml &= "<div class='course-item'>" & courseName & "</div>"

End While

End Using

End Using

End Sub

Private Sub LoadStudentSessions(ByVal email As String)

Dim query As String = "SELECT S.SessionDate, S.SessionTime FROM SCHEDULE\_SESSIONS S " &

"JOIN STUDENT\_COURSE\_LANGUAGE SC ON S.CourseID = SC.CourseID " &

"WHERE SC.StudentID IN (SELECT StudentID FROM STUDENT4 WHERE Email = @Email) " &

"AND S.SessionDate > GETDATE()"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Email", email)

conn.Open()

Dim reader As SqlDataReader = cmd.ExecuteReader()

While reader.Read()

Dim sessionDate As String = reader("SessionDate").ToString()

Dim sessionTime As String = reader("SessionTime").ToString()

studentSessions.InnerHtml &= "<div class='session-item'>Session on " & sessionDate & " at " & sessionTime & "</div>"

End While

End Using

End Using

End Sub

Private Sub LoadTutorCourses(ByVal email As String)

Dim query As String = "SELECT C.CourseName FROM COURSE9 C " &

"JOIN TUTOR\_COURSE\_LANGUAGE TC ON C.CourseID = TC.CourseID " &

"WHERE TC.TutorID IN (SELECT TutorID FROM TUTORS WHERE Email = @Email)"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Email", email)

conn.Open()

Dim reader As SqlDataReader = cmd.ExecuteReader()

While reader.Read()

Dim courseName As String = reader("CourseName").ToString()

tutorCourses.InnerHtml &= "<div class='course-item'>" & courseName & "</div>"

End While

End Using

End Using

End Sub

Private Sub LoadTutorSessions(ByVal email As String)

Dim query As String = "SELECT S.SessionDate, S.SessionTime FROM SCHEDULE\_SESSIONS S " &

"JOIN TUTOR\_COURSE\_LANGUAGE TC ON S.CourseID = TC.CourseID " &

"WHERE TC.TutorID IN (SELECT TutorID FROM TUTORS WHERE Email = @Email) " &

"AND S.SessionDate > GETDATE()"

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Email", email)

conn.Open()

Dim reader As SqlDataReader = cmd.ExecuteReader()

While reader.Read()

Dim sessionDate As String = reader("SessionDate").ToString()

Dim sessionTime As String = reader("SessionTime").ToString()

tutorSessions.InnerHtml &= "<div class='session-item'>Session on " & sessionDate & " at " & sessionTime & "</div>"

End While

End Using

End Using

End Sub

End Class

**Code explanation:**

Once a user logs in, this is likely the **dashboard page**. It provides quick access to core features like enrolled courses, upcoming sessions, tutor contact info, and personal updates depending on the role (student/tutor).

**6.Login page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Security.Cryptography

Imports System.Text

Imports System

Partial Class Login

Inherits System.Web.UI.Page

Private ReadOnly connString As String = ConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Public Shared Function ComputeSHA256Hash(password As String) As String

Using sha256 As SHA256 = SHA256.Create()

Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)

Dim hash As Byte() = sha256.ComputeHash(bytes)

Return BitConverter.ToString(hash).Replace("-", "").ToLower()

End Using

End Function

Protected Sub btnLogin\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnLogin.Click

lblError.Visible = False

If String.IsNullOrWhiteSpace(txtEmail.Text) OrElse String.IsNullOrWhiteSpace(txtPassword.Text) Then

lblError.Text = "Please enter both email and password."

lblError.Visible = True

Return

End If

Dim email As String = txtEmail.Text.Trim()

Dim passwordHash As String = ComputeSHA256Hash(txtPassword.Text.Trim())

Try

Using conn As New SqlConnection(connString)

Dim query As String = "SELECT student\_id FROM STUDENT4 WHERE email = @Email AND Password = @Password"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@Email", email)

cmd.Parameters.AddWithValue("@Password", passwordHash)

conn.Open()

Dim userId As Object = cmd.ExecuteScalar()

If userId IsNot Nothing Then

Session("student\_id") = userId.ToString()

Response.Redirect("Courses.aspx")

Else

lblError.Text = "Invalid email or password."

lblError.Visible = True

End If

End Using

End Using

Catch ex As Exception

lblError.Text = "Login error: " & ex.Message

lblError.Visible = True

End Try

End Sub

End Class

**Code explanation:**

This page contains the **login interface** for students, tutors, and possibly admins. It validates email and password credentials against the database, sets session variables, and redirects users based on their roles.

**7.Logout page code**

Imports System

Imports System.Web

Imports System.Web.Security

Partial Class Logout

Inherits System.Web.UI.Page

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load

If Not IsPostBack Then

Session.Clear()

Session.Abandon()

FormsAuthentication.SignOut()

Response.Cache.SetExpires(DateTime.UtcNow.AddMinutes(-1))

Response.Cache.SetCacheability(HttpCacheability.NoCache)

Response.Cache.SetNoStore()

Response.AppendHeader("pragma", "no-cache")

Response.AppendHeader("Cache-Control", "no-store, no-cache, must-revalidate, post-check=0, pre-check=0")

Dim cookie As New HttpCookie("ASP.NET\_SessionId", "")

cookie.Expires = DateTime.Now.AddYears(-1)

Response.Cookies.Add(cookie)

End If

End Sub

End Class

**Code explanation:**

This page handles **user logout functionality**. It ends the session, clears cookies or tokens, and redirects the user to the login or default page, ensuring secure exit from the application.

**9.mycourses page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Data

Imports System

Partial Class MyCourses

Inherits System.Web.UI.Page

Private ReadOnly connString As String = ConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load

If Not IsPostBack Then

If Session("student\_id") Is Nothing Then

Response.Redirect("Login.aspx")

Else

LoadEnrolledCourses()

End If

End If

End Sub

Private Sub LoadEnrolledCourses()

Try

Dim studentId As Integer = Convert.ToInt32(Session("student\_id"))

Using conn As New SqlConnection(connString)

Dim query As String = "SELECT c.course\_id, c.course\_name, c.course\_description, ec.Enroll\_Date " &

"FROM ENROLLED\_COURSES ec " &

"JOIN COURSE9 c ON ec.course\_id = c.course\_id " &

"WHERE ec.student\_id = @StudentID " &

"ORDER BY ec.Enroll\_Date DESC"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@StudentID", studentId)

conn.Open()

Dim adapter As New SqlDataAdapter(cmd)

Dim dt As New DataTable()

adapter.Fill(dt)

gvEnrolledCourses.DataSource = dt

gvEnrolledCourses.DataBind()

End Using

End Using

Catch ex As Exception

lblError.Text = "Error loading courses: " & ex.Message

lblError.Visible = True

End Try

End Sub

End Class

**Code explanation:**

This page shows **all courses a logged-in student is enrolled in** or a tutor is teaching. It provides access to course material, session schedules, and potentially options to drop a course or contact the tutor.

**10.Register page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Security.Cryptography

Imports System.Text

Imports System

Imports System.Linq

Partial Class Register

Inherits System.Web.UI.Page

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

Public Shared Function ComputeSHA256Hash(password As String) As String

Using sha256 As SHA256 = SHA256.Create()

Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)

Dim hash As Byte() = sha256.ComputeHash(bytes)

Return BitConverter.ToString(hash).Replace("-", "").ToLower()

End Using

End Function

Private Function IsStrongPassword(password As String) As Boolean

Dim hasMinimumLength As Boolean = password.Length >= 8

Dim hasUpperCase As Boolean = password.Any(AddressOf Char.IsUpper)

Dim hasLowerCase As Boolean = password.Any(AddressOf Char.IsLower)

Dim hasDigit As Boolean = password.Any(AddressOf Char.IsDigit)

Dim hasSpecialChar As Boolean = password.Any(Function(c) Not Char.IsLetterOrDigit(c))

Return hasMinimumLength AndAlso hasUpperCase AndAlso hasLowerCase AndAlso hasDigit AndAlso hasSpecialChar

End Function

Protected Sub btnRegister\_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btnRegister.Click

Try

If String.IsNullOrEmpty(txtFirstName.Text.Trim()) OrElse

String.IsNullOrEmpty(txtEmail.Text.Trim()) OrElse

String.IsNullOrEmpty(txtPassword.Text.Trim()) OrElse

String.IsNullOrEmpty(txtAddress.Text.Trim()) OrElse

String.IsNullOrEmpty(txtProfileDesc.Text.Trim()) Then

lblError.Text = "Please fill all required fields (\*)"

lblError.Visible = True

Return

End If

Dim studentName As String = txtFirstName.Text.Trim()

Dim email As String = txtEmail.Text.Trim()

Dim password As String = txtPassword.Text.Trim()

Dim address As String = txtAddress.Text.Trim()

Dim profileDesc As String = txtProfileDesc.Text.Trim()

If Not IsStrongPassword(password) Then

lblError.Text = "Password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, one digit, and one special character."

lblError.Visible = True

Return

End If

Dim passwordHash As String = ComputeSHA256Hash(password)

Using checkConn As New SqlConnection(connString)

Dim checkQuery As String = "SELECT COUNT(\*) FROM STUDENT4 WHERE email = @Email"

Using checkCmd As New SqlCommand(checkQuery, checkConn)

checkCmd.Parameters.AddWithValue("@Email", email)

checkConn.Open()

Dim emailExists As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())

If emailExists > 0 Then

lblError.Text = "Email already registered. Please use a different email."

lblError.Visible = True

Return

End If

End Using

End Using

Using conn As New SqlConnection(connString)

Dim query As String = "INSERT INTO STUDENT4 (student\_name, email, Password, address, profile\_description) " &

"VALUES (@StudentName, @Email, @Password, @Address, @ProfileDesc)"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@StudentName", studentName)

cmd.Parameters.AddWithValue("@Email", email)

cmd.Parameters.AddWithValue("@Password", passwordHash)

cmd.Parameters.AddWithValue("@Address", address)

cmd.Parameters.AddWithValue("@ProfileDesc", profileDesc)

conn.Open()

cmd.ExecuteNonQuery()

Response.Redirect("Login.aspx?registered=true")

End Using

End Using

Catch ex As SqlException When ex.Number = 2627

lblError.Text = "Email already registered. Please use a different email."

lblError.Visible = True

Catch ex As Exception

lblError.Text = "Registration error: " & ex.Message

lblError.Visible = True

End Try

End Sub

End Class

**Code explanation:**

This page allows **new student registration** by collecting basic details like name, email, password, and address. It includes password hashing, email uniqueness checks, and redirects to login upon successful signup.

**11.tutor details page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Data

Imports System

Partial Class Tutors

Inherits System.Web.UI.Page

Private ReadOnly connString As String = ConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Protected Sub Page\_Load(sender As Object, e As EventArgs) Handles Me.Load

If Not IsPostBack Then

LoadTutorList()

pnlTutorDetails.Visible = False

End If

End Sub

Private Sub LoadTutorList(Optional searchName As String = "")

Using conn As New SqlConnection(connString)

conn.Open()

Dim query As String = "SELECT tutor\_id, tutor\_name, bio, hourly\_rate, experience FROM TUTORS WHERE tutor\_name LIKE @SearchName"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@SearchName", "%" & searchName & "%")

Dim adapter As New SqlDataAdapter(cmd)

Dim dtTutors As New DataTable()

adapter.Fill(dtTutors)

gvTutorList.DataSource = dtTutors

gvTutorList.DataBind()

End Using

End Using

End Sub

Protected Sub LoadTutorDetails(tutorId As Integer)

Using conn As New SqlConnection(connString)

conn.Open()

Dim tutorQueryBasic As String = "SELECT tutor\_id, tutor\_name, email, bio, hourly\_rate, experience FROM TUTORS WHERE tutor\_id = @TutorID"

Using tutorCmdBasic As New SqlCommand(tutorQueryBasic, conn)

tutorCmdBasic.Parameters.AddWithValue("@TutorID", tutorId)

Using reader = tutorCmdBasic.ExecuteReader()

If reader.Read() Then

pnlTutorDetails.Visible = True

lblError.Visible = False

lblTutorIDDetail.Text = reader("tutor\_id").ToString()

lblNameDetail.Text = "Name: " & reader("tutor\_name").ToString()

lblEmailDetail.Text = "Email: " & reader("email").ToString()

lblBioDetail.Text = "Bio: " & reader("bio").ToString()

lblRateDetail.Text = "Hourly Rate: $" & reader("hourly\_rate").ToString()

lblExperienceDetail.Text = "Experience: " & reader("experience").ToString() & " years"

Else

ShowNoTutorFound()

Return

End If

End Using

End Using

Dim courseQuery As String = "SELECT DISTINCT C.course\_name AS CourseName, " &

"C.course\_description AS CourseDescription " &

"FROM COURSE9 C " &

"INNER JOIN TUTOR\_COURSE\_LANGUAGE TCL ON C.course\_id = TCL.Course\_ID " &

"INNER JOIN TUTORS T ON TCL.Tutor\_ID = T.tutor\_id " &

"WHERE T.tutor\_id = @TutorID"

Using courseCmd As New SqlCommand(courseQuery, conn)

courseCmd.Parameters.AddWithValue("@TutorID", tutorId)

Dim courseAdapter As New SqlDataAdapter(courseCmd)

Dim dtCourses As New DataTable()

courseAdapter.Fill(dtCourses)

gvCoursesDetail.DataSource = dtCourses

gvCoursesDetail.DataBind()

End Using

Dim langQuery As String = "SELECT DISTINCT L.language\_name AS LanguageName, " &

"L.language\_description AS Description " &

"FROM LANGUAGE1 L " &

"INNER JOIN TUTOR\_COURSE\_LANGUAGE TCL ON L.language\_id = TCL.Language\_ID " &

"INNER JOIN TUTORS T ON TCL.Tutor\_ID = T.tutor\_id " &

"WHERE T.tutor\_id = @TutorID"

Using langCmd As New SqlCommand(langQuery, conn)

langCmd.Parameters.AddWithValue("@TutorID", tutorId)

Dim langAdapter As New SqlDataAdapter(langCmd)

Dim dtLangs As New DataTable()

langAdapter.Fill(dtLangs)

gvLanguagesDetail.DataSource = dtLangs

gvLanguagesDetail.DataBind()

End Using

End Using

End Sub

Private Sub ShowNoTutorFound()

pnlTutorDetails.Visible = False

lblError.Text = "No tutor found with that name."

lblError.Visible = True

gvTutorList.DataSource = Nothing

gvTutorList.DataBind()

gvCoursesDetail.DataSource = Nothing

gvCoursesDetail.DataBind()

gvLanguagesDetail.DataSource = Nothing

gvLanguagesDetail.DataBind()

End Sub

Protected Sub txtSearchTutor\_TextChanged(sender As Object, e As EventArgs) Handles txtSearchTutor.TextChanged

LoadTutorList(txtSearchTutor.Text.Trim())

pnlTutorDetails.Visible = False

End Sub

Protected Sub gvTutorList\_RowCommand(sender As Object, e As System.Web.UI.WebControls.GridViewCommandEventArgs) Handles gvTutorList.RowCommand

If e.CommandName = "ViewDetails" Then

Dim tutorId As Integer = Convert.ToInt32(e.CommandArgument)

LoadTutorDetails(tutorId)

End If

End Sub

End Class

**Code explanation:**

This page shows **profile details of a selected tutor**, including their bio, hourly rate, experience, and available courses. Students may use it to decide whether to book sessions with that tutor.

**12.update attendance page code**

Imports System.Data.SqlClient

Imports System.Configuration

Imports System.Data

Partial Class Attendance

Inherits System.Web.UI.Page

Private connString As String = ConfigurationManager.ConnectionStrings("iaddatabase").ConnectionString

Protected Sub Page\_Load(sender As Object, e As EventArgs) Handles Me.Load

If Not IsPostBack Then

LoadDropdowns()

End If

End Sub

Private Sub LoadDropdowns()

LoadDropdown("SELECT tutor\_id, tutor\_name FROM TUTORS", ddlTutors, "tutor\_name", "tutor\_id", "-- Select Tutor --")

LoadDropdown("SELECT course\_id, course\_name FROM COURSE9", ddlCourses, "course\_name", "course\_id", "-- Select Course --")

End Sub

Private Sub LoadDropdown(query As String, dropdown As DropDownList, textField As String, valueField As String, defaultText As String)

Using conn As New SqlConnection(connString)

Using cmd As New SqlCommand(query, conn)

conn.Open()

dropdown.DataSource = cmd.ExecuteReader()

dropdown.DataTextField = textField

dropdown.DataValueField = valueField

dropdown.DataBind()

dropdown.Items.Insert(0, New ListItem(defaultText, "0"))

End Using

End Using

End Sub

Protected Sub btnCreateSession\_Click(sender As Object, e As EventArgs) Handles btnCreateSession.Click

If ddlTutors.SelectedValue = "0" Or ddlCourses.SelectedValue = "0" Then

ShowAlert("Select tutor and course.")

Return

End If

Dim sessionDate As Date, sessionTime As TimeSpan, duration As Integer

If Not Date.TryParse(txtSessionDate.Text, sessionDate) Or Not TimeSpan.TryParse(txtSessionTime.Text, sessionTime) Or Not Integer.TryParse(txtDuration.Text, duration) Then

ShowAlert("Invalid date/time/duration.")

Return

End If

Dim topic = If(String.IsNullOrEmpty(txtTopic.Text), DBNull.Value, txtTopic.Text)

Using conn As New SqlConnection(connString)

Dim query = "INSERT INTO Attendance (tutor\_id, course\_id, session\_date, session\_time, duration\_minutes, topic\_covered) " &

"VALUES (@TutorID, @CourseID, @Date, @Time, @Duration, @Topic); SELECT SCOPE\_IDENTITY();"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@TutorID", ddlTutors.SelectedValue)

cmd.Parameters.AddWithValue("@CourseID", ddlCourses.SelectedValue)

cmd.Parameters.AddWithValue("@Date", sessionDate)

cmd.Parameters.AddWithValue("@Time", sessionTime)

cmd.Parameters.AddWithValue("@Duration", duration)

cmd.Parameters.AddWithValue("@Topic", topic)

conn.Open()

Dim sessionID = Convert.ToInt32(cmd.ExecuteScalar())

ViewState("CurrentSessionID") = sessionID

End Using

End Using

LoadStudents()

End Sub

Private Sub LoadStudents()

Dim sessionID = Convert.ToInt32(ViewState("CurrentSessionID"))

Using conn As New SqlConnection(connString)

Dim query = "SELECT s.student\_id, s.student\_name FROM STUDENT4 s " &

"JOIN ENROLLED\_COURSES sc ON s.student\_id = sc.student\_id " &

"WHERE sc.course\_id = (SELECT course\_id FROM Attendance WHERE session\_id = @SessionID)"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@SessionID", sessionID)

Dim adapter As New SqlDataAdapter(cmd)

Dim dt As New DataTable()

Try

conn.Open()

adapter.Fill(dt)

If dt.Rows.Count > 0 Then

gvStudents.DataSource = dt

gvStudents.DataBind()

pnlRecordAttendance.Visible = True

Else

ShowAlert("No students found for this course.")

pnlRecordAttendance.Visible = False

End If

Catch ex As SqlException

ShowAlert("Database error: " & ex.Message)

Catch ex As Exception

ShowAlert("An error occurred while loading students.")

End Try

End Using

End Using

End Sub

Protected Sub btnSaveAttendance\_Click(sender As Object, e As EventArgs) Handles btnSaveAttendance.Click

If ViewState("CurrentSessionID") Is Nothing Then

ShowAlert("No session found.")

Return

End If

Dim sessionID = Convert.ToInt32(ViewState("CurrentSessionID"))

Using conn As New SqlConnection(connString)

conn.Open()

' Delete old attendance

Using delCmd As New SqlCommand("DELETE FROM StudentAttendance WHERE session\_id=@SessionID", conn)

delCmd.Parameters.AddWithValue("@SessionID", sessionID)

delCmd.ExecuteNonQuery()

End Using

' Insert new attendance

For Each row As GridViewRow In gvStudents.Rows

Dim studentID = Convert.ToInt32(gvStudents.DataKeys(row.RowIndex).Value)

Dim status As String = CType(row.FindControl("ddlStatus"), DropDownList).SelectedValue

Dim notes As String = CType(row.FindControl("txtNotes"), TextBox).Text

Using insCmd As New SqlCommand("INSERT INTO StudentAttendance (session\_id, student\_id, attendance\_status, notes) VALUES (@SID, @StuID, @Status, @Notes)", conn)

insCmd.Parameters.AddWithValue("@SID", sessionID)

insCmd.Parameters.AddWithValue("@StuID", studentID)

insCmd.Parameters.AddWithValue("@Status", status)

insCmd.Parameters.AddWithValue("@Notes", If(String.IsNullOrEmpty(notes), DBNull.Value, notes))

insCmd.ExecuteNonQuery()

End Using

Next

End Using

ShowAlert("Attendance saved.")

pnlRecordAttendance.Visible = False

End Sub

Private Sub ShowAlert(msg As String)

Dim safeMsg As String = msg.Replace("'", "\'")

ClientScript.RegisterStartupScript(Me.GetType(), "alert", "alert('" & safeMsg & "');", True)

End Sub

End Class

**Code explanation:**

This page allows tutors (or admins) to **update attendance records** for tutoring sessions. They can mark students as present or absent for each scheduled session, which reflects in the system's records.