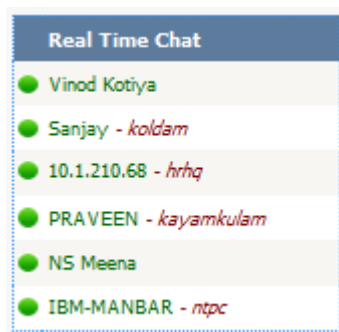


Gmail/Facebook Style AJAX Instant Messenger Module

Introduction

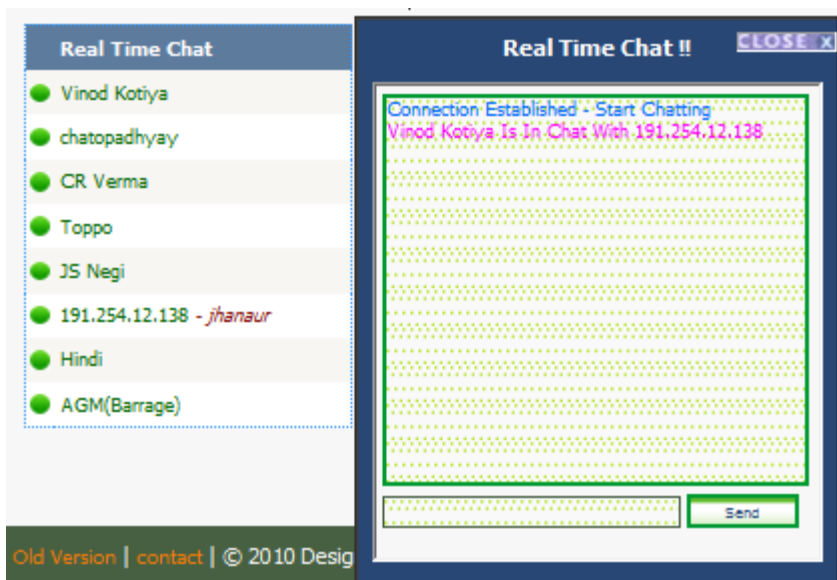
We've developed a simple IM web application for LPHPP intranet <http://191.254.186.1> that deals with sql express 2008 to store information about online users, using ASP.NET, VB, SQL Server, Ajax on .NET Framework.

It is one of the most innovative and easy-to-use IM web service designed for web sites and intranets. It provides an easy and inexpensive way to enhance the web communication. Everyone loves the gmail and facebook inline IM modules. This ajax based IM module enables you to seamlessly integrate Gmail/Facebook style IM into your existing website.



Features

1. Gmail style modal pop up display of IM boxes.
2. Auto login through users ipaddress



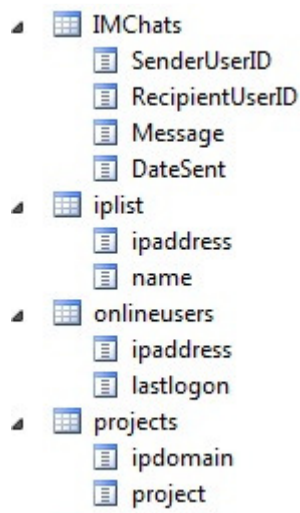
3. Displays IP address when name is not in database.
4. Displays "X says..." & start sound(currently disabled) when IM boxes open.
5. close IM boxes
6. Keeps IM boxes open on any pages of intranet.
7. Auto-scrolling of IM text
8. Auto-back-off polling policy (hits the server less-often when IM activity is low)
9. Extremely simple to integrate into existing site

Demo

Please open the following url <http://191.254.186.1>

(the ipaddress is actually not supposed to be used by multiple users at the same time)

Database: You must first create few SQL Express table as below



1. IMChats : It contain temporary IM data between sender and recipient and deleted during closing / initiating IM session.
2. Iplist : contains username with ipaddress. Have more then 5000 address of allover ntpc.
3. Onlineusers : it contains list of online users. Updates every 10 seconds and sort them according to lastlogon.
4. Projects : contains initial 3 octets of ipaddress to get ntpc project name. This database has been disabled cause to decrease load on sqlserver instead a javascript has been used to complete that task since database was small.

Browser Compatibility

1. Firefox 2+
2. Internet Explorer 6+
3. Safari 2+
5. Opera 9+
6. Google chrome

How to code (short):

The script does manage rosters (i.e. which users are online etc.), it enables users to IM with each other. I assume your intranet may provide that functionality for user tracking. You must also make sure that SESSION['myIP'] is being set with client ip address when your website session begins. You will understand the logic better after you try the sample files that we have provided at <http://191.254.186.1/source.html>

- (A) Get a placeholder for onlineusers in ur aspx page. (We have done only in home page)
- (b) Set variables in page load event and create a listener event using javascript for incoming IM request.
(It has been done in all web pages)
- (c) using Ajax timer control we will refresh online users.
- (d) A web service is created named **chatservice.asmx** using chatservice.vb in App_code folder

```
'ENQ = chr(5)           # Request to send
'EOT = chr(4)           # Ready to receive
'ACK = chr(6)           # Correct reception
'NAK = chr(21)          # Incorrect reception
'////////////////////////
'// Sender init chat
'////////////////////////
'// Sender ENQ      Wait 4 ACK
'// Recip  ACK ENQ Wait 4 ACK ACK
'// Sender ACK ACK Wait 4 EOT
'// Recip  EOT
'////////////////////////
'// Recip wait 4 chat
'////////////////////////
'// Recip  ACK ENQ Wait 4 ACK
'// Sender ACK ACK Wait 4 EOT
'// Recip  EOT
```

- (e) Getting Project list

How to code (Detail):

- A. Get a placeholder for onlineusers in ur aspx page.
A javascript inside head tag

```
1. <script type="text/javascript" runat="server">
2.     Function getProject(ByVal ipaddress As String) As String
3.         Dim oct3 As String = Strings.Left(ipaddress, Strings.InStrRev(ipaddress,
4.         ".") - 1)
5.         Select Case oct3
6.             Case "191.254.186", "127.0.0"
7.                 oct3 = ""
8.             Case "191.254.20"
9.                 oct3 = " - ANTA"
10.            Case "191.254.16"
```

```

10.         oct3 = " - AURAIYA"
11.     Case "191.254.178"
12.         oct3 = " - BADARPUR"
13.     Case "191.254.100"
14.         oct3 = " - BARH"
15.     Case "10.1.249"
16.         oct3 = " - BONGAIGAON"
17.     Case "10.0.9"
18.         oct3 = " - CC"
19.     Case "191.254.157"
20.         oct3 = " - DADRI"
21.     Case "190.190.25"
22.         oct3 = " - EOC"
23.     Case "191.254.171"
24.         oct3 = " - ERHQ I"
25.     Case "10.1.240"
26.         oct3 = " - ERHQ II"
27. Case Else
28.     oct3 = "- ntpc"
29. End Select
30. Return oct3.ToLower()
31. End Function
32. </script>

33. <asp:UpdatePanel ID="UpdatePanel2" runat="server">
34.     <ContentTemplate>
35.         <div style="width: 139px; height: 197px; overflow-x: hidden;
overflow-y: auto; white-space: nowrap;" >
36.             <asp:GridView ID="gvUsers" runat="server" AutoGenerateColumns="False"
37.                 CellPadding="2" ForeColor="#333333"
38.                 GridLines="None" Style="position: static"
39.                 EmptyDataText="No Users Online" Font-Size="9px" Width="137px"
BorderColor="#3399FF"
40.                 BorderStyle="Dotted" BorderWidth="1px" >
41.                 <FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White" />
42.                 <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
43.                 <Columns>
44.
45.                     <asp:TemplateField ShowHeader="False">
46.                         <ItemTemplate>
47.                             <asp:Image ID="Image1" ImageUrl='<%# "images/greendot.gif" %>'
runat="server" width="8px" height="8px" />
48.                         </ItemTemplate>
49.                         <HeaderStyle HorizontalAlign="Center" />
50.                         <ItemStyle HorizontalAlign="Left" Width="8px" />
51.                     </asp:TemplateField>
52.
53.                     <asp:TemplateField HeaderText="Real Time Chat" >
54.                         <ItemTemplate>
55.                             <a href="javascript:vinModal('<%# Eval("ipaddress") %>');" ><%#
Eval("name")%> <font color="maroon"><i> <%# getProject(Eval("ipaddress"))%></i>
</font> </a>
56.                         </ItemTemplate>
57.                         <HeaderStyle HorizontalAlign="Left" />
58.                         <ItemStyle HorizontalAlign="Left" />
59.                     </asp:TemplateField>

```

```

60.         </Columns>
61.         <PagerStyle BackColor="#284775" ForeColor="White"
HorizontalAlign="Center" />
62.         <EmptyDataTemplate>
63.             No users are online now.
64.         </EmptyDataTemplate>
65.         <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True"
ForeColor="#333333" />
66.         <HeaderStyle BackColor="#5D7B9D" Font-Bold="True"
ForeColor="White" />
67.         <EditRowStyle BackColor="#999999" />
68.         <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
69.     </asp:GridView>
70. </div>
71.     <asp:Timer ID="Timer1" runat="server" ontick="Timer1_Tick"
Interval="10000">
72.
73.     </asp:Timer>
74.
75.     </ContentTemplate>
76. </asp:UpdatePanel>
77.
78. <asp:ModalPopupExtender ID="ModalPopupExtender2" runat="server"
TargetControlID="net" PopupControlID="upNet" BackgroundCssClass="modalBackground"
dropshadow="true" OkControlID="btnok2" ></asp:ModalPopupExtender>
79.
80.
81.
82.     <div id="chatsection" >
83.         <asp:Panel ID="PanelChatRequest" runat="server" Style="display: none"
CssClass="modalheader" scrolling="no" BorderWidth="2px" BorderStyle="Outset"
BorderColor="Black">
84.             <center class="modalcontent"> <div id="ChatRequest">
85.                 </div>
86.                 <asp:Label ID="Label1" runat="server" Text="Real Time Chat !!"
ForeColor="White"></asp:Label>
87.                 <asp:ImageButton ID="CancelButton" runat="server"
ImageAlign="Right" CssClass="closemodal"/>
88.                 <p style="text-align: center;">
89.                     <iframe id="chatframe" width="188" height="200"
scrolling="no"></iframe>
90.                 </p>
91.             </center>
92.         </asp:Panel>
93.         <asp:Button runat="server" ID="hiddenTargetControlForModalPopup"
style="display:none"/>
94.         <asp:TextBox ID="hiddenSenderUserID" runat="server"
style="display:none"></asp:TextBox>
95.         <asp:TextBox ID="hiddenMyUserID" runat="server"
style="display:none"></asp:TextBox>
96.         <asp:TextBox ID="hiddenAppPath" runat="server"
style="display:none"></asp:TextBox>
97.         <asp:ModalPopupExtender runat="server" ID="programmaticModalPopup"
BehaviorID="programmaticModalPopupBehaviorChatReq"
TargetControlID="hiddenTargetControlForModalPopup"
PopupControlID="PanelChatRequest"
RepositionMode="RepositionOnWindowResizeAndScroll"
CancelControlID="CancelButton"

```

```

103.         OnCancelScript="OnCancel()"
104.         PopupDragHandleControlID = "PanelChatRequest" >
105.     </asp:ModalPopupExtender>
106.</div>

```

B. Set variables in page load event and create a listener event for incoming chat request

```

1. Dim ChatService As New ChatService
2.     Session("myIP") = Request.UserHostAddress ' for LAN/WAN or
3.     'Session("myIP") = Request.UserHostAddress + Request.Browser.Browser
   'for single PC
4.     'Session("myIP") = Request.UserHostAddress + Str(Now.TimeOfDay.Seconds
   * Rnd(30))
5.     If Not Page.IsPostBack Then
6.         'this will be executed only first time any event happen on web page
7.
8.         ChatService.trackOnlineVisitor(Session("myIP"))
9.
10.        ' Clear user paging/sort
11.
12.
13.        Session("CurrentPage") = 0
14.        Session("SortExpression") = Nothing
15.        GridViewSortDirection = "ASC"
16.        ' Set current query
17.        Session("CurrentQuery") = "SELECT LEFT(sub, 35) + '...' AS
   sub,filename, convert(varchar, cdate, 104) as cdate1, cdate from circulars
   ORDER BY cdate DESC"
18.        Session("CurrentQuery1") = "SELECT LEFT(topic, 35) + '.. 'as
   topic, filename from articles ORDER BY cdate DESC"
19.        '' setting to show popup should be made in web.config file : inside
   <configuration> <appSettings>
20.        If
   System.Configuration.ConfigurationManager.AppSettings("showPopup") = "1" Then
21.            ModalPopupExtender1.Show()
22.        End If
23.
24.        If Not String.IsNullOrEmpty(Session("eid")) Then
25.            lblName.Text = "Welcome <font color='#9218AB'>" +
   Session("name") + "</font><br/> Your Visits : " +
   (trackVisitor(Request.UserHostAddress))
26.        Else
27.            lblName.Text = "Welcome Guest: <font color='#9218AB'>" +
   Request.UserHostAddress + "</font><br/> Your Visits : " +
   (trackVisitor(Request.UserHostAddress)) '
   HttpContext.Current.User.Identity.Name +
28.        End If
29.        lblQuotes.Text = getQuotes()
30.
31.
32.
33.
34.    End If
35.    Dim oQuery = " SELECT          o.ipaddress as ipaddress, o.lastlogon,
   ISNULL(i.name, o.ipaddress) AS name " & _
36.    "FROM          onlineusers AS o LEFT OUTER JOIN " & _
37.    "iplist AS i ON o.ipaddress = i.ipaddress " & _

```

```

38. "ORDER BY o.lastlogon DESC"
39.         ChatService.rebindGridView2(oQuery, gvUsers)
40.
41.         'this will be executed every time any event happen on web page
42.
43.         ' Bind GridView to current query & always store ur query into
session("currentquery") before calling
44.         ' reason is whenever page indexed changed or sort.. then it will show
data from currentquery
45.         'Response.Write(loadwishes())
46.
47.         If (Not ClientScript.IsStartupScriptRegistered(Me.GetType(),
"vinscript")) Then
48.             ClientScript.RegisterClientScriptBlock(Me.GetType(), "vinscript",
loadwishes())
49.         End If
50.
51.
52.         ' Request.Browser.Browser
53.         ' Session("myIP") = Request.UserHostAddress '+'
Str(Now.TimeOfDay.Seconds * Rnd(30))
54.
55.         ' End If
56.         If Not String.IsNullOrEmpty(Session("myIP")) Then ' If Not
String.IsNullOrEmpty(Page.User.Identity.Name) Then
57.             hiddenAppPath.Text = Request.ServerVariables("SERVER_NAME") & ":" &
Request.ServerVariables("SERVER_PORT") & Request.ApplicationPath()
58.
59.             Dim csname As String = "CheckChatReq"
60.             Dim cstype As Type = Me.GetType()
61.             Dim cs As ClientScriptManager = Page.ClientScript
62.             If (Not cs.IsClientScriptBlockRegistered(cstype, csname)) Then
63.                 Dim cstext As New StringBuilder()
64.                 cstext.Append("var DelayInSeconds = 12;" & vbCrLf)
65.                 cstext.Append("function InitCheckChatReq(strMyUserID) {" &
vbCrLf)
66.                 cstext.Append("    setTimeout('\"CheckChatReq('\" + strMyUserID
+ \"')\"\", DelayInSeconds * 1000);" & vbCrLf)
67.                 cstext.Append("}" & vbCrLf)
68.                 cstext.Append("function CheckChatReq(strMyUserID) {" & vbCrLf)
69.                 cstext.Append("    ret = ChatService.CheckChatReq(strMyUserID,
CheckChatReqOnComplete, CheckChatReqOnTimeout);" & vbCrLf)
70.                 cstext.Append("    document.getElementById('\" &
hiddenMyUserID.ClientID & '\").value = strMyUserID;" & vbCrLf)
71.                 cstext.Append("}" & vbCrLf)
72.                 cstext.Append("function CheckChatReqOnComplete(strSenderUserID)
{" & vbCrLf)
73.                 cstext.Append("    if (strSenderUserID != \"\")" & vbCrLf)
74.                 cstext.Append("    {" & vbCrLf)
75.                 cstext.Append("        var popupWindow = null;" & vbCrLf)
76.                 cstext.Append("        document.getElementById('\" &
hiddenSenderUserID.ClientID & '\").value = strSenderUserID;" & vbCrLf)
77.                 cstext.Append("        //
document.getElementById('ChatRequest').innerHTML = \"<p style=\\\"text-align:
center;\\\">\"\" & vbCrLf)
78.                 cstext.Append("        //          + \"You have an incoming chat
request from: \"\" + strSenderUserID + \"\"<br/><br/>\"\" & vbCrLf)

```

```

79.         cstext.Append(" //          + ""Would you like to
accept?</p>"";" & vbCrLf)
80.         cstext.Append("     var strSenderUserID =
document.getElementById('' & hiddenSenderUserID.ClientID & '').value;" &
vbCrLf)
81.         cstext.Append("     var strMyUserID = document.getElementById(''
& hiddenMyUserID.ClientID & '').value;" & vbCrLf)
82.         cstext.Append("     var strAppPath = document.getElementById(''
& hiddenAppPath.ClientID & '').value;" & vbCrLf)
83.         cstext.Append("     $get('chatframe').src='http://' +
strAppPath + '/chat.aspx?Usr=' + strSenderUserID + '&Init=0';" & vbCrLf)
84.         cstext.Append("     var modalPopupBehavior =
$find('programmaticModalPopupBehaviorChatReq');" & vbCrLf)
85.         cstext.Append("     modalPopupBehavior.show();" & vbCrLf)
86.         cstext.Append(" // redirect('http://' + strAppPath +
'/chat.aspx?Usr=' + strSenderUserID + '&Init=0');" & vbCrLf)
87.         cstext.Append(" // window.location = 'http://' + strAppPath +
'/chat.aspx?Usr=' + strSenderUserID + '&Init=0');" & vbCrLf)
88.         cstext.Append(" // popupWindow = window.open('http://' +
strAppPath + '/chat.aspx?Usr=' + strSenderUserID + '&Init=0');" & vbCrLf)
89.         cstext.Append("     setTimeout(''CheckChatReq('' + strMyUserID +
''')'', 1000);" & vbCrLf)
90.         cstext.Append("     }" & vbCrLf)
91.         cstext.Append("     else" & vbCrLf)
92.         cstext.Append("     {" & vbCrLf)
93.         cstext.Append("         var strMyUserID =
document.getElementById('' & hiddenMyUserID.ClientID & '').value;" & vbCrLf)
94.         cstext.Append("         setTimeout(''CheckChatReq('' +
strMyUserID + ''')'', DelayInSeconds * 1000);" & vbCrLf)
95.         cstext.Append("     }" & vbCrLf)
96.         cstext.Append("}" & vbCrLf)
97.         cstext.Append("function vinModal( strOther) {" & vbCrLf) 'this
will call other person window so init =1
98.         cstext.Append("     var popupWindow = null;" & vbCrLf)
99.         cstext.Append("     document.getElementById('' &
hiddenSenderUserID.ClientID & '').value = strOther;" & vbCrLf)
100.        cstext.Append("     var strSenderUserID =
document.getElementById('' & hiddenSenderUserID.ClientID & '').value;" &
vbCrLf)
101.        cstext.Append("     var strMyUserID =
document.getElementById('' & hiddenMyUserID.ClientID & '').value;" & vbCrLf)
102.        cstext.Append("     var strAppPath =
document.getElementById('' & hiddenAppPath.ClientID & '').value;" & vbCrLf)
103.        cstext.Append("     $get('chatframe').src= 'chat.aspx?Usr='
+ strSenderUserID+ '&Init=1';" & vbCrLf)
104.        cstext.Append("     var modalPopupBehavior =
$find('programmaticModalPopupBehaviorChatReq');" & vbCrLf)
105.        cstext.Append("     modalPopupBehavior.show();" & vbCrLf)
106.        cstext.Append(" //movePanel();" & vbCrLf)
107.        cstext.Append("     }" & vbCrLf)
108.        cstext.Append("var id = null;" & vbCrLf)
109.        cstext.Append("function movePanel() { " & vbCrLf)
110.        cstext.Append("     var pnl = $get('PanelChatRequest');" &
vbCrLf)
111.        cstext.Append(" if (pnl != null) { " & vbCrLf)
112.        cstext.Append("     pnl.style.top = '200'; " & vbCrLf)
113.        cstext.Append("     pnl.style.left = '200'; } " & vbCrLf)

```



```

114.                cstext.Append(" id = setTimeout('movePanel();', 100);" &
vbCrLf)
115.                cstext.Append("}" & vbCrLf)
116.                cstext.Append(" function stopMoving() { " & vbCrLf)
117.                cstext.Append(" clearTimeout(id);" & vbCrLf)
118.                cstext.Append("}" & vbCrLf)
119.                cstext.Append("function
CheckChatReqOnTimeout(strSenderUserID) {" & vbCrLf)
120.                cstext.Append("    var strMyUserID =
document.getElementById('" & hiddenMyUserID.ClientID & "').value;" & vbCrLf)
121.                cstext.Append("    setTimeout('\"CheckChatReq('\" +
strMyUserID + \"'\")\", DelayInSeconds * 1000);" & vbCrLf)
122.                cstext.Append("}" & vbCrLf)
123.                cstext.Append("function OnOk() {" & vbCrLf)
124.                cstext.Append("    var strSenderUserID =
document.getElementById('" & hiddenSenderUserID.ClientID & "').value;" &
vbCrLf)
125.                cstext.Append("    var strMyUserID =
document.getElementById('" & hiddenMyUserID.ClientID & "').value;" & vbCrLf)
126.                cstext.Append("    var strAppPath =
document.getElementById('" & hiddenAppPath.ClientID & "').value;" & vbCrLf)
127.                cstext.Append("    // var w = window.open();" & vbCrLf)
128.                cstext.Append("    // w.opener = null;" & vbCrLf)
129.                cstext.Append("    // w.document.location = 'http://' +
strAppPath + '/chat.aspx?Usr=' + strSenderUserID + '&Init=0';" & vbCrLf)
130.                cstext.Append("    // window.location = 'http://' +
strAppPath + '/chat.aspx?Usr=' + strSenderUserID + '&Init=0';" & vbCrLf)
131.                cstext.Append("    // popupWindow = window.open('http://' +
strAppPath + '/chat.aspx?Usr=' + strSenderUserID + '&Init=0');" & vbCrLf)
132.                cstext.Append("    setTimeout('\"CheckChatReq('\" +
strMyUserID + \"'\")\", 1000);" & vbCrLf)
133.                cstext.Append("}" & vbCrLf)
134.                cstext.Append("function OnCancel() {" & vbCrLf)
135.                cstext.Append("    var strSenderUserID =
document.getElementById('" & hiddenSenderUserID.ClientID & "').value;" &
vbCrLf)
136.                cstext.Append("    var strMyUserID =
document.getElementById('" & hiddenMyUserID.ClientID & "').value;" & vbCrLf)
137.                cstext.Append("    ret = ChatService.SendNak(strMyUserID,
strSenderUserID);" & vbCrLf)
138.                cstext.Append("    setTimeout('\"CheckChatReq('\" +
strMyUserID + \"'\")\", 1000);" & vbCrLf)
139.                cstext.Append("//stopMoving();" & vbCrLf)
140.                cstext.Append(" }" & vbCrLf)
141.
142.                cs.RegisterClientScriptBlock(cstype, csname,
cstext.ToString, True)
143.                End If
144.
145.                ' Dim strScript As String = "InitCheckChatReq('" &
HttpUtility.UrlEncode(Page.User.Identity.Name) & "');"
146.                Dim strScript As String = "InitCheckChatReq('" &
Session("myIP") & "');"
147.                csname = "InitCheckChatReq"
148.                cs = Page.ClientScript
149.                If (Not cs.IsStartupScriptRegistered(cstype, csname)) Then
150.                    cs.RegisterStartupScript(cstype, csname, strScript, True)
151.                End If

```

```

152.
153.         End If
154.
155.
156.         'rebindGridView(Session("CurrentQuery"), gvCirculars)
157.         'rebindGridView(Session("CurrentQuery1"), gvArticles)
158.         rebindCachedGridView(Session("CurrentQuery"), gvCirculars,
"gvCirculars")
159.         rebindCachedGridView(Session("CurrentQuery1"), gvArticles,
"gvArticles")
160.

```

C. using Ajax timer control we will refresh online users.

```

1. Protected Sub Timer1_Tick(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
2.     Dim ChatService As New ChatService
3.     ChatService.trackOnlineVisitor(Session("myIP"))
4.     ' rebindGridView("select * from onlineusers", gvUsers)
5.     Dim oQuery = " SELECT          o.ipaddress as ipaddress, o.lastlogon,
ISNULL(i.name, o.ipaddress) AS name " & _
6. "FROM          onlineusers AS o LEFT OUTER JOIN " & _
7. "              "iplist AS i ON o.ipaddress = i.ipaddress " & _
8. "ORDER BY o.lastlogon DESC"
9.     ChatService.rebindGridView2(oQuery, gvUsers)
10. End Sub

```

D. . A web service is created named **chatservice.asmx** using chatservice.vb in App_code folder

```

1. ' ASPDOTNETAJAXIM - Copyright 2010 Vinod Kotiya
2. Imports System.Web
3. Imports System.Web.Services
4. Imports System.Web.Services.Protocols
5. Imports System.Data.SqlClient
6. Imports System.Data
7.
8.
9. <WebService(Namespace:="http://www.vinodkotiya.com/")> _
10. <WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
11. <Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
12. <System.Web.Script.Services.ScriptService()> _
13. Public Class ChatService
14.     Inherits System.Web.Services.WebService
15.     <WebMethod()> _
16.     Public Sub rebindGridView2(ByVal query As String, ByVal gridViewControl As
GridView)
17.         'Binds Paging/Sorting GridView with data from the specified query
18.         ' Bind GridView to current query & always store ur query into
session("currentquery") before calling
19.         ' reason is whenever page indexed changed or sort.. then it will show
data from currentquery
20.
21.         Using connection As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
22.
23.             Dim sqlComm As SqlCommand

```

```

24.         Dim sqlReader As SqlDataReader
25.         Dim dt As New DataTable()
26.         Dim dataTableRowCount As Integer
27.         Try
28.             connection.Open()
29.             sqlComm = New SqlCommand(query, connection)
30.             sqlReader = sqlComm.ExecuteReader()
31.             dt.Load(sqlReader)
32.             dataTableRowCount = dt.Rows.Count
33.
34.             If dataTableRowCount > 0 Then
35.                 gridViewControl.DataSource = dt
36.                 gridViewControl.DataBind()
37.             End If
38.             sqlComm.Dispose()
39.             connection.Close()
40.             sqlReader.Close()
41.         Catch e As Exception
42.             'lblDebug.text = e.Message
43.
44.             connection.Close()
45.         End Try
46.
47.
48.         connection.Close()
49.     End Using
50. End Sub
51. <WebMethod()> _
52. Public Function trackOnlineVisitor(ByVal ipaddress As String) As String
53.
54.     Using connection As SqlConnection = New
55.         SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
56.         ng)
57.         Dim sqlComm As SqlCommand
58.         Dim sqlReader As SqlDataReader
59.         Dim dt As New DataTable()
60.
61.         Try
62.
63.             connection.Open()
64.
65.             sqlComm = New SqlCommand("select * from onlineusers WHERE
66.             (ipaddress = '" + ipaddress + "') ", connection) ' fname + ' ' + lname AS name
67.             sqlReader = sqlComm.ExecuteReader()
68.             dt.Load(sqlReader)
69.             sqlReader.Close()
70.             If dt.Rows.Count = 1 Then ' if ip exists
71.                 dt.Dispose()
72.                 sqlComm.Dispose()
73.                 'connection.Close()
74.                 'connection.Open()
75.                 sqlComm = New SqlCommand("UPDATE onlineusers set lastlogon
76.                 = '" + Now.TimeOfDay().ToString + "' WHERE (ipaddress = '" + ipaddress + "') ",
77.                 connection) ' fname + ' ' + lname AS name
78.                 sqlReader = sqlComm.ExecuteReader()
79.                 sqlComm.Dispose()
80.                 sqlReader.Close()
81.                 'connection.Close()

```

```

77.
78.             ' now delete idle users having more then 5 minutes or
             having -ve value of yesterdays
79.
80.             'connection.Open()
81.             sqlComm = New SqlCommand("DELETE FROM onlineusers WHERE
(DATEDIFF(mi, lastlogon, CONVERT(VARCHAR(8), GETDATE(), 108)) > 1 or
DATEDIFF(mi, lastlogon, CONVERT(VARCHAR(8), GETDATE(), 108)) < 0)", connection)
             ' delete who are ideal more than 1 min
82.             'SELECT          DATEDIFF(mi, lastlogon, CONVERT(VARCHAR(8),
GETDATE(), 108)) AS Expr1 FROM          onlineusers
83.             sqlReader = sqlComm.ExecuteReader()
84.             sqlComm.Dispose()
85.             sqlReader.Close()
86.             connection.Close()
87.             Return "updated"
88.
89.             Else             'insert ip
90.             'connection.Close()
91.             'connection.Open()
92.             sqlComm = New SqlCommand("Insert Into onlineusers (
ipaddress, lastlogon) " & _
93.             " VALUES ( '" + ipaddress + "' , '" + Now.TimeOfDay().ToString + "'
) ", connection)
94.             sqlReader = sqlComm.ExecuteReader()
95.             sqlComm.Dispose()
96.             connection.Close()
97.             sqlReader.Close()
98.             Return "new insertion"
99.             End If
100.
101.
102.             Catch e As Exception
103.             'lblDebug.text = e.Message
104.             connection.Close()
105.             Return e.Message
106.             End Try
107.             connection.Close()
108.         End Using
109.
110.     End Function
111.     <WebMethod()> _
112.     Public Function GetMessages(ByVal strFromUserID As String, ByVal
strToUserID As String) As String
113.         Dim blnSucess As Boolean = False
114.         Dim strMessage As String
115.         Dim dDateSent As Date
116.
117.         strMessage = ""
118.
119.         Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
120.             conn.Open()
121.
122.             Dim cmd As SqlCommand
123.
124.             'Dim MemUser As MembershipUser

```

```

125.         Dim strFromUserGUID As String
126.         'MemUser = Membership.GetUser(HttpUtility.UrlDecode(strFromUserID))
127.         'strFromUserGUID = MemUser.ProviderUserKey.ToString
128.         Dim strToUserGUID As String
129.         'MemUser = Membership.GetUser(HttpUtility.UrlDecode(strToUserID))
130.         'strToUserGUID = MemUser.ProviderUserKey.ToString
131.
132.         strFromUserGUID = strFromUserID
133.         strToUserGUID = strToUserID
134.
135.         cmd = New SqlCommand("SELECT TOP(1) Message, DateSent FROM
[IMChats] WHERE ([SenderUserID] = @SenderUserID AND [RecipientUserID] =
@RecipientUserID) ORDER BY DateSent DESC", conn)
136.         cmd.Parameters.AddWithValue("@SenderUserID", strToUserGUID)
137.         cmd.Parameters.AddWithValue("@RecipientUserID", strFromUserGUID)
138.         Dim reader As SqlDataReader = cmd.ExecuteReader
139.         While reader.Read()
140.             dDateSent = reader("DateSent")
141.             'If dDateSent > DateAdd(DateInterval.Minute, -5, DateTime.Now)
Then
142.                 strMessage = reader("Message")
143.                 blnSucess = True
144.                 'End If
145.             End While
146.             reader.Close()
147.             cmd.Dispose()
148.
149.             If blnSucess Then
150.                 cmd = conn.CreateCommand()
151.                 cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID]
= @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
152.                 cmd.Parameters.AddWithValue("@SenderUserID", strToUserGUID)
153.                 cmd.Parameters.AddWithValue("@RecipientUserID",
strFromUserGUID)
154.                 cmd.ExecuteNonQuery()
155.                 cmd.Dispose()
156.             End If
157.             conn.Close()
158.
159.         End Using
160.         Return strMessage
161.     End Function
162.     <WebMethod()> _
163.     Public Function SendMessage(ByVal strSenderUserID As String, ByVal
strRecipientUserID As String, ByVal strMessage As String) As String
164.         Dim blnSucess As Boolean = False
165.         Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
166.
167.             conn.Open()
168.
169.             Dim cmd As SqlCommand
170.
171.             'Dim MemUser As MembershipUser
172.             Dim strRecipientUserGUID As String
173.             'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))

```

```

174.         'strRecipientUserGUID = MemUser.ProviderUserKey.ToString
175.         Dim strSenderUserGUID As String
176.         'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
177.         'strSenderUserGUID = MemUser.ProviderUserKey.ToString
178.
179.         strRecipientUserGUID = strRecipientUserID
180.         strSenderUserGUID = strSenderUserID
181.
182.         cmd = conn.CreateCommand()
183.         cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID] =
@SenderUserID AND [RecipientUserID] = @RecipientUserID)"
184.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
185.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
186.         cmd.ExecuteNonQuery()
187.         cmd.Dispose()
188.
189.         cmd = conn.CreateCommand()
190.         cmd.CommandText = "INSERT INTO IMChats(RecipientUserID,
SenderUserID, Message, DateSent) VALUES (@RecipientUserID, @SenderUserID,
@Message, @DateSent)"
191.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
192.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
193.         cmd.Parameters.AddWithValue("@Message", strMessage)
194.         cmd.Parameters.AddWithValue("@DateSent", DateTime.Now)
195.
196.         cmd.ExecuteNonQuery()
197.         blnSucess = True
198.         cmd.Dispose()
199.         conn.Close()
200.     End Using
201.     Return blnSucess
202. End Function
203.
204.         'ENQ = chr(5)                                # Request to send
205.         'EOT = chr(4)                                # Ready to receive
206.         'ACK = chr(6)                                # Correct reception
207.         'NAK = chr(21)                               # Incorrect reception
208.         '//////////
209.         '// Sender init chat
210.         '//////////
211.         '// Sender ENQ      Wait 4 ACK
212.         '// Recip  ACK ENQ Wait 4 ACK ACK
213.         '// Sender ACK ACK Wait 4 EOT
214.         '// Recip  EOT
215.         '//////////
216.         '// Recip wait 4 chat
217.         '//////////
218.         '// Recip  ACK ENQ Wait 4 ACK
219.         '// Sender ACK ACK Wait 4 EOT
220.         '// Recip  EOT
221.
222.     <WebMethod()> _
223.     Public Function CheckChatReq(ByVal strMyUserID As String) As String
224.         Dim strSenderUserID, strMessage As String
225.         Dim dDateSent As Date

```

```

226.         strSenderUserID = ""
227.         Using conn As SqlConnection = New
                SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
228.             conn.Open()
229.
230.             Dim cmd As SqlCommand
231.
232.             Dim strMyUserGUID As String = ""
233.             Dim strSenderUserGUID As String = ""
234.             'Dim MemUser As MembershipUser =
Membership.GetUser(HttpUtility.UrlDecode(strMyUserID))
235.             'strMyUserGUID = MemUser.ProviderUserKey.ToString
236.             strMyUserGUID = strMyUserID
237.
238.
239.             cmd = New SqlCommand("SELECT TOP(1) SenderUserID, Message, DateSent
FROM [IMChats] WHERE ([RecipientUserID] = @RecipientUserID) ORDER BY DateSent
DESC", conn)
240.             cmd.Parameters.AddWithValue("@RecipientUserID", strMyUserGUID)
241.             Dim reader As SqlDataReader = cmd.ExecuteReader
242.             While reader.Read()
243.                 dDateSent = reader("DateSent")
244.                 'If dDateSent > DateAdd(DateInterval.Minute, -5, DateTime.Now)
Then
245.                     strMessage = reader("Message")
246.                     ' ENQ
247.                     If strMessage = "~::~:=[(HANDSHAKE)]=::~:~[ENQ]" Then
248.                         strSenderUserGUID = reader("SenderUserID").ToString
249.                         'strSenderUserID = Membership.GetUser(New
Guid(strSenderUserGUID)).UserName
250.                         strSenderUserID = strSenderUserGUID
251.                         'End If
252.                     End If
253.             End While
254.             reader.Close()
255.             cmd.Dispose()
256.
257.             If Not String.IsNullOrEmpty(strSenderUserGUID) Then
258.                 cmd = conn.CreateCommand()
259.                 cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID]
= @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
260.                 cmd.Parameters.AddWithValue("@SenderUserID",
strSenderUserGUID)
261.                 cmd.Parameters.AddWithValue("@RecipientUserID", strMyUserGUID)
262.                 cmd.ExecuteNonQuery()
263.                 cmd.Dispose()
264.             End If
265.             conn.Close()
266.
267.         End Using
268.         Return strSenderUserID
269.     End Function
270.
271.     <WebMethod()> _
272.     Public Function SendChatReq(ByVal strSenderUserID As String, ByVal
strRecipientUserID As String) As Boolean
273.         Dim blnSucess As Boolean = False

```

```

274.         Using conn As SqlConnection = New
                SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
                ng)
275.
276.             conn.Open()
277.
278.             Dim cmd As SqlCommand
279.
280.                 ' Dim MemUser As MembershipUser
281.             Dim strRecipientUserGUID As String
282.             ' MemUser =
                Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
283.             ' strRecipientUserGUID = MemUser.ProviderUserKey.ToString
284.             Dim strSenderUserGUID As String
285.             ' MemUser =
                Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
286.             ' strSenderUserGUID = MemUser.ProviderUserKey.ToString
287.             strRecipientUserGUID = strRecipientUserID
288.             strSenderUserGUID = strSenderUserID
289.
290.             cmd = conn.CreateCommand()
291.             cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID] =
                @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
292.             cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
293.             cmd.Parameters.AddWithValue("@RecipientUserID",
                strRecipientUserGUID)
294.             cmd.ExecuteNonQuery()
295.             cmd.Dispose()
296.             cmd = conn.CreateCommand()
297.             cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID] =
                @RecipientUserID AND [RecipientUserID] = @SenderUserID)"
298.             cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
299.             cmd.Parameters.AddWithValue("@RecipientUserID",
                strRecipientUserGUID)
300.             cmd.ExecuteNonQuery()
301.             cmd.Dispose()
302.
303.             cmd = conn.CreateCommand()
304.             cmd.CommandText = "INSERT INTO IMChats(RecipientUserID,
                SenderUserID, Message, DateSent) VALUES (@RecipientUserID, @SenderUserID,
                @Message, @DateSent)"
305.             cmd.Parameters.AddWithValue("@RecipientUserID",
                strRecipientUserGUID)
306.             cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
307.             ' ENQ
308.             cmd.Parameters.AddWithValue("@Message",
                "~::~:=[(HANDSHAKE)]=::~:~[ENQ]")
309.             cmd.Parameters.AddWithValue("@DateSent", DateTime.Now)
310.
311.             cmd.ExecuteNonQuery()
312.             blnSucess = True
313.             cmd.Dispose()
314.             conn.Close()
315.         End Using
316.         Return blnSucess
317.     End Function
318.     <WebMethod()> _
319.     Public Function ChatCleanUpAll(ByVal strMyUserID As String) As String

```



```

320.         Dim blnSucess As Boolean = False
321.         Using conn As SqlConnection = New
322.             SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
323.                 ng)
324.                 conn.Open()
325.
326.                 Dim cmd As SqlCommand
327.
328.                 Dim strMyUserGUID As String
329.                 ' Dim MemUser As MembershipUser =
330.                 Membership.GetUser(HttpUtility.UrlDecode(strMyUserID))
331.                 ' strMyUserGUID = MemUser.ProviderUserKey.ToString
332.                 strMyUserGUID = strMyUserID
333.
334.                 cmd = conn.CreateCommand()
335.                 cmd.CommandText = "DELETE FROM [IMChats] WHERE ([RecipientUserID] =
336.                     @RecipientUserID)"
337.                 cmd.Parameters.AddWithValue("@RecipientUserID", strMyUserGUID)
338.                 cmd.ExecuteNonQuery()
339.                 cmd.Dispose()
340.                 conn.Close()
341.                 blnSucess = True
342.         End Using
343.         Return blnSucess
344.     End Function
345.     <WebMethod(> _
346.     Public Function Cleanup(ByVal strMyUserID As String, ByVal strSenderUserID
347.         As String, ByVal blnSendEOT As Boolean) As Boolean
348.         Dim blnSucess As Boolean = False
349.         Using conn As SqlConnection = New
350.             SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
351.                 ng)
352.                 conn.Open()
353.
354.                 Dim cmd As SqlCommand
355.
356.                 Dim strMyUserGUID As String
357.                 ' Dim MemUser As MembershipUser =
358.                 Membership.GetUser(HttpUtility.UrlDecode(strMyUserID))
359.                 ' strMyUserGUID = MemUser.ProviderUserKey.ToString
360.                 Dim strSenderUserGUID As String
361.                 ' MemUser =
362.                 Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
363.                 ' strSenderUserGUID = MemUser.ProviderUserKey.ToString
364.                 strMyUserGUID = strMyUserID
365.                 strSenderUserGUID = strSenderUserID
366.
367.                 cmd = conn.CreateCommand()
368.                 cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID] =
369.                     @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
370.                 cmd.Parameters.AddWithValue("@RecipientUserID", strMyUserGUID)
371.                 cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
372.                 cmd.ExecuteNonQuery()
373.                 cmd.Dispose()
374.
375.                 If blnSendEOT Then
376.                     cmd = conn.CreateCommand()

```

```

367.         cmd.CommandText = "INSERT INTO IMChats(RecipientUserID,
SenderUserID, Message, DateSent) VALUES (@RecipientUserID, @SenderUserID,
@Message, @DateSent)"
368.         cmd.Parameters.AddWithValue("@RecipientUserID",
strSenderUserGUID)
369.         cmd.Parameters.AddWithValue("@SenderUserID", strMyUserGUID)
370.         ' EOT
371.         cmd.Parameters.AddWithValue("@Message",
"~::~:=[(HANDSHAKE)]=::~~[EOT]")
372.         cmd.Parameters.AddWithValue("@DateSent", DateTime.Now)
373.         cmd.ExecuteNonQuery()
374.         cmd.Dispose()
375.     End If
376.     conn.Close()
377.     blnSucess = True
378.
379.     End Using
380.     Return blnSucess
381.
382. End Function
383. <WebMethod()> _
384. Public Function SendNak(ByVal strSenderUserID As String, ByVal
strRecipientUserID As String) As Boolean
385.     Dim blnSucess As Boolean = False
386.     Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
387.
388.         conn.Open()
389.
390.         Dim cmd As SqlCommand
391.
392.         ' Dim MemUser As MembershipUser
393.         Dim strRecipientUserGUID As String = strRecipientUserID
394.         ' MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
395.         ' strRecipientUserGUID = MemUser.ProviderUserKey.ToString
396.         Dim strSenderUserGUID As String = strSenderUserID
397.         ' MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
398.         ' strSenderUserGUID = MemUser.ProviderUserKey.ToString
399.
400.         cmd = conn.CreateCommand()
401.         cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID] =
@SenderUserID AND [RecipientUserID] = @RecipientUserID)"
402.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
403.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
404.         cmd.ExecuteNonQuery()
405.         cmd.Dispose()
406.
407.         cmd = conn.CreateCommand()
408.         cmd.CommandText = "INSERT INTO IMChats(RecipientUserID,
SenderUserID, Message, DateSent) VALUES (@RecipientUserID, @SenderUserID,
@Message, @DateSent)"
409.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
410.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)

```

```

411.         ' NAK
412.         cmd.Parameters.AddWithValue("@Message",
"~::~:=[(HANDSHAKE)]=::~~[NAK]")
413.         cmd.Parameters.AddWithValue("@DateSent", DateTime.Now)
414.
415.         cmd.ExecuteNonQuery()
416.         blnSucess = True
417.         cmd.Dispose()
418.         conn.Close()
419.     End Using
420.     Return blnSucess
421. End Function
422. <WebMethod()> _
423. Public Function SendAck(ByVal strSenderUserID As String, ByVal
strRecipientUserID As String) As Boolean
424.     Dim blnSucess As Boolean = False
425.     Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
426.
427.         conn.Open()
428.
429.         Dim cmd As SqlCommand
430.
431.         ' Dim MemUser As MembershipUser
432.         Dim strRecipientUserGUID As String = strRecipientUserID
433.         'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
434.         'strRecipientUserGUID = MemUser.ProviderUserKey.ToString
435.         Dim strSenderUserGUID As String = strSenderUserID
436.         'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
437.         'strSenderUserGUID = MemUser.ProviderUserKey.ToString
438.
439.         cmd = conn.CreateCommand()
440.         cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID]
= @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
441.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
442.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
443.         cmd.ExecuteNonQuery()
444.         cmd.Dispose()
445.
446.         cmd = conn.CreateCommand()
447.         cmd.CommandText = "INSERT INTO IMChats(RecipientUserID,
SenderUserID, Message, DateSent) VALUES (@RecipientUserID, @SenderUserID,
@Message, @DateSent)"
448.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
449.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
450.         ' ACK
451.         cmd.Parameters.AddWithValue("@Message",
"~::~:=[(HANDSHAKE)]=::~~[ACK]")
452.         cmd.Parameters.AddWithValue("@DateSent", DateTime.Now)
453.
454.         cmd.ExecuteNonQuery()
455.         blnSucess = True
456.         cmd.Dispose()

```

```

457.         conn.Close()
458.     End Using
459.     Return blnSucess
460. End Function
461. <WebMethod()> _
462. Public Function GetAck(ByVal strRecipientUserID As String, ByVal
strSenderUserID As String) As Boolean
463.     Dim blnSucess As Boolean = False
464.     Dim strMessage As String
465.     Dim dDateSent As Date
466.
467.     Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
468.         conn.Open()
469.
470.         Dim cmd As SqlCommand
471.
472.         ' Dim MemUser As MembershipUser
473.         Dim strRecipientUserGUID As String = strRecipientUserID
474.         ' MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
475.         'strRecipientUserGUID = MemUser.ProviderUserKey.ToString
476.         Dim strSenderUserGUID As String = strSenderUserID
477.         'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
478.         'strSenderUserGUID = MemUser.ProviderUserKey.ToString
479.
480.         cmd = New SqlCommand("SELECT TOP(1) Message, DateSent FROM
[IMChats] WHERE ([SenderUserID] = @SenderUserID AND [RecipientUserID] =
@RecipientUserID) ORDER BY DateSent DESC", conn)
481.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
482.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
483.         Dim reader As SqlDataReader = cmd.ExecuteReader
484.         While reader.Read()
485.             dDateSent = reader("DateSent")
486.             'If dDateSent > DateAdd(DateInterval.Minute, -5, DateTime.Now)
Then
487.                 strMessage = reader("Message")
488.                 ' ACK
489.                 If strMessage = "~::~:=[(HANDSHAKE)]=::~:~[ACK]" Then
490.                     blnSucess = True
491.                     'End If
492.                 End If
493.             End While
494.             reader.Close()
495.             cmd.Dispose()
496.
497.             If blnSucess Then
498.                 cmd = conn.CreateCommand()
499.                 cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID]
= @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
500.                 cmd.Parameters.AddWithValue("@SenderUserID",
strSenderUserGUID)
501.                 cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
502.                 cmd.ExecuteNonQuery()

```

```

503.         cmd.Dispose()
504.     End If
505.     conn.Close()
506.
507.     End Using
508.     Return blnSucess
509. End Function
510. <WebMethod()> _
511. Public Function GetNak(ByVal strRecipientUserID As String, ByVal
strSenderUserID As String) As Boolean
512.     Dim blnSucess As Boolean = False
513.     Dim strMessage As String
514.     Dim dDateSent As Date
515.
516.     Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
517.         conn.Open()
518.
519.         Dim cmd As SqlCommand
520.
521.         ' Dim MemUser As MembershipUser
522.         Dim strRecipientUserGUID As String = strRecipientUserID
523.         ' MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
524.         ' strRecipientUserGUID = MemUser.ProviderUserKey.ToString
525.         Dim strSenderUserGUID As String = strSenderUserID
526.         ' MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
527.         ' strSenderUserGUID = MemUser.ProviderUserKey.ToString
528.
529.         cmd = New SqlCommand("SELECT TOP(1) Message, DateSent FROM
[IMChats] WHERE ([SenderUserID] = @SenderUserID AND [RecipientUserID] =
@RecipientUserID) ORDER BY DateSent DESC", conn)
530.         cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
531.         cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
532.         Dim reader As SqlDataReader = cmd.ExecuteReader
533.         While reader.Read()
534.             dDateSent = reader("DateSent")
535.             'If dDateSent > DateAdd(DateInterval.Minute, -5, DateTime.Now)
Then
536.                 strMessage = reader("Message")
537.                 ' NAK
538.                 If strMessage = "~::~:=[(HANDSHAKE)]=::~:~[NAK]" Then
539.                     blnSucess = True
540.                     'End If
541.                 End If
542.             End While
543.         reader.Close()
544.         cmd.Dispose()
545.
546.         If blnSucess Then
547.             cmd = conn.CreateCommand()
548.             cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID]
= @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
549.             cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)

```

```

550.            cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
551.            cmd.ExecuteNonQuery()
552.            cmd.Dispose()
553.        End If
554.        conn.Close()
555.
556.        End Using
557.        Return blnSucess
558.    End Function
559.    <WebMethod()> _
560.    Public Function SendEot(ByVal strSenderUserID As String, ByVal
strRecipientUserID As String) As Boolean
561.        Dim blnSucess As Boolean = False
562.        Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
563.
564.            conn.Open()
565.
566.            Dim cmd As SqlCommand
567.
568.            ' Dim MemUser As MembershipUser
569.            Dim strRecipientUserGUID As String = strRecipientUserID
570.            'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
571.            'strRecipientUserGUID = MemUser.ProviderUserKey.ToString
572.            Dim strSenderUserGUID As String = strSenderUserID
573.            'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
574.            'strSenderUserGUID = MemUser.ProviderUserKey.ToString
575.
576.            cmd = conn.CreateCommand()
577.            cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID] =
@RecipientUserID AND [RecipientUserID] = @SenderUserID)"
578.            cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
579.            cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
580.            cmd.ExecuteNonQuery()
581.            cmd.Dispose()
582.
583.            cmd = conn.CreateCommand()
584.            cmd.CommandText = "INSERT INTO IMChats(RecipientUserID,
SenderUserID, Message, DateSent) VALUES (@RecipientUserID, @SenderUserID,
@Message, @DateSent)"
585.            cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
586.            cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
587.            ' EOT
588.            cmd.Parameters.AddWithValue("@Message",
"~::~:=[(HANDSHAKE)]=::~:~[EOT]")
589.            cmd.Parameters.AddWithValue("@DateSent", DateTime.Now)
590.
591.            cmd.ExecuteNonQuery()
592.            blnSucess = True
593.            cmd.Dispose()
594.            conn.Close()
595.        End Using

```

```

596.         Return blnSucess
597.     End Function
598.     <WebMethod()> _
599.     Public Function GetEot(ByVal strRecipientUserID As String, ByVal
strSenderUserID As String) As Boolean
600.         Dim blnSucess As Boolean = False
601.         Dim strMessage As String
602.         Dim dDateSent As Date
603.
604.         Using conn As SqlConnection = New
SqlConnection(ConfigurationManager.ConnectionStrings("vinConn2").ConnectionStri
ng)
605.             conn.Open()
606.
607.             Dim cmd As SqlCommand
608.
609.             ' Dim MemUser As MembershipUser
610.             Dim strRecipientUserGUID As String = strRecipientUserID
611.             'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strRecipientUserID))
612.             'strRecipientUserGUID = MemUser.ProviderUserKey.ToString
613.             Dim strSenderUserGUID As String = strSenderUserID
614.             'MemUser =
Membership.GetUser(HttpUtility.UrlDecode(strSenderUserID))
615.             'strSenderUserGUID = MemUser.ProviderUserKey.ToString
616.
617.             cmd = New SqlCommand("SELECT TOP(1) Message, DateSent FROM
[IMChats] WHERE ([SenderUserID] = @SenderUserID AND [RecipientUserID] =
@RecipientUserID) ORDER BY DateSent DESC", conn)
618.             cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
619.             cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
620.             Dim reader As SqlDataReader = cmd.ExecuteReader
621.             While reader.Read()
622.                 dDateSent = reader("DateSent")
623.                 'If dDateSent > DateAdd(DateInterval.Minute, -5, DateTime.Now)
Then
624.                     strMessage = reader("Message")
625.                     ' EOT
626.                     If strMessage = "~::~:=[(HANDSHAKE)]=::~:~[EOT]" Then
627.                         blnSucess = True
628.                         'End If
629.                     End If
630.                 End While
631.             reader.Close()
632.             cmd.Dispose()
633.
634.             If blnSucess Then
635.                 cmd = conn.CreateCommand()
636.                 cmd.CommandText = "DELETE FROM [IMChats] WHERE ([SenderUserID]
= @SenderUserID AND [RecipientUserID] = @RecipientUserID)"
637.                 cmd.Parameters.AddWithValue("@SenderUserID", strSenderUserGUID)
638.                 cmd.Parameters.AddWithValue("@RecipientUserID",
strRecipientUserGUID)
639.                 cmd.ExecuteNonQuery()
640.                 cmd.Dispose()
641.             End If
642.             conn.Close()

```

```
643.  
644.      End Using  
645.      Return blnSucess  
646.  End Function  
647.  
648.End Class
```