

Name:Shahzaib khan

Semester Project code

Login.aspx:

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

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
  <title>Login to SocialSync</title>
```

```
  <style>
```

```
    body {
```

```
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
      background-color: #f0f2f5;
```

```
      margin: 0;
```

```
      padding: 0;
```

```
      display: flex;
```

```
      justify-content: center;
```

```
      align-items: center;
```

```
      height: 100vh;
```

```
    }
```

```
    .login-container {
```

```
      background-color: white;
```

```
      border-radius: 8px;
```

```
      box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
```

```
      padding: 25px;
```

```
      width: 400px;
```

```
    }
```

```
    h2 {
```

```
      color: #1877f2;
```

```
      margin-bottom: 25px;
```

```
      text-align: center;
```

```
    }
```

```
    .form-group {
```

```
    margin-bottom: 15px;
}
```

```
.form-group label {
    display: block;
    margin-bottom: 5px;
    font-weight: 500;
}
```

```
.form-control {
    width: 100%;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 5px;
    font-size: 14px;
    box-sizing: border-box;
}
```

```
.btn {
    width: 100%;
    padding: 10px;
    background-color: #1877f2;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    font-weight: bold;
    margin-top: 10px;
}
```

```
.btn:hover {
    background-color: #166fe5;
}
```

```
.error-message {
    color: #c62828;
    margin-top: 5px;
    font-size: 13px;
}
```

```
.register-link {
    text-align: center;
    margin-top: 20px;
}
```

```

        font-size: 14px;
    }

    .register-link a {
        color: #1877f2;
        text-decoration: none;
    }

    .register-link a:hover {
        text-decoration: underline;
    }
</style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="login-container">
            <h2>Login to Social Media Platform</h2>

            <asp:Label ID="lblMessage" runat="server" CssClass="error-message"
Visible="false"></asp:Label>

            <div class="form-group">
                <label for="txtEmail">Email</label>
                <asp:TextBox ID="txtEmail" runat="server" CssClass="form-control"
placeholder="Enter your email"></asp:TextBox>
            </div>

            <div class="form-group">
                <label for="txtPassword">Password</label>
                <asp:TextBox ID="txtPassword" runat="server" CssClass="form-control"
TextMode="Password" placeholder="Enter your password"></asp:TextBox>
            </div>

            <asp:Button ID="BtnLogin" runat="server" Text="Login" CssClass="btn"
OnClick="BtnLogin_Click" />

            <div class="register-link">
                Don't have an account? <a href="Registration.aspx">Sign up</a>
            </div>
        </div>
    </form>
</body>
</html>

```

Login.aspx.vb:

Imports System.Data.SqlClient

Imports System.Web.Security

Partial Class Login

Inherits System.Web.UI.Page

' Database connection string

Dim connectionString As String =
ConfigurationManager.ConnectionStrings("Shahzaib").ConnectionString

Protected Sub BtnLogin_Click(sender As Object, e As EventArgs) Handles BtnLogin.Click

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

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

Using conn As New SqlConnection(connectionString)

Try

conn.Open()

Dim query As String = "SELECT ID, USER_NAME, USERNAME, USER_PASSWORD FROM
USERS WHERE USER_EMAIL = @Email"

Using cmd As New SqlCommand(query, conn)

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

Dim reader As SqlDataReader = cmd.ExecuteReader()

If reader.Read() Then

Dim storedPassword As String = reader("USER_PASSWORD").ToString()

Dim userId As Integer = Convert.ToInt32(reader("ID"))

Dim fullName As String = reader("USER_NAME").ToString()

Dim username As String = reader("USERNAME").ToString()

reader.Close()

If VerifyPassword(password, storedPassword) Then

Session.Clear()

Session("UserID") = userId

Session("FullName") = fullName

Session("Username") = username

FormsAuthentication.SetAuthCookie(userId.ToString(), False)

```
Session.Timeout = 20
```

```
    Dim roleQuery As String = "SELECT COUNT(*) FROM GROUP_MEMBERS WHERE  
USER_ID = @UserID AND MEMBER_ROLE = 'Admin'"  
    Using roleCmd As New SqlCommand(roleQuery, conn)  
        roleCmd.Parameters.AddWithValue("@UserID", userId)  
        Dim isAdmin As Integer = Convert.ToInt32(roleCmd.ExecuteScalar())  
        Session("Role") = If(isAdmin > 0, "Admin", "Member")  
    End Using  
  
    Response.Redirect("Home.aspx")  
Else  
    lblMessage.Text = "Invalid email or password."  
    lblMessage.Visible = True  
End If  
Else  
    reader.Close()  
    lblMessage.Text = "No user found with this email."  
    lblMessage.Visible = True  
End If  
End Using  
Catch ex As Exception  
    lblMessage.Text = "Login error: " & ex.Message  
    lblMessage.Visible = True  
Finally  
    conn.Close()  
End Try  
End Using  
End Sub
```

```
Private Function VerifyPassword(password As String, storedPassword As String) As Boolean
```

```
    Return password = storedPassword  
End Function  
End Class
```

Home.aspx:

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

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>SocialSync - Home</title>
  <style>
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background-color: #f0f2f5;
      margin: 0;
      padding: 0;
    }

    .header {
      background-color: #1877f2;
      color: white;
      padding: 8px 16px;
      display: flex;
      justify-content: space-between;
      align-items: center;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
      position: sticky;
      top: 0;
      z-index: 100;
    }

    .logo {
      font-size: 24px;
      font-weight: bold;
    }

    .nav-links a {
      color: white;
      text-decoration: none;
      margin-left: 20px;
      padding: 5px 10px;
      border-radius: 4px;
    }

    .nav-links a:hover {
      background-color: rgba(255, 255, 255, 0.1);
    }

    .container {
```

```
    max-width: 1000px;
    margin: 0 auto;
    padding: 20px;
    display: flex;
}
```

```
.sidebar {
    width: 250px;
    padding-right: 20px;
}
```

```
.profile-card {
    background-color: white;
    border-radius: 8px;
    padding: 15px;
    margin-bottom: 20px;
    box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}
```

```
.profile-info {
    display: flex;
    align-items: center;
    margin-bottom: 15px;
}
```

```
.profile-avatar {
    width: 60px;
    height: 60px;
    border-radius: 50%;
    background-color: #e4e6eb;
    margin-right: 10px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 24px;
    color: #1877f2;
}
```

```
.profile-name {
    font-weight: bold;
    font-size: 18px;
}
```

```
.profile-role {
```

```
font-size: 14px;  
color: #65676b;  
}
```

```
.menu-item {  
  display: flex;  
  align-items: center;  
  padding: 8px 10px;  
  margin-bottom: 5px;  
  border-radius: 8px;  
  cursor: pointer;  
  text-decoration: none;  
  color: #050505;  
}
```

```
.menu-item:hover {  
  background-color: #e4e6eb;  
}
```

```
.menu-icon {  
  margin-right: 10px;  
  width: 20px;  
  height: 20px;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
.main-content {  
  flex: 1;  
}
```

```
.post-composer {  
  background-color: white;  
  border-radius: 8px;  
  padding: 15px;  
  margin-bottom: 20px;  
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);  
}
```

```
.post-input {  
  width: 100%;  
  padding: 12px;  
  border: none;
```

```
border-radius: 18px;
background-color: #f0f2f5;
font-size: 15px;
margin-bottom: 10px;
resize: none;
}

.post-actions {
  display: flex;
  justify-content: space-between;
}

.post-button {
  padding: 8px 16px;
  background-color: #1877f2;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-weight: bold;
}

.post-button:hover {
  background-color: #166fe5;
}

.news-feed {
  margin-bottom: 20px;
}

.post {
  background-color: white;
  border-radius: 8px;
  padding: 15px;
  margin-bottom: 15px;
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}

.post-header {
  display: flex;
  align-items: center;
  margin-bottom: 10px;
}
```

```
.post-avatar {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  background-color: #e4e6eb;
  margin-right: 10px;
}

.post-user {
  font-weight: bold;
}

.post-time {
  font-size: 12px;
  color: #65676b;
  margin-left: 5px;
}

.post-content {
  margin-bottom: 15px;
  font-size: 15px;
  line-height: 1.4;
}

.post-engagement {
  border-top: 1px solid #e4e6eb;
  padding-top: 10px;
  display: flex;
}

.engagement-action {
  flex: 1;
  display: flex;
  align-items: center;
  justify-content: center;
  padding: 6px 0;
  border-radius: 4px;
  cursor: pointer;
}

.engagement-action:hover {
  background-color: #f0f2f5;
}
```

```

.engagement-count {
    margin-left: 5px;
    color: #65676b;
}

.hidden {
    display: none;
}

.message {
    padding: 10px;
    border-radius: 4px;
    margin-bottom: 15px;
}

.success-message {
    background-color: #e8f5e9;
    color: #2e7d32;
    border: 1px solid #c8e6c9;
}

.error-message {
    background-color: #ffebee;
    color: #c62828;
    border: 1px solid #ffcdd2;
}
</style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="header">
            <div class="logo">SocialSync</div>
            <div class="nav-links">
                <asp:HyperLink ID="InkHome" runat="server"
NavigateUrl="Home.aspx">Home</asp:HyperLink>
                <asp:LinkButton ID="InkLogout" runat="server"
OnClick="InkLogout_Click">Logout</asp:LinkButton>
            </div>
        </div>

        <div class="container">
            <div class="sidebar">
                <div class="profile-card">
                    <div class="profile-info">

```

```

<div class="profile-avatar">
    <asp:Literal ID="litInitial" runat="server"></asp:Literal>
</div>
<div>
    <div class="profile-name">
        <asp:Label ID="lblUsername" runat="server"></asp:Label>
    </div>
    <div class="profile-role">
        <asp:Label ID="lblRole" runat="server"></asp:Label>
    </div>
</div>
</div>
</div>

```

```

    <asp:HyperLink ID="lnkPosts" runat="server" NavigateUrl="Posts.aspx"
    CssClass="menu-item">
        <div class="menu-icon"></div>
        <span>My Posts</span>
    </asp:HyperLink>

```

```

    <asp:Panel ID="pnlAdminMenu" runat="server" Visible="false">
        <asp:HyperLink ID="lnkAdmin" runat="server" NavigateUrl="AdminPanel.aspx"
        CssClass="menu-item">
            <div class="menu-icon"></div>
            <span>Admin Panel</span>
        </asp:HyperLink>
    </asp:Panel>
</div>

```

```

<div class="main-content">
    <asp:Label ID="lblMessage" runat="server" CssClass="message success-message"
    Visible="false"></asp:Label>

```

```

    <div class="post-composer">
        <asp:TextBox ID="txtPostContent" runat="server" CssClass="post-input"
        TextMode="MultiLine"
        Rows="3" placeholder="What's on your mind?"></asp:TextBox>
        <div class="post-actions">
            <div></div>
            <asp:Button ID="btnPost" runat="server" Text="Post" CssClass="post-button"
            OnClick="btnPost_Click" />
        </div>
    </div>

```

```

        </div>

        <div class="news-feed">
            <h3>News Feed</h3>
            <asp:Literal ID="litPosts" runat="server"></asp:Literal>
        </div>
    </div>
</div>
</form>
</body>
</html>

```

Home.aspx.vb:

Imports System.Data.SqlClient

Partial Class Home

Inherits System.Web.UI.Page

' Database connection string

Dim connectionString As String =
ConfigurationManager.ConnectionStrings("Shahzaib").ConnectionString

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

' Check if user is logged in

If Session("UserID") Is Nothing Then

Response.Redirect("Login.aspx")

Return

End If

' Set username and role info

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

lblRole.Text = Session("Role").ToString()

' Set the user initial for avatar

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

If Not String.IsNullOrEmpty(username) Then

litInitial.Text = username.Substring(0, 1).ToUpper()

End If

' Show admin panel link if user is admin

pnlAdminMenu.Visible = (Session("Role").ToString() = "Admin")

```

' Check for post success message from redirect
If Not IsPostBack Then
    If Request.QueryString("postSuccess") = "true" Then
        lblMessage.Text = "Your post was created successfully!"
        lblMessage.Visible = True
    End If

    ' Load news feed
    LoadNewsFeed()
End If
End Sub

Protected Sub btnPost_Click(sender As Object, e As EventArgs) Handles btnPost.Click
    Dim postContent As String = txtPostContent.Text.Trim()

    ' Validate post content
    If String.IsNullOrEmpty(postContent) Then
        lblMessage.Text = "Post content cannot be empty."
        lblMessage.CssClass = "message error-message"
        lblMessage.Visible = True
        Return
    End If

    ' Get current user ID
    Dim userId As Integer = Convert.ToInt32(Session("UserID"))

    Using conn As New SqlConnection(connectionString)
        Try
            conn.Open()

            ' Get the next post ID
            Dim newPostId As Integer = 1
            Dim idQuery As String = "SELECT ISNULL(MAX(POST_ID), 0) + 1 FROM POSTS"
            Using idCmd As New SqlCommand(idQuery, conn)
                newPostId = Convert.ToInt32(idCmd.ExecuteScalar())
            End Using

            ' Insert the new post
            Dim insertQuery As String = "INSERT INTO POSTS (POST_ID, POST_CONTENT,
POST_TYPE, CREATED_AT, UPDATED_AT, USER_ID) " &
                "VALUES (@PostId, @Content, 'text', GETDATE(), GETDATE(), @UserId)"
            Using cmd As New SqlCommand(insertQuery, conn)
                cmd.Parameters.AddWithValue("@PostId", newPostId)
            End Using
        Catch ex As Exception
            ' Handle exception
        End Try
    End Using
End Sub

```

```

        cmd.Parameters.AddWithValue("@Content", postContent)
        cmd.Parameters.AddWithValue("@UserId", userId)
        cmd.ExecuteNonQuery()
    End Using

    ' Clear the text box and reload feed
    txtPostContent.Text = String.Empty
    LoadNewsFeed()

    ' Show success message
    lblMessage.Text = "Your post was created successfully!"
    lblMessage.CssClass = "message success-message"
    lblMessage.Visible = True

Catch ex As Exception
    lblMessage.Text = "Error creating post: " & ex.Message
    lblMessage.CssClass = "message error-message"
    lblMessage.Visible = True
Finally
    conn.Close()
End Try
End Using
End Sub

Private Sub LoadNewsFeed()
    Dim userId As Integer = Convert.ToInt32(Session("UserID"))
    Dim postsHtml As New StringBuilder()

    Using conn As New SqlConnection(connectionString)
        Try
            conn.Open()

            ' Query to get posts from current user and people they follow
            Dim query As String = "SELECT P.POST_ID, P.POST_CONTENT, P.CREATED_AT,
U.USER_NAME, U.ID AS USER_ID, " &
                "(SELECT COUNT(*) FROM LIKES WHERE POST_ID = P.POST_ID) AS
LIKE_COUNT, " &
                "(SELECT COUNT(*) FROM COMMENTS WHERE POST_ID = P.POST_ID) AS
COMMENT_COUNT, " &
                "(SELECT COUNT(*) FROM LIKES WHERE POST_ID = P.POST_ID AND
USER_ID = @UserId) AS USER_LIKED " &
                "FROM POSTS P " &
                "INNER JOIN USERS U ON P.USER_ID = U.ID " &
                "WHERE P.USER_ID = @UserId " &

```

```

"OR P.USER_ID IN (SELECT FOLLOWING_ID FROM FOLLOWING_T WHERE
FOLLOWER_ID = @UserId) " &
"ORDER BY P.CREATED_AT DESC"

```

```

Using cmd As New SqlCommand(query, conn)
cmd.Parameters.AddWithValue("@UserId", userId)

```

```

Using reader As SqlDataReader = cmd.ExecuteReader()
If Not reader.HasRows Then
    postsHtml.Append("<div class='post'>No posts in your feed. Follow more users
or create a post!</div>")

```

```

Else
    While reader.Read()
        Dim postId As Integer = Convert.ToInt32(reader("POST_ID"))
        Dim content As String = reader("POST_CONTENT").ToString()
        Dim createdAt As DateTime = Convert.ToDateTime(reader("CREATED_AT"))
        Dim postUsername As String = reader("USER_NAME").ToString()
        Dim postId As Integer = Convert.ToInt32(reader("USER_ID"))
        Dim likeCount As Integer = Convert.ToInt32(reader("LIKE_COUNT"))
        Dim commentCount As Integer =
Convert.ToInt32(reader("COMMENT_COUNT"))
        Dim userLiked As Boolean = (Convert.ToInt32(reader("USER_LIKED")) > 0)

        ' Format the time ago string
        Dim timeAgo As String = GetTimeAgo(createdAt)

        ' Format user initial for avatar
        Dim initial As String = If(postUsername.Length > 0, postUsername.Substring(0,
1).ToUpper(), "?")

```

```

        ' Build post HTML
        postsHtml.AppendFormat("<div class='post' id='post-{0}'>", postId)
        postsHtml.Append("<div class='post-header'>")
        postsHtml.AppendFormat("<div class='post-avatar'>{0}</div>", initial)
        postsHtml.Append("<div>")
        postsHtml.AppendFormat("<span class='post-user'>{0}</span>",
postUsername)
        postsHtml.AppendFormat("<span class='post-time'>{0}</span>", timeAgo)
        postsHtml.Append("</div>")
        postsHtml.Append("</div>")

        postsHtml.AppendFormat("<div class='post-content'>{0}</div>", content)

        postsHtml.Append("<div class='post-engagement'>")

```

```

        ' Like button
        postsHtml.AppendFormat("<a href='Posts.aspx?action=like&postId={0}'
class='engagement-action'>", postId)
        postsHtml.AppendFormat("<span> Like{0}</span>", If(userLiked, "d", ""))
        postsHtml.AppendFormat("<span class='engagement-count'>{0}</span>",
likeCount)
        postsHtml.Append("</a>")

        ' Comment button
        postsHtml.AppendFormat("<a href='Posts.aspx?postId={0}#comments'
class='engagement-action'>", postId)
        postsHtml.Append("<span> Comment</span>")
        postsHtml.AppendFormat("<span class='engagement-count'>{0}</span>",
commentCount)
        postsHtml.Append("</a>")

        postsHtml.Append("</div>")
        postsHtml.Append("</div>")
    End While
End If
End Using
End Using

Catch ex As Exception
    postsHtml.AppendFormat("<div class='message error-message'>Error loading feed:
{0}</div>", ex.Message)
Finally
    conn.Close()
End Try
End Using

litPosts.Text = postsHtml.ToString()
End Sub

Private Function GetTimeAgo(dateTime As DateTime) As String
    Dim timeSpan As TimeSpan = DateTime.Now - dateTime

    If timeSpan.TotalMinutes < 1 Then
        Return "Just now"
    ElseIf timeSpan.TotalMinutes < 60 Then
        Return String.Format("{0} minute{1} ago", Math.Floor(timeSpan.TotalMinutes),
If(Math.Floor(timeSpan.TotalMinutes) = 1, "", "s"))
    ElseIf timeSpan.TotalHours < 24 Then

```

```

        Return String.Format("{0} hour{1} ago", Math.Floor(timeSpan.TotalHours),
If(Math.Floor(timeSpan.TotalHours) = 1, "", "s"))
        Elself timeSpan.TotalDays < 7 Then
            Return String.Format("{0} day{1} ago", Math.Floor(timeSpan.TotalDays),
If(Math.Floor(timeSpan.TotalDays) = 1, "", "s"))
        Else
            Return dateTime.ToString("MMM dd, yyyy")
        End If
    End Function

```

```

Protected Sub InkLogout_Click(sender As Object, e As EventArgs) Handles InkLogout.Click
    ' Clear session and redirect to login page
    Session.Clear()
    Session.Abandon()
    FormsAuthentication.SignOut()
    Response.Redirect("Login.aspx")
End Sub
End Class

```

Posts.aspx:

```

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

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head runat="server">

```

```

    <title>SocialSync - Posts</title>

```

```

    <style>

```

```

        body {

```

```

            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

```

```

            background-color: #f0f2f5;

```

```

            margin: 0;

```

```

            padding: 0;

```

```

        }

```

```
.header {  
  background-color: #1877f2;  
  color: white;  
  padding: 8px 16px;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
  position: sticky;  
  top: 0;  
  z-index: 100;  
}
```

```
.logo {  
  font-size: 24px;  
  font-weight: bold;  
}
```

```
.nav-links a {  
  color: white;  
  text-decoration: none;  
  margin-left: 20px;  
  padding: 5px 10px;  
  border-radius: 4px;  
}
```

```
.nav-links a:hover {
```

```
background-color: rgba(255, 255, 255, 0.1);  
}
```

```
.container {  
  max-width: 1000px;  
  margin: 0 auto;  
  padding: 20px;  
}
```

```
.section-title {  
  font-size: 24px;  
  margin-bottom: 20px;  
  color: #1c1e21;  
}
```

```
.post-composer {  
  background-color: white;  
  border-radius: 8px;  
  padding: 15px;  
  margin-bottom: 20px;  
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);  
}
```

```
.post-input {  
  width: 100%;  
  padding: 12px;  
  border: none;
```

```
border-radius: 18px;
background-color: #f0f2f5;
font-size: 15px;
margin-bottom: 10px;
resize: none;
}
```

```
.post-actions {
  display: flex;
  justify-content: space-between;
}
```

```
.post-button {
  padding: 8px 16px;
  background-color: #1877f2;
  color: white;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-weight: bold;
}
```

```
.post-button:hover {
  background-color: #166fe5;
}
```

```
.post {
```

```
background-color: white;
border-radius: 8px;
padding: 15px;
margin-bottom: 15px;
box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}
```

```
.post-header {
  display: flex;
  align-items: center;
  margin-bottom: 10px;
}
```

```
.post-actions-menu {
  margin-left: auto;
  position: relative;
}
```

```
.post-menu-button {
  border: none;
  background: none;
  cursor: pointer;
  font-size: 20px;
  color: #65676b;
}
```

```
.post-avatar {
```

```
width: 40px;
height: 40px;
border-radius: 50%;
background-color: #e4e6eb;
margin-right: 10px;
display: flex;
align-items: center;
justify-content: center;
color: #1877f2;
font-weight: bold;
}
```

```
.post-user {
  font-weight: bold;
}
```

```
.post-time {
  font-size: 12px;
  color: #65676b;
  margin-left: 5px;
}
```

```
.post-content {
  margin-bottom: 15px;
  font-size: 15px;
  line-height: 1.4;
}
```

```
.post-engagement {  
  border-top: 1px solid #e4e6eb;  
  padding-top: 10px;  
  display: flex;  
}
```

```
.engagement-action {  
  flex: 1;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  padding: 6px 0;  
  border-radius: 4px;  
  cursor: pointer;  
  text-decoration: none;  
  color: #65676b;  
}
```

```
.engagement-action:hover {  
  background-color: #f0f2f5;  
}
```

```
.engagement-count {  
  margin-left: 5px;  
  color: #65676b;  
}
```

```
.comment-section {  
    margin-top: 10px;  
    border-top: 1px solid #e4e6eb;  
    padding-top: 10px;  
}
```

```
.comment {  
    display: flex;  
    margin-bottom: 10px;  
    padding: 5px 0;  
}
```

```
.comment-avatar {  
    width: 32px;  
    height: 32px;  
    border-radius: 50%;  
    background-color: #e4e6eb;  
    margin-right: 10px;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    color: #1877f2;  
    font-size: 12px;  
}
```

```
.comment-bubble {
```

```
background-color: #f0f2f5;  
border-radius: 18px;  
padding: 8px 12px;  
max-width: calc(100% - 42px);  
}
```

```
.comment-user {  
    font-weight: bold;  
    font-size: 13px;  
}
```

```
.comment-text {  
    font-size: 13px;  
}
```

```
.comment-time {  
    font-size: 11px;  
    color: #65676b;  
    margin-top: 3px;  
}
```

```
.comment-box {  
    display: flex;  
    margin-top: 10px;  
}
```

```
.comment-input {
```

```
flex: 1;
padding: 8px 12px;
border: none;
border-radius: 18px;
background-color: #f0f2f5;
font-size: 13px;
}
```

```
.admin-section {
  background-color: white;
  border-radius: 8px;
  padding: 15px;
  margin-top: 30px;
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}
```

```
.admin-title {
  font-size: 18px;
  font-weight: bold;
  margin-bottom: 15px;
  color: #1c1e21;
}
```

```
.user-list {
  margin-top: 15px;
}
```

```
.user-item {  
  padding: 10px;  
  border: 1px solid #e4e6eb;  
  border-radius: 8px;  
  margin-bottom: 10px;  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}
```

```
.user-info {  
  display: flex;  
  align-items: center;  
}
```

```
.user-avatar {  
  width: 36px;  
  height: 36px;  
  border-radius: 50%;  
  background-color: #e4e6eb;  
  margin-right: 10px;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  color: #1877f2;  
}
```

```
.user-name {  
    font-weight: 500;  
}
```

```
.user-email {  
    font-size: 12px;  
    color: #65676b;  
}
```

```
.delete-button {  
    padding: 5px 10px;  
    background-color: #f5245f;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    font-size: 12px;  
}
```

```
.delete-button:hover {  
    background-color: #e41e4f;  
}
```

```
.message {  
    padding: 10px;  
    border-radius: 4px;  
    margin-bottom: 15px;
```

```
}
```

```
.success-message {  
    background-color: #e8f5e9;  
    color: #2e7d32;  
    border: 1px solid #c8e6c9;  
}
```

```
.error-message {  
    background-color: #ffebee;  
    color: #c62828;  
    border: 1px solid #ffcdd2;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div class="header">
```

```
<div class="logo">SocialSync</div>
```

```
<div class="nav-links">
```

```
<asp:HyperLink ID="InkHome" runat="server"  
NavigateUrl="Home.aspx">Home</asp:HyperLink>
```

```
<asp:LinkButton ID="InkLogout" runat="server"  
OnClick="InkLogout_Click">Logout</asp:LinkButton>
```

```
</div>
```

```
</div>
```

```
<div class="container">

    <asp:Label ID="lblMessage" runat="server" CssClass="message success-message"
Visible="false"></asp:Label>

    <h2 class="section-title">My Posts</h2>

    <div class="post-composer">

        <asp:TextBox ID="txtPostContent" runat="server" CssClass="post-input"
TextMode="MultiLine"

            Rows="3" placeholder="What's on your mind?"></asp:TextBox>

        <div class="post-actions">

            <div></div>

            <asp:Button ID="btnCreatePost" runat="server" Text="Create Post" CssClass="post-
button" OnClick="btnCreatePost_Click" />

        </div>

    </div>

    <div id="posts-container">

        <asp:Literal ID="litPosts" runat="server"></asp:Literal>

    </div>

    <!-- Admin section for managing users -->

    <asp:Panel ID="pnlAdminSection" runat="server" CssClass="admin-section" Visible="false">

        <h3 class="admin-title">Admin: Manage Users</h3>

        <div class="user-list">

            <asp:Repeater ID="rptUsers" runat="server"
OnItemCommand="rptUsers_ItemCommand">

                <ItemTemplate>
```

```

<div class="user-item">
    <div class="user-info">
        <div class="user-avatar"><%# Eval("Username").ToString().Substring(0,
1).ToUpper() %></div>
        <div>
            <div class="user-name"><%# Eval("Username") %></div>
            <div class="user-email"><%# Eval("Email") %></div>
        </div>
    </div>
    <asp:Button ID="btnDeleteUser" runat="server" Text="Delete User"
        CssClass="delete-button"
        CommandName="DeleteUser"
        CommandArgument='<%# Eval("UserID") %>'
        OnClientClick="return confirm('Are you sure you want to delete this user? This
action cannot be undone.');" />
    </div>
</ItemTemplate>
</asp:Repeater>
<asp:Panel ID="pnlNoUsers" runat="server" Visible="false">
    <div class="post">No users found that can be managed.</div>
</asp:Panel>
</div>
</form>
</body>

```

</html>

Posts.aspx.vb:

Imports System.Data

Imports System.Data.SqlClient

Partial Class Posts

 Inherits System.Web.UI.Page

 ' Database connection string

 Dim connectionString As String =
 ConfigurationManager.ConnectionStrings("Shahzaib").ConnectionString

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

 ' Check if user is logged in

 If Session("UserID") Is Nothing Then

 Response.Redirect("Login.aspx")

 Return

 End If

 ' Process actions from URL parameters (like, delete)

 If Not IsPostBack Then

 ProcessQueryStringActions()

 ' Load user posts

 LoadUserPosts()

```
' Load admin section if user is admin
If Session("Role").ToString() = "Admin" Then
    pnlAdminSection.Visible = True
    LoadUsers()
End If
End If
End Sub
```

Protected Sub btnCreatePost_Click(sender As Object, e As EventArgs) Handles
btnCreatePost.Click

```
Dim postContent As String = txtPostContent.Text.Trim()
```

```
' Validate post content
```

```
If String.IsNullOrEmpty(postContent) Then
```

```
    lblMessage.Text = "Post content cannot be empty."
```

```
    lblMessage.CssClass = "message error-message"
```

```
    lblMessage.Visible = True
```

```
    Return
```

```
End If
```

```
' Get current user ID
```

```
Dim userId As Integer = Convert.ToInt32(Session("UserID"))
```

```
Using conn As New SqlConnection(connectionString)
```

```
    Try
```

```
        conn.Open()
```

' Get the next post ID

Dim newPostId As Integer = 1

Dim idQuery As String = "SELECT ISNULL(MAX(POST_ID), 0) + 1 FROM POSTS"

Using idCmd As New SqlCommand(idQuery, conn)

newPostId = Convert.ToInt32(idCmd.ExecuteScalar())

End Using

' Insert the new post

Dim insertQuery As String = "INSERT INTO POSTS (POST_ID, POST_CONTENT,
POST_TYPE, CREATED_AT, UPDATED_AT, USER_ID) " &

"VALUES (@PostId, @Content, 'text', GETDATE(), GETDATE(), @UserId)"

Using cmd As New SqlCommand(insertQuery, conn)

cmd.Parameters.AddWithValue("@PostId", newPostId)

cmd.Parameters.AddWithValue("@Content", postContent)

cmd.Parameters.AddWithValue("@UserId", userId)

cmd.ExecuteNonQuery()

End Using

' Clear the text box and reload the posts

txtPostContent.Text = String.Empty

lblMessage.Text = "Post created successfully!"

lblMessage.CssClass = "message success-message"

lblMessage.Visible = True

' Reload posts

LoadUserPosts()

Catch ex As Exception

lblMessage.Text = "Error creating post: " & ex.Message

lblMessage.CssClass = "message error-message"

lblMessage.Visible = True

Finally

conn.Close()

End Try

End Using

End Sub

Private Sub ProcessQueryStringActions()

' Handle post like actions

If Request.QueryString("action") = "like" And Not
String.IsNullOrEmpty(Request.QueryString("postId")) Then

Dim postId As Integer

If Integer.TryParse(Request.QueryString("postId"), postId) Then

ToggleLikePost(postId)

End If

End If

' Handle post delete actions

If Request.QueryString("action") = "delete" And Not
String.IsNullOrEmpty(Request.QueryString("postId")) Then

Dim postId As Integer

If Integer.TryParse(Request.QueryString("postId"), postId) Then

DeletePost(postId)

End If

End If

End Sub

Private Sub LoadUserPosts()

Dim userId As Integer = Convert.ToInt32(Session("UserID"))

Dim postsHtml As New StringBuilder()

Using conn As New SqlConnection(connectionString)

Try

conn.Open()

' Query to get user's posts with like and comment counts

Dim query As String = "SELECT P.POST_ID, P.POST_CONTENT, P.CREATED_AT, " &
 "(SELECT COUNT(*) FROM LIKES WHERE POST_ID = P.POST_ID) AS
LIKE_COUNT, " &
 "(SELECT COUNT(*) FROM COMMENTS WHERE POST_ID = P.POST_ID) AS
COMMENT_COUNT, " &
 "(SELECT COUNT(*) FROM LIKES WHERE POST_ID = P.POST_ID AND
USER_ID = @UserId) AS USER_LIKED " &
 "FROM POSTS P " &
 "WHERE P.USER_ID = @UserId " &
 "ORDER BY P.CREATED_AT DESC"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@UserId", userId)

Using reader As SqlDataReader = cmd.ExecuteReader()

If Not reader.HasRows Then

postsHtml.Append("<div class='post'>You haven't created any posts yet.</div>")

Else

While reader.Read()

Dim postId As Integer = Convert.ToInt32(reader("POST_ID"))

Dim content As String = reader("POST_CONTENT").ToString()

Dim createdAt As DateTime = Convert.ToDateTime(reader("CREATED_AT"))

Dim likeCount As Integer = Convert.ToInt32(reader("LIKE_COUNT"))

Dim commentCount As Integer =
Convert.ToInt32(reader("COMMENT_COUNT"))

Dim userLiked As Boolean = (Convert.ToInt32(reader("USER_LIKED")) > 0)

' Get user info for the post

Dim userInitial As String = Session("Username").ToString().Substring(0,
1).ToUpper()

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

' Format the time ago string

Dim timeAgo As String = GetTimeAgo(createdAt)

' Build post HTML

postsHtml.AppendFormat("<div class='post' id='post-{0}'>", postId)

postsHtml.Append("<div class='post-header'>")

postsHtml.AppendFormat("<div class='post-avatar'>{0}</div>", userInitial)

postsHtml.Append("<div>")

postsHtml.AppendFormat("{0}", username)

postsHtml.AppendFormat("{0}", timeAgo)

postsHtml.Append("</div>")

' Add post actions menu (delete)

```

        postsHtml.Append("<div class='post-actions-menu'>")

        postsHtml.AppendFormat("<a href='Posts.aspx?action=delete&postId={0}'
onclick='return confirm(\"\"Are you sure you want to delete this post?\"\");' class='post-menu-
button'>:</a>", postId)

        postsHtml.Append("</div>")

        postsHtml.Append("</div>")

        postsHtml.AppendFormat("<div class='post-content'>{0}</div>", content)

        postsHtml.Append("<div class='post-engagement'>")

        ' Like button

        postsHtml.AppendFormat("<a href='Posts.aspx?action=like&postId={0}'
class='engagement-action'>", postId)

        postsHtml.AppendFormat("<span> Like{0}</span>", If(userLiked, "d", ""))

        postsHtml.AppendFormat("<span class='engagement-count'>{0}</span>",
likeCount)

        postsHtml.Append("</a>")

        ' Comment button/link

        postsHtml.AppendFormat("<a href='Posts.aspx?postId={0}#comments'
class='engagement-action'>", postId)

        postsHtml.Append("<span> Comment</span>")

        postsHtml.AppendFormat("<span class='engagement-count'>{0}</span>",
commentCount)

        postsHtml.Append("</a>")

```

```

        postsHtml.Append("</div>")

        ' Load comments if this is the active post
        If Not String.IsNullOrEmpty(Request.QueryString("postId")) Then

            Dim activePostId As Integer

            If Integer.TryParse(Request.QueryString("postId"), activePostId) AndAlso
activePostId = postId Then

                postsHtml.Append(LoadCommentsForPost(postId))

            End If

        End If

        postsHtml.Append("</div>")

    End While

End If

End Using

End Using

Catch ex As Exception

    postsHtml.AppendFormat("<div class='message error-message'>Error loading posts:
{0}</div>", ex.Message)

    Finally

        conn.Close()

    End Try

End Using

litPosts.Text = postsHtml.ToString()

End Sub

```

```
Private Function LoadCommentsForPost(postId As Integer) As String
```

```
    Dim commentsHtml As New StringBuilder()
```

```
    commentsHtml.Append("<div class='comment-section' id='comments'>")
```

```
    commentsHtml.Append("<h4>Comments</h4>")
```

```
    Using conn As New SqlConnection(connectionString)
```

```
        Try
```

```
            conn.Open()
```

```
            ' Get comments for this post
```

```
            Dim query As String = "SELECT C.COMMENT_CONTENT, C.CREATED_AT,  
U.USER_NAME, U.USERNAME " &
```

```
                "FROM COMMENTS C " &
```

```
                "INNER JOIN USERS U ON C.USER_ID = U.ID " &
```

```
                "WHERE C.POST_ID = @PostId " &
```

```
                "ORDER BY C.CREATED_AT ASC"
```

```
            Using cmd As New SqlCommand(query, conn)
```

```
                cmd.Parameters.AddWithValue("@PostId", postId)
```

```
            Using reader As SqlDataReader = cmd.ExecuteReader()
```

```
                If Not reader.HasRows Then
```

```
                    commentsHtml.Append("<div>No comments yet. Be the first to  
comment!</div>")
```

```
                Else
```

```
                    While reader.Read()
```

```
                        Dim commentContent As String = reader("COMMENT_CONTENT").ToString()
```

```

        Dim commentCreatedAt As DateTime =
Convert.ToDateTime(reader("CREATED_AT"))

        Dim commentUsername As String = reader("USER_NAME").ToString()

        ' Format user initial for avatar
        Dim initial As String = commentUsername.Substring(0, 1).ToUpper()

        ' Format time ago
        Dim timeAgo As String = GetTimeAgo(commentCreatedAt)

        ' Build comment HTML
        commentsHtml.Append("<div class='comment'>")
        commentsHtml.AppendFormat("<div class='comment-avatar'>{0}</div>",
initial)
        commentsHtml.Append("<div class='comment-bubble'>")
        commentsHtml.AppendFormat("<div class='comment-user'>{0}</div>",
commentUsername)
        commentsHtml.AppendFormat("<div class='comment-text'>{0}</div>",
commentContent)
        commentsHtml.AppendFormat("<div class='comment-time'>{0}</div>",
timeAgo)
        commentsHtml.Append("</div>")
        commentsHtml.Append("</div>")
    End While
End If
End Using
End Using

' Add comment form

```

```

        commentsHtml.AppendFormat("<div class='comment-box'>")
        commentsHtml.AppendFormat("<div class='comment-avatar'>{0}</div>",
Session("Username").ToString().Substring(0, 1).ToUpper())
        commentsHtml.AppendFormat("<input type='text' class='comment-input'
id='txtComment-{0}' placeholder='Write a comment...' />", postId)
        commentsHtml.AppendFormat("<button type='button' onclick='addComment({0})'
class='post-button'>Post</button>", postId)
        commentsHtml.Append("</div>")

```

Catch ex As Exception

```

        commentsHtml.AppendFormat("<div class='message error-message'>Error loading
comments: {0}</div>", ex.Message)

```

Finally

```

        conn.Close()

```

End Try

End Using

```

commentsHtml.Append("</div>")

```

```

Return commentsHtml.ToString()

```

End Function

Private Sub ToggleLikePost(postId As Integer)

```

    Dim userId As Integer = Convert.ToInt32(Session("UserID"))

```

```

Using conn As New SqlConnection(connectionString)

```

Try

```

    conn.Open()

```

' Check if user already liked the post

```
Dim checkQuery As String = "SELECT COUNT(*) FROM LIKES WHERE POST_ID =  
@PostId AND USER_ID = @UserId"
```

```
Using checkCmd As New SqlCommand(checkQuery, conn)
```

```
    checkCmd.Parameters.AddWithValue("@PostId", postId)
```

```
    checkCmd.Parameters.AddWithValue("@UserId", userId)
```

```
Dim likeExists As Integer = Convert.ToInt32(checkCmd.ExecuteScalar())
```

If likeExists > 0 Then

' Unlike: Remove the like

```
Dim deleteQuery As String = "DELETE FROM LIKES WHERE POST_ID = @PostId  
AND USER_ID = @UserId"
```

```
Using deleteCmd As New SqlCommand(deleteQuery, conn)
```

```
    deleteCmd.Parameters.AddWithValue("@PostId", postId)
```

```
    deleteCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deleteCmd.ExecuteNonQuery()
```

```
End Using
```

Else

' Like: Add a new like

```
Dim insertQuery As String = "INSERT INTO LIKES (POST_ID, USER_ID) VALUES  
(@PostId, @UserId)"
```

```
Using insertCmd As New SqlCommand(insertQuery, conn)
```

```
    insertCmd.Parameters.AddWithValue("@PostId", postId)
```

```
    insertCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    insertCmd.ExecuteNonQuery()
```

```
End Using
```

```

        End If
    End Using

    Catch ex As Exception
        lblMessage.Text = "Error processing like: " & ex.Message
        lblMessage.CssClass = "message error-message"
        lblMessage.Visible = True
    Finally
        conn.Close()
    End Try
End Using
End Sub

Private Sub DeletePost(postId As Integer)
    Dim userId As Integer = Convert.ToInt32(Session("UserID"))

    Using conn As New SqlConnection(connectionString)
        Try
            conn.Open()

            ' Check if user owns the post or is admin
            Dim checkQuery As String = "SELECT USER_ID FROM POSTS WHERE POST_ID =
@PostId"

            Using checkCmd As New SqlCommand(checkQuery, conn)
                checkCmd.Parameters.AddWithValue("@PostId", postId)

                Dim postUserId As Object = checkCmd.ExecuteScalar()

```

```

        ' Verify user owns the post or is admin

        If postUserId IsNot Nothing AndAlso (Convert.ToInt32(postUserId) = userId OrElse
Session("Role").ToString() = "Admin") Then

            ' Delete associated comments

            Dim deleteCommentsQuery As String = "DELETE FROM COMMENTS WHERE
POST_ID = @PostId"

            Using deleteCommentsCmd As New SqlCommand(deleteCommentsQuery, conn)

                deleteCommentsCmd.Parameters.AddWithValue("@PostId", postId)

                deleteCommentsCmd.ExecuteNonQuery()

            End Using

            ' Delete associated likes

            Dim deleteLikesQuery As String = "DELETE FROM LIKES WHERE POST_ID =
@PostId"

            Using deleteLikesCmd As New SqlCommand(deleteLikesQuery, conn)

                deleteLikesCmd.Parameters.AddWithValue("@PostId", postId)

                deleteLikesCmd.ExecuteNonQuery()

            End Using

            ' Delete the post

            Dim deletePostQuery As String = "DELETE FROM POSTS WHERE POST_ID =
@PostId"

            Using deletePostCmd As New SqlCommand(deletePostQuery, conn)

                deletePostCmd.Parameters.AddWithValue("@PostId", postId)

                deletePostCmd.ExecuteNonQuery()

            End Using

```

```
lblMessage.Text = "Post deleted successfully!"  
lblMessage.CssClass = "message success-message"  
lblMessage.Visible = True
```

```
Else
```

```
lblMessage.Text = "You don't have permission to delete this post."  
lblMessage.CssClass = "message error-message"  
lblMessage.Visible = True
```

```
End If
```

```
End Using
```

```
Catch ex As Exception
```

```
lblMessage.Text = "Error deleting post: " & ex.Message  
lblMessage.CssClass = "message error-message"  
lblMessage.Visible = True
```

```
Finally
```

```
conn.Close()
```

```
End Try
```

```
End Using
```

```
End Sub
```

```
Private Sub LoadUsers()
```

```
Using conn As New SqlConnection(connectionString)
```

```
Try
```

```
conn.Open()
```

```
' Get all users except current user
```

```
Dim query As String = "SELECT ID AS UserID, USER_NAME AS Username, USER_EMAIL  
AS Email " &
```

```
"FROM USERS WHERE ID <> @CurrentUserId"
```

```
Using cmd As New SqlCommand(query, conn)
```

```
cmd.Parameters.AddWithValue("@CurrentUserId",  
Convert.ToInt32(Session("UserID")))
```

```
Dim adapter As New SqlDataAdapter(cmd)
```

```
Dim usersTable As New DataTable()
```

```
adapter.Fill(usersTable)
```

```
rptUsers.DataSource = usersTable
```

```
rptUsers.DataBind()
```

```
End Using
```

```
Catch ex As Exception
```

```
lblMessage.Text = "Error loading users: " & ex.Message
```

```
lblMessage.CssClass = "message error-message"
```

```
lblMessage.Visible = True
```

```
Finally
```

```
conn.Close()
```

```
End Try
```

```
End Using
```

```
End Sub
```

```
Protected Sub rptUsers_ItemCommand(source As Object, e As RepeaterCommandEventArgs)
```

```
If e.CommandName = "DeleteUser" Then
```

```

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

        DeleteUser(userId)

    End If

End Sub

Private Sub DeleteUser(userId As Integer)

    ' Only admins can delete users

    If Session("Role").ToString() <> "Admin" Then

        lblMessage.Text = "You don't have permission to delete users."

        lblMessage.CssClass = "message error-message"

        lblMessage.Visible = True

        Return

    End If

    Using conn As New SqlConnection(connectionString)

        Try

            conn.Open()

            ' Start a transaction since we need to delete multiple related records

            Dim transaction As SqlTransaction = conn.BeginTransaction()

            Try

                ' Delete user's comments

                Dim deleteCommentsQuery As String = "DELETE FROM COMMENTS WHERE USER_ID
= @UserId"

                Using deleteCommentsCmd As New SqlCommand(deleteCommentsQuery, conn,
transaction)

                    deleteCommentsCmd.Parameters.AddWithValue("@UserId", userId)

```

```
deleteCommentsCmd.ExecuteNonQuery()
```

```
End Using
```

```
' Delete user's likes
```

```
Dim deleteLikesQuery As String = "DELETE FROM LIKES WHERE USER_ID = @UserId"
```

```
Using deleteLikesCmd As New SqlCommand(deleteLikesQuery, conn, transaction)
```

```
deleteLikesCmd.Parameters.AddWithValue("@UserId", userId)
```

```
deleteLikesCmd.ExecuteNonQuery()
```

```
End Using
```

```
' Delete user's followers/following relationships
```

```
Dim deleteFollowingQuery As String = "DELETE FROM FOLLOWING_T WHERE  
FOLLOWER_ID = @UserId OR FOLLOWING_ID = @UserId"
```

```
Using deleteFollowingCmd As New SqlCommand(deleteFollowingQuery, conn,  
transaction)
```

```
deleteFollowingCmd.Parameters.AddWithValue("@UserId", userId)
```

```
deleteFollowingCmd.ExecuteNonQuery()
```

```
End Using
```

```
' Delete user's group memberships
```

```
Dim deleteGroupMembersQuery As String = "DELETE FROM GROUP_MEMBERS  
WHERE USER_ID = @UserId"
```

```
Using deleteGroupMembersCmd As New SqlCommand(deleteGroupMembersQuery,  
conn, transaction)
```

```
deleteGroupMembersCmd.Parameters.AddWithValue("@UserId", userId)
```

```
deleteGroupMembersCmd.ExecuteNonQuery()
```

```
End Using
```

' Delete user's messages

```
Dim deleteMessagesQuery As String = "DELETE FROM MESSAGES WHERE  
SENDER_ID = @UserId OR RECEIVER_ID = @UserId"
```

```
Using deleteMessagesCmd As New SqlCommand(deleteMessagesQuery, conn,  
transaction)
```

```
    deleteMessagesCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deleteMessagesCmd.ExecuteNonQuery()
```

```
End Using
```

' Delete user's group messages

```
Dim deleteGroupMessagesQuery As String = "DELETE FROM GROUP_MESSAGES  
WHERE SENDER_ID = @UserId"
```

```
Using deleteGroupMessagesCmd As New  
SqlCommand(deleteGroupMessagesQuery, conn, transaction)
```

```
    deleteGroupMessagesCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deleteGroupMessagesCmd.ExecuteNonQuery()
```

```
End Using
```

' Delete user's posts (and related likes/comments through foreign key constraints)

```
Dim deletePostsQuery As String = "DELETE FROM POSTS WHERE USER_ID =  
@UserId"
```

```
Using deletePostsCmd As New SqlCommand(deletePostsQuery, conn, transaction)
```

```
    deletePostsCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deletePostsCmd.ExecuteNonQuery()
```

```
End Using
```

' Finally, delete the user

```
Dim deleteUserQuery As String = "DELETE FROM USERS WHERE ID = @UserId"
```

```
Using deleteUserCmd As New SqlCommand(deleteUserQuery, conn, transaction)
```

```
deleteUserCmd.Parameters.AddWithValue("@UserId", userId)
```

```
deleteUserCmd.ExecuteNonQuery()
```

```
End Using
```

```
' Commit the transaction
```

```
transaction.Commit()
```

```
lblMessage.Text = "User deleted successfully!"
```

```
lblMessage.CssClass = "message success-message"
```

```
lblMessage.Visible = True
```

```
' Reload the users
```

```
LoadUsers()
```

```
Catch ex As Exception
```

```
' Rollback the transaction if anything fails
```

```
transaction.Rollback()
```

```
Throw
```

```
End Try
```

```
Catch ex As Exception
```

```
lblMessage.Text = "Error deleting user: " & ex.Message
```

```
lblMessage.CssClass = "message error-message"
```

```
lblMessage.Visible = True
```

```
Finally
```

```
conn.Close()
```

```
End Try
```

End Using

End Sub

Private Function GetTimeAgo(dateTime As DateTime) As String

Dim timeSpan As TimeSpan = DateTime.Now - dateTime

If timeSpan.TotalMinutes < 1 Then

Return "Just now"

ElseIf timeSpan.TotalMinutes < 60 Then

Return String.Format("{0} minute{1} ago", Math.Floor(timeSpan.TotalMinutes),
If(Math.Floor(timeSpan.TotalMinutes) = 1, "", "s"))

ElseIf timeSpan.TotalHours < 24 Then

Return String.Format("{0} hour{1} ago", Math.Floor(timeSpan.TotalHours),
If(Math.Floor(timeSpan.TotalHours) = 1, "", "s"))

ElseIf timeSpan.TotalDays < 7 Then

Return String.Format("{0} day{1} ago", Math.Floor(timeSpan.TotalDays),
If(Math.Floor(timeSpan.TotalDays) = 1, "", "s"))

Else

Return dateTime.ToString("MMM dd, yyyy")

End If

End Function

Protected Sub InkLogout_Click(sender As Object, e As EventArgs) Handles InkLogout.Click

' Allowing logout without posting - as per requirement

Session.Clear()

Session.Abandon()

FormsAuthentication.SignOut()

Response.Redirect("Login.aspx")

End Sub

End Class

AdminPanel.aspx:

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

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>SocialMediaPlatform - Admin Panel</title>
  <style>
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background-color: #f0f2f5;
      margin: 0;
      padding: 0;
    }

    .header {
      background-color: #1877f2;
      color: white;
      padding: 8px 16px;
      display: flex;
      justify-content: space-between;
      align-items: center;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
      position: sticky;
      top: 0;
      z-index: 100;
    }

    .logo {
      font-size: 24px;
      font-weight: bold;
    }

    .nav-links a {
      color: white;
      text-decoration: none;
      margin-left: 20px;
```

```
padding: 5px 10px;
border-radius: 4px;
}

.nav-links a:hover {
  background-color: rgba(255, 255, 255, 0.1);
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}

.section-title {
  font-size: 24px;
  margin-bottom: 20px;
  color: #1c1e21;
}

.admin-section {
  background-color: white;
  border-radius: 8px;
  padding: 15px;
  margin-bottom: 20px;
  box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}

.admin-title {
  font-size: 18px;
  font-weight: bold;
  margin-bottom: 15px;
  color: #1c1e21;
}

.user-list, .post-list, .comment-list {
  margin-top: 15px;
}

.user-item, .post-item, .comment-item {
  padding: 10px;
  border: 1px solid #e4e6eb;
  border-radius: 8px;
  margin-bottom: 10px;
}
```

```
display: flex;
align-items: center;
justify-content: space-between;
}
```

```
.user-info, .post-info, .comment-info {
display: flex;
align-items: center;
}
```

```
.user-avatar, .post-avatar, .comment-avatar {
width: 36px;
height: 36px;
border-radius: 50%;
background-color: #e4e6eb;
margin-right: 10px;
display: flex;
align-items: center;
justify-content: center;
color: #1877f2;
}
```

```
.user-name, .post-user, .comment-user {
font-weight: 500;
}
```

```
.user-email, .post-content, .comment-content {
font-size: 12px;
color: #65676b;
}
```

```
.delete-button {
padding: 5px 10px;
background-color: #f5245f;
color: white;
border: none;
border-radius: 4px;
cursor: pointer;
font-size: 12px;
}
```

```
.delete-button:hover {
background-color: #e41e4f;
}
```

```

.message {
    padding: 10px;
    border-radius: 4px;
    margin-bottom: 15px;
}

.success-message {
    background-color: #e8f5e9;
    color: #2e7d32;
    border: 1px solid #c8e6c9;
}

.error-message {
    background-color: #ffebee;
    color: #c62828;
    border: 1px solid #ffcdd2;
}
</style>
</head>
<body>
<form id="form1" runat="server">
    <div class="header">
        <div class="logo">SocialSync - Admin Panel</div>
        <div class="nav-links">
            <asp:HyperLink ID="lnkHome" runat="server"
NavigateUrl="Home.aspx">Home</asp:HyperLink>

            <asp:LinkButton ID="lnkLogout" runat="server"
OnClick="LnkLogout_Click">Logout</asp:LinkButton>
        </div>
    </div>

    <div class="container">
        <asp:Label ID="lblMessage" runat="server" CssClass="message success-message"
Visible="false"></asp:Label>

        <h2 class="section-title">Admin Panel</h2>

        <!-- Manage Users -->
        <div class="admin-section">
            <h3 class="admin-title">Manage Users</h3>
            <div class="user-list">

```

```

        <asp:Repeater ID="rptUsers" runat="server"
OnItemCommand="RptUsers_ItemCommand">
        <ItemTemplate>
        <div class="user-item">
        <div class="user-info">
        <div class="user-avatar"><%# Eval("Username").ToString().Substring(0,
1).ToUpper() %></div>
        <div>
        <div class="user-name"><%# Eval("Username") %></div>
        <div class="user-email"><%# Eval("Email") %></div>
        </div>
        </div>
        <asp:Button ID="btnDeleteUser" runat="server" Text="Delete User"
        CssClass="delete-button"
        CommandName="DeleteUser"
        CommandArgument='<%# Eval("UserID") %>'
        OnClientClick="return confirm('Are you sure you want to delete this
user? This action cannot be undone.');" />
        </div>
        </ItemTemplate>
    </asp:Repeater>
    <asp:Panel ID="pnlNoUsers" runat="server" CssClass="post" Visible="false">
        No users found.
    </asp:Panel>
</div>
</div>

```

```

<!-- Manage Posts -->
<div class="admin-section">
    <h3 class="admin-title">Manage Posts</h3>
    <div class="post-list">
        <asp:Repeater ID="rptPosts" runat="server"
OnItemCommand="RptPosts_ItemCommand">
        <ItemTemplate>
        <div class="post-item">
        <div class="post-info">
        <div class="post-avatar"><%# Eval("Username").ToString().Substring(0,
1).ToUpper() %></div>
        <div>
        <div class="post-user"><%# Eval("Username") %></div>
        <div class="post-content"><%# Eval("PostContent") %></div>
        </div>
        </div>
        <asp:Button ID="btnDeletePost" runat="server" Text="Delete Post"

```

```

        CssClass="delete-button"
        CommandName="DeletePost"
        CommandArgument='<%# Eval("PostID") %>'
        OnClientClick="return confirm('Are you sure you want to delete this
post? This action cannot be undone.');" />
    </div>
</ItemTemplate>
</asp:Repeater>
<asp:Panel ID="pnlNoPosts" runat="server" CssClass="post" Visible="false">
    No posts found.
</asp:Panel>
</div>
</div>

<!-- Manage Comments -->
<div class="admin-section">
    <h3 class="admin-title">Manage Comments</h3>
    <div class="comment-list">
        <asp:Repeater ID="rptComments" runat="server"
OnItemCommand="RptComments_ItemCommand">
            <ItemTemplate>
                <div class="comment-item">
                    <div class="comment-info">
                        <div class="comment-avatar"><%#
Eval("Username").ToString().Substring(0, 1).ToUpper() %></div>
                        <div>
                            <div class="comment-user"><%# Eval("Username") %></div>
                            <div class="comment-content"><%# Eval("CommentContent") %></div>
                        </div>
                    </div>
                    <asp:Button ID="btnDeleteComment" runat="server" Text="Delete
Comment"

                        CssClass="delete-button"
                        CommandName="DeleteComment"
                        CommandArgument='<%# Eval("CommentID") %>'
                        OnClientClick="return confirm('Are you sure you want to delete this
comment? This action cannot be undone.');" />
                </div>
            </ItemTemplate>
        </asp:Repeater>
        <asp:Panel ID="pnlNoComments" runat="server" CssClass="post" Visible="false">
            No comments found.
        </asp:Panel>
    </div>
</div>

```

```
        </div>
    </div>
</form>
</body>
</html>
```

AdminPanel.aspx.vb:

Imports System.Data

Imports System.Data.SqlClient

Partial Class AdminPanel

Inherits System.Web.UI.Page

' Database connection string

Dim connectionString As String =
 ConfigurationManager.ConnectionStrings("Shahzaib").ConnectionString

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

' Check if user is logged in and is an admin

If Session("UserID") Is Nothing Then

Response.Redirect("Login.aspx")

Return

End If

If Session("Role").ToString() <> "Admin" Then

Response.Redirect("Home.aspx")

Return

End If

If Not IsPostBack Then

LoadUsers()

LoadPosts()

LoadComments()

End If

End Sub

Private Sub LoadUsers()

Using conn As New SqlConnection(connectionString)

Try

conn.Open()

Dim query As String = "SELECT ID AS UserID, USER_NAME AS Username, USER_EMAIL
AS Email " &

"FROM USERS WHERE ID <> @CurrentUserId"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@CurrentUserId",
Convert.ToInt32(Session("UserID")))

Dim adapter As New SqlDataAdapter(cmd)

Dim usersTable As New DataTable()

adapter.Fill(usersTable)

rptUsers.DataSource = usersTable

rptUsers.DataBind()

pnlNoUsers.Visible = (usersTable.Rows.Count = 0)

End Using

Catch ex As Exception

lblMessage.Text = "Error loading users: " & ex.Message

```

        lblMessage.CssClass = "message error-message"

        lblMessage.Visible = True

    Finally

        conn.Close()

    End Try

End Using

End Sub

Private Sub LoadPosts()

    Using conn As New SqlConnection(connectionString)

        Try

            conn.Open()

            Dim query As String = "SELECT P.POST_ID AS PostID, P.POST_CONTENT AS
PostContent, U.USER_NAME AS Username " &

                "FROM POSTS P INNER JOIN USERS U ON P.USER_ID = U.ID " &

                "ORDER BY P.CREATED_AT DESC"

            Using cmd As New SqlCommand(query, conn)

                Dim adapter As New SqlDataAdapter(cmd)

                Dim postsTable As New DataTable()

                adapter.Fill(postsTable)

                rptPosts.DataSource = postsTable

                rptPosts.DataBind()

                pnlNoPosts.Visible = (postsTable.Rows.Count = 0)

            End Using

        Catch ex As Exception

```

```
lblMessage.Text = "Error loading posts: " & ex.Message
```

```
lblMessage.CssClass = "message error-message"
```

```
lblMessage.Visible = True
```

```
Finally
```

```
    conn.Close()
```

```
End Try
```

```
End Using
```

```
End Sub
```

```
Private Sub LoadComments()
```

```
    Using conn As New SqlConnection(connectionString)
```

```
        Try
```

```
            conn.Open()
```

```
            Dim query As String = "SELECT C.COMMENT_ID AS CommentID,  
C.COMMENT_CONTENT AS CommentContent, U.USER_NAME AS Username " &  
                "FROM COMMENTS C INNER JOIN USERS U ON C.USER_ID = U.ID " &  
                "ORDER BY C.CREATED_AT DESC"
```

```
            Using cmd As New SqlCommand(query, conn)
```

```
                Dim adapter As New SqlDataAdapter(cmd)
```

```
                Dim commentsTable As New DataTable()
```

```
                adapter.Fill(commentsTable)
```

```
                rptComments.DataSource = commentsTable
```

```
                rptComments.DataBind()
```

```
                pnlNoComments.Visible = (commentsTable.Rows.Count = 0)
```

```
            End Using
```

Catch ex As Exception

lblMessage.Text = "Error loading comments: " & ex.Message

lblMessage.CssClass = "message error-message"

lblMessage.Visible = True

Finally

conn.Close()

End Try

End Using

End Sub

Protected Sub RptUsers_ItemCommand(source As Object, e As RepeaterCommandEventArgs)
Handles rptUsers.ItemCommand

If e.CommandName = "DeleteUser" Then

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

DeleteUser(userId)

End If

End Sub

Protected Sub RptPosts_ItemCommand(source As Object, e As RepeaterCommandEventArgs)
Handles rptPosts.ItemCommand

If e.CommandName = "DeletePost" Then

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

DeletePost(postId)

End If

End Sub

Protected Sub RptComments_ItemCommand(source As Object, e As
RepeaterCommandEventArgs) Handles rptComments.ItemCommand

```
If e.CommandName = "DeleteComment" Then
    Dim commentId As Integer = Convert.ToInt32(e.CommandArgument)
    DeleteComment(commentId)
End If
End Sub
```

```
Private Sub DeleteUser(userId As Integer)
    Using conn As New SqlConnection(connectionString)
        Try
            conn.Open()

            Dim transaction As SqlTransaction = conn.BeginTransaction()

            Try
                ' Delete user's comments

                Dim deleteCommentsQuery As String = "DELETE FROM COMMENTS WHERE USER_ID
= @UserId"

                Using deleteCommentsCmd As New SqlCommand(deleteCommentsQuery, conn,
transaction)

                    deleteCommentsCmd.Parameters.AddWithValue("@UserId", userId)

                    deleteCommentsCmd.ExecuteNonQuery()

                End Using

                ' Delete user's likes

                Dim deleteLikesQuery As String = "DELETE FROM LIKES WHERE USER_ID = @UserId"

                Using deleteLikesCmd As New SqlCommand(deleteLikesQuery, conn, transaction)

                    deleteLikesCmd.Parameters.AddWithValue("@UserId", userId)

                    deleteLikesCmd.ExecuteNonQuery()

                End Using
            End Try
        End Try
    End Using
End Sub
```

' Delete user's followers/following relationships

```
Dim deleteFollowingQuery As String = "DELETE FROM FOLLOWING_T WHERE  
FOLLOWER_ID = @UserId OR FOLLOWING_ID = @UserId"
```

```
Using deleteFollowingCmd As New SqlCommand(deleteFollowingQuery, conn,  
transaction)
```

```
    deleteFollowingCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deleteFollowingCmd.ExecuteNonQuery()
```

```
End Using
```

' Delete user's group memberships

```
Dim deleteGroupMembersQuery As String = "DELETE FROM GROUP_MEMBERS  
WHERE USER_ID = @UserId"
```

```
Using deleteGroupMembersCmd As New SqlCommand(deleteGroupMembersQuery,  
conn, transaction)
```

```
    deleteGroupMembersCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deleteGroupMembersCmd.ExecuteNonQuery()
```

```
End Using
```

' Delete user's messages

```
Dim deleteMessagesQuery As String = "DELETE FROM MESSAGES WHERE  
SENDER_ID = @UserId OR RECEIVER_ID = @UserId"
```

```
Using deleteMessagesCmd As New SqlCommand(deleteMessagesQuery, conn,  
transaction)
```

```
    deleteMessagesCmd.Parameters.AddWithValue("@UserId", userId)
```

```
    deleteMessagesCmd.ExecuteNonQuery()
```

```
End Using
```

' Delete user's group messages

```
Dim deleteGroupMessagesQuery As String = "DELETE FROM GROUP_MESSAGES  
WHERE SENDER_ID = @UserId"
```

```
Using deleteGroupMessagesCmd As New  
SqlCommand(deleteGroupMessagesQuery, conn, transaction)  
    deleteGroupMessagesCmd.Parameters.AddWithValue("@UserId", userId)  
    deleteGroupMessagesCmd.ExecuteNonQuery()  
End Using
```

' Delete user's posts (and related likes/comments through foreign key constraints)

```
Dim deletePostsQuery As String = "DELETE FROM POSTS WHERE USER_ID =  
@UserId"
```

```
Using deletePostsCmd As New SqlCommand(deletePostsQuery, conn, transaction)  
    deletePostsCmd.Parameters.AddWithValue("@UserId", userId)  
    deletePostsCmd.ExecuteNonQuery()  
End Using
```

' Finally, delete the user

```
Dim deleteUserQuery As String = "DELETE FROM USERS WHERE ID = @UserId"  
Using deleteUserCmd As New SqlCommand(deleteUserQuery, conn, transaction)  
    deleteUserCmd.Parameters.AddWithValue("@UserId", userId)  
    deleteUserCmd.ExecuteNonQuery()  
End Using
```

```
transaction.Commit()
```

```
lblMessage.Text = "User deleted successfully!"
```

```
lblMessage.CssClass = "message success-message"
```

```
lblMessage.Visible = True
```

```

        LoadUsers()
    Catch ex As Exception
        transaction.Rollback()
        Throw
    End Try
Catch ex As Exception
    lblMessage.Text = "Error deleting user: " & ex.Message
    lblMessage.CssClass = "message error-message"
    lblMessage.Visible = True
Finally
    conn.Close()
End Try
End Using
End Sub

Private Sub DeletePost(postId As Integer)
    Using conn As New SqlConnection(connectionString)
        Try
            conn.Open()

            Dim transaction As SqlTransaction = conn.BeginTransaction()
            Try
                ' Delete associated comments

                Dim deleteCommentsQuery As String = "DELETE FROM COMMENTS WHERE POST_ID
= @PostId"

                Using deleteCommentsCmd As New SqlCommand(deleteCommentsQuery, conn,
transaction)
                    deleteCommentsCmd.Parameters.AddWithValue("@PostId", postId)

```

```
deleteCommentsCmd.ExecuteNonQuery()
```

```
End Using
```

```
' Delete associated likes
```

```
Dim deleteLikesQuery As String = "DELETE FROM LIKES WHERE POST_ID = @PostId"
```

```
Using deleteLikesCmd As New SqlCommand(deleteLikesQuery, conn, transaction)
```

```
deleteLikesCmd.Parameters.AddWithValue("@PostId", postId)
```

```
deleteLikesCmd.ExecuteNonQuery()
```

```
End Using
```

```
' Delete the post
```

```
Dim deletePostQuery As String = "DELETE FROM POSTS WHERE POST_ID = @PostId"
```

```
Using deletePostCmd As New SqlCommand(deletePostQuery, conn, transaction)
```

```
deletePostCmd.Parameters.AddWithValue("@PostId", postId)
```

```
deletePostCmd.ExecuteNonQuery()
```

```
End Using
```

```
transaction.Commit()
```

```
lblMessage.Text = "Post deleted successfully!"
```

```
lblMessage.CssClass = "message success-message"
```

```
lblMessage.Visible = True
```

```
LoadPosts()
```

```
Catch ex As Exception
```

```
transaction.Rollback()
```

```
Throw
```

End Try

Catch ex As Exception

lblMessage.Text = "Error deleting post: " & ex.Message

lblMessage.CssClass = "message error-message"

lblMessage.Visible = True

Finally

conn.Close()

End Try

End Using

End Sub

Private Sub DeleteComment(commentId As Integer)

Using conn As New SqlConnection(connectionString)

Try

conn.Open()

Dim query As String = "DELETE FROM COMMENTS WHERE COMMENT_ID =
@CommentId"

Using cmd As New SqlCommand(query, conn)

cmd.Parameters.AddWithValue("@CommentId", commentId)

cmd.ExecuteNonQuery()

End Using

lblMessage.Text = "Comment deleted successfully!"

lblMessage.CssClass = "message success-message"

lblMessage.Visible = True

LoadComments()

Catch ex As Exception

lblMessage.Text = "Error deleting comment: " & ex.Message

lblMessage.CssClass = "message error-message"

lblMessage.Visible = True

Finally

conn.Close()

End Try

End Using

End Sub

Protected Sub lnkLogout_Click(sender As Object, e As EventArgs) Handles lnkLogout.Click

Session.Clear()

Session.Abandon()

FormsAuthentication.SignOut()

Response.Redirect("Login.aspx")

End Sub

End Class

Registration.aspx:

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

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>Register for SocialSync</title>
```

```
<style>
```

```
body {
```

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
background-color: #f0f2f5;
```

```
margin: 0;
```

```
padding: 0;
display: flex;
justify-content: center;
align-items: center;
min-height: 100vh;
padding: 20px 0;
}
```

```
.register-container {
  background-color: white;
  border-radius: 8px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
  padding: 25px;
  width: 450px;
}
```

```
h2 {
  color: #1877f2;
  margin-bottom: 25px;
  text-align: center;
}
```

```
.form-group {
  margin-bottom: 15px;
}
```

```
.form-group label {
  display: block;
  margin-bottom: 5px;
  font-weight: 500;
}
```

```
.form-control {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 5px;
  font-size: 14px;
  box-sizing: border-box;
}
```

```
.btn {
  width: 100%;
  padding: 10px;
```

```
background-color: #42b72a;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
font-weight: bold;
margin-top: 10px;
}

.btn:hover {
background-color: #36a420;
}

.error-message {
color: #c62828;
margin-top: 5px;
font-size: 13px;
}

.success-message {
color: #2e7d32;
margin-top: 5px;
font-size: 14px;
text-align: center;
}

.login-link {
text-align: center;
margin-top: 20px;
font-size: 14px;
}

.login-link a {
color: #1877f2;
text-decoration: none;
}

.login-link a:hover {
text-decoration: underline;
}
</style>
</head>
<body>
```

```

<form id="form1" runat="server">
  <div class="register-container">
    <h2>Create a SocialSync Account</h2>

    <div class="form-group">
      <label for="txtEmail">Email</label>
      <asp:TextBox ID="txtEmail" runat="server" CssClass="form-control"
placeholder="Enter your email"></asp:TextBox>
      <asp:RequiredFieldValidator ID="rfvEmail" runat="server"
ControlToValidate="txtEmail"
      ErrorMessage="Email is required" CssClass="error-message"
Display="Dynamic"></asp:RequiredFieldValidator>
      <asp:RegularExpressionValidator ID="revEmail" runat="server"
ControlToValidate="txtEmail"
      ErrorMessage="Invalid email format" CssClass="error-message" Display="Dynamic"
      ValidationExpression="^[\\w-\\.]+@([\\w-]+\\.)+[\\w-
]{2,4}$"></asp:RegularExpressionValidator>
    </div>

    <div class="form-group">
      <label for="txtFullName">Full Name</label>
      <asp:TextBox ID="txtFullName" runat="server" CssClass="form-control"
placeholder="Enter your full name"></asp:TextBox>
      <asp:RequiredFieldValidator ID="rfvFullName" runat="server"
ControlToValidate="txtFullName"
      ErrorMessage="Full name is required" CssClass="error-message"
Display="Dynamic"></asp:RequiredFieldValidator>
    </div>

    <div class="form-group">
      <label for="txtUsername">Username</label>
      <asp:TextBox ID="txtUsername" runat="server" CssClass="form-control"
placeholder="Choose a username"></asp:TextBox>
      <asp:RequiredFieldValidator ID="rfvUsername" runat="server"
ControlToValidate="txtUsername"
      ErrorMessage="Username is required" CssClass="error-message"
Display="Dynamic"></asp:RequiredFieldValidator>
    </div>

    <div class="form-group">
      <label for="txtPassword">Password</label>
      <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
CssClass="form-control" placeholder="Create a password"></asp:TextBox>

```

```

        <asp:RequiredFieldValidator ID="rfvPassword" runat="server"
ControlToValidate="txtPassword"
        ErrorMessage="Password is required" CssClass="error-message"
Display="Dynamic"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator ID="revPassword" runat="server"
ControlToValidate="txtPassword"
        ErrorMessage="Password must be at least 8 characters" CssClass="error-message"
Display="Dynamic"
        ValidationExpression=".{8,}"></asp:RegularExpressionValidator>
    </div>

    <div class="form-group">
        <label for="txtBirthDate">Birth Date</label>
        <asp:TextBox ID="txtBirthDate" runat="server" TextMode="Date" CssClass="form-
control"></asp:TextBox>
        <asp:RequiredFieldValidator ID="rfvBirthDate" runat="server"
ControlToValidate="txtBirthDate"
        ErrorMessage="Birth date is required" CssClass="error-message"
Display="Dynamic"></asp:RequiredFieldValidator>
    </div>

    <asp:Button ID="btnRegister" runat="server" Text="Sign Up" CssClass="btn"
OnClick="btnRegister_Click" />

    <asp:Label ID="lblMessage" runat="server" CssClass="success-message"></asp:Label>

    <div class="login-link">
        <p>Already have an account? <asp:HyperLink ID="lnkLogin" runat="server"
NavigateUrl="Login.aspx">Login here</asp:HyperLink></p>
    </div>
</div>
</form>
</body>
</html>

```

Registration.aspx.vb:

```

Imports System.Data.SqlClient
Imports System.Text.RegularExpressions

```

```

Partial Class Register
    Inherits System.Web.UI.Page

```

```
' Database connection string
Dim connectionString As String =
ConfigurationManager.ConnectionStrings("Shahzaib").ConnectionString
```

```
Protected Sub btnRegister_Click(sender As Object, e As EventArgs) Handles btnRegister.Click
    Dim email As String = txtEmail.Text.Trim()
    Dim fullName As String = txtFullName.Text.Trim()
    Dim username As String = txtUsername.Text.Trim()
    Dim password As String = txtPassword.Text.Trim()
    Dim birthDate As String = txtBirthDate.Text.Trim()
```

```
' Input validation
If Not IsValidEmail(email) Then
    lblMessage.Text = "Invalid email format."
    lblMessage.CssClass = "error-message"
    Return
End If
```

```
If fullName.Length < 2 OrElse fullName.Length > 100 Then
    lblMessage.Text = "Full name must be between 2 and 100 characters."
    lblMessage.CssClass = "error-message"
    Return
End If
```

```
If username.Length < 3 OrElse username.Length > 50 Then
    lblMessage.Text = "Username must be between 3 and 50 characters."
    lblMessage.CssClass = "error-message"
    Return
End If
```

```
If password.Length < 8 Then
    lblMessage.Text = "Password must be at least 8 characters."
    lblMessage.CssClass = "error-message"
    Return
End If
```

```
If Not IsValidDate(birthDate) Then
    lblMessage.Text = "Invalid birth date."
    lblMessage.CssClass = "error-message"
    Return
End If
```

```
Using conn As New SqlConnection(connectionString)
```

```

Try
    conn.Open()
    ' Check if email already exists
    Dim checkEmailQuery As String = "SELECT COUNT(*) FROM USERS WHERE
USER_EMAIL = @Email"
    Using checkCmd As New SqlCommand(checkEmailQuery, conn)
        checkCmd.Parameters.AddWithValue("@Email", email)
        If Convert.ToInt32(checkCmd.ExecuteScalar()) > 0 Then
            lblMessage.Text = "Email already exists."
            lblMessage.CssClass = "error-message"
            Return
        End If
    End Using

    ' Check if username already exists
    Dim checkUsernameQuery As String = "SELECT COUNT(*) FROM USERS WHERE
USERNAME = @Username"
    Using checkCmd As New SqlCommand(checkUsernameQuery, conn)
        checkCmd.Parameters.AddWithValue("@Username", username)
        If Convert.ToInt32(checkCmd.ExecuteScalar()) > 0 Then
            lblMessage.Text = "Username already exists."
            lblMessage.CssClass = "error-message"
            Return
        End If
    End Using

    ' Get next available ID
    Dim idQuery As String = "SELECT ISNULL(MAX(ID), 0) + 1 FROM USERS"
    Using idCmd As New SqlCommand(idQuery, conn)
        Dim newId As Integer = Convert.ToInt32(idCmd.ExecuteScalar())
        ' Store password as-is for demo - in real app use password hashing

        Dim insertQuery As String = "INSERT INTO USERS (ID, USER_EMAIL, USER_NAME,
USERNAME, USER_PASSWORD, BIRTH_DATE, CREATED_AT) " &
        "VALUES (@ID, @Email, @FullName, @Username, @Password,
@BirthDate, GETDATE())"
        Using insertCmd As New SqlCommand(insertQuery, conn)
            insertCmd.Parameters.AddWithValue("@ID", newId)
            insertCmd.Parameters.AddWithValue("@Email", email)
            insertCmd.Parameters.AddWithValue("@FullName", fullName)
            insertCmd.Parameters.AddWithValue("@Username", username)
            insertCmd.Parameters.AddWithValue("@Password", password) ' Should use
proper hashing in real app
            insertCmd.Parameters.AddWithValue("@BirthDate", birthDate)

```

```

        insertCmd.ExecuteNonQuery()
    End Using
End Using

lblMessage.Text = "Registration successful! Redirecting to login..."
lblMessage.CssClass = "success-message"
' Redirect to login page after 2 seconds
Response.AddHeader("REFRESH", "2;URL=Login.aspx")
Catch ex As Exception
    lblMessage.Text = "Error: " & ex.Message
    lblMessage.CssClass = "error-message"
Finally
    conn.Close()
End Try
End Using
End Sub

Private Function IsValidEmail(email As String) As Boolean
    Return Regex.IsMatch(email, "^[\\w-\\.]+@([\\w-]+\\.){1,4}$")
End Function

Private Function IsValidDate(dateStr As String) As Boolean
    Try
        Dim dt As DateTime = DateTime.Parse(dateStr)
        Return dt <= DateTime.Now
    Catch
        Return False
    End Try
End Function
End Class

```