



NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

ONLINE BOOKING SYSTEM

**Graduation Project
COM 400**

Students: İSKENDER CİHAN (20031426)

Supervisor: ÜMİT SOYER

Nicosia – 2008

ACKNOWLEDGMENT

“ First i would like to thank my supervisor Umit Soyer for his invaluable advice and belief in my work and myself over the course fo this Graduation Project..

Second , I would like to express my gratitude to Near East University for the scholarship that made the work possible.

Third , I thank my family for their constant encouragement and support during the preparation of this project.

Finally I would also like to thank all my friends Nadeem and Serkan for their advice and support. “

LIST OF ABBREVIATIONS

NEU Near East University

WWW World Wide Web

DMS Database Management Systems

SQL Structured Query Language

ASP Active Server Pages

.NET .NET Framework

C# C sharp

TABLE OF CONTENTS

ACKNOWLEDGMENT	i
LIST OF ABBREVIATIONS	ii
TABLE OF CONTENTS	iii
INTRODUCTION	1
 CHAPTER 1 : MASTER PAGES	 2
1.1 - MASTER PAGE	2
1.1 -a) BaşSayfa.Master	2
1.1-b) CODE ASP FOR MASTER PAGE başsayfa.master	2
1.1 -c) CODE C# FOR MASTER PAGE başsayfa.master.....	4
 CHAPTER 2 : ALLUSERS PAGES	 5
2.1 - FOR GUESTS Anasayfa.aspx	5
2.1 – a) ASP CODE FOR GUEST Anasayfa.aspx	5
2.2 - FOR Admin Anasayfa.aspx.....	7
2.2 - a) ASP CODE FOR Admin Anasayfa.aspx	7
2.2 – b) C# Code FOR Anasayfa.aspx.....	8
2.3 - Search.aspx	9
2.3 – a) ASP CODE FOR Search.aspx	10
2.3 - b) C# CODE FOR search.aspx	11
2.4 - Hesap.aspx	12
2.4 – a) ASP CODE FOR hesap.aspx	12
2.4 - b) C# CODE FOR hesap.aspx	15
2.4 - c) Continue to hesap .aspx and Complete to create user	16
2.5 - Login Page	17

2.5 – a) ASP CODE FOR Login.aspx	17
2.5 – b) C# CODE FOR Login.aspx	19
2.6 - ÜrünSeçme.aspx Page	20
2.6 – a) ASP CODE FOR UrunSecme.aspx	21
2.6 – b) C# CODE FOR UrunSecme.aspx	25
2.7 - Sepet PAGE	26
2.7 – a) ASP CODE FOR Sepet.aspx	26
2.7 - b) C# CODE FOR Sepet.aspx	28
2.8 - SatınAl PAGE	29
2.8 - a) ASP CODE FOR SatınAl.aspx	29
2.8 - b) CODE C# FOR SatınAl.aspx	31
 CHAPTER 3 : ONLY ADMIN PAGES	 33
3.1 - PAGE KategoriEkle.aspx	33
3.1 – a) ASP CODE FOR KategoriEkle.aspx	33
3.1 – b) C# CODE FOR KategoriEkle.aspx	36
3.2 - PAGE MarkaEkle.aspx	37
3.2 - a) ASP CODE FOR MarkaEkle.aspx	37
3.2 – b) C# CODE FOR MarkaEkle.aspx	40
3.3 - PAGE ModelEkle	41
3.3 – a) ASP CODE FOR ModelEkle.aspx	41
3.3 – b) C# CODE FOR ModelEkle.aspx	45
3.4 – PAGE UrunEkle.aspx	46
3.4 – a) ASP CODE FOR UrunEkle.aspx	47
3.4 – b) C# CODE FOR UrunEkle.aspx	49
 CHAPER 4 : CONFIGURATION PAGES	 51

4.1 - CONFIGURATION FOR ADMIN PAGES	51
4.1 – a) PAGE Web.config	51
4.2 - CONFIGURATION FOR ALL USERS	52
4.2 – a) PAGE web.config	52
 CHAPTER 5 : WORK ENVIRONMENT	 53
 CHAPTER 6 : TABLES	 54
6.1 - APPLICATION TABLE	54
6.2 - MEMBERSHIP TABLE	54
6.3 - PATH TABLE	55
6.4 - PERSONALIZATION ALLUSER TABLE	55
6.5 - PERSONALIZATION PER USER TABLE	55
6.6 - PROFILE TABLE	56
6.7 - ROLES TABLE	56
6.8 - SCHEMA TABLE	56
6.9 - USERS TABLE	56
6.10 - USERSINROLE TABLE	57
6.11 - WEB EVENT TABLE	57
6.12 - KATEGORI TABLE	58
6.13 - MARKA TABLE	58
6.14 - MODEL TABLE	58
6.15 - MODELOZELLIK TABLE	58
6.16 - SEPET TABLE	59
6.17 - URUN TABLE	59
 CHAPTER 7 : QUERIES AND DESIGNS AND RESULTS FOR VIEWS	 60

7.1 - DETAY VIEWS	60
7.1 – a) RESULT	61
7.2 - SEPET VIEW	61
7.2 – a) RESULT	62
7.3 - APPLICATIONS VIEW	63
7.4 - MEMBERSHIPUSERS VIEW	64
7.5 - PROFILES VIEW	64
7.6 - ROLES VIEW	65
7.7 - USERS VIEW	65
7.8 - USERSINROLES VIEW	65
7.9 - PATHS VIEW	65
7.10 - SHARED VIEW	65
7.11 - WEBPARTUSERS VIEWS	66
CHAPTER 8 : STORED PROCEDURES	67
8.1 - ANYDATAINTABLES	67
8.2 - CREATE APLICATION	69
8.3 - CHECHSCHEMAVERSION	70
8.4 - EDITMEMBERSHIP	71
8.5 - CREATEUSER	72
8.6 - FINDUSERBYEMAIL	76
8.7 - FINDUSERSBYNAME	78
8.8 - GETALLUSERS	79
8.9 - GETNUMBEROFUSERSONLINE	80
8.10 - GETPASSWORD	81
8.11 - GETPASSWORDWITHFORMAT	82
8.12 - GETUSERSBYEMAIL	83
8.13 - GETUSERBYNAME	84

8.14 - GETUSERBYUSERID	85
8.15 - RESETPASSWORD	86
8.16 - SETPASSWORD	89
8.17 - UNLOCKUSERS	89
8.18 - UPDATEUSERS	90
8.19 - UPDATEUSERINFO	92
8.20 - CREATEPATH	96
8.21 - GETAPPLICATIONID	96
8.22 - DELETEALLSTATE	97
8.23 - FINDSTATE	97
8.24 - GETCOUNTOFSTATE	100
8.25 - RESETSHAREDSTATE	101
8.26 - RESETUSERSTATE	101
8.27 - GETPAGESETTINGS	102
8.28 - RESETPAGESETTINGS	103
8.29 - SETPAGESETTINGS	104
8.30 - PERUSERGETPAGESETTINGS	105
8.31 - PERUSERRESETPAGESETTINGS	106
8.32 - PERUSERSETPAGESETTINGS	107
8.33 - DELETEINACTIVEUSERS	108
8.34 - DELETEPROFILES	109
8.35 - GETNUMBEROFINACTIVEUSERS	110
8.36 - GETPROFILES	111
8.37 - GETPROPERTIES	113
8.38 - SETPROPERTIES	113
8.39 - REGISTERSCHEMAVERSIONS	116
8.40 - CREATEROLES	116

8.41 - DELETEROLES	118
8.42 - GETALLROLES	120
8.43 - ROLEEXISTS	120
8.44 - REMOVEALLROLEMEMBERS	121
8.45 - RESTOREPERMISSIONS	122
8.46 - UNREGISTERSCHEMAVERSION	122
8.47 - CREATEUSER	123
8.48 - DELETEUSER	124
8.49 - ADDUSERSTOROLES	127
8.50 - FINDUSERSINROLES	130
8.51 - GETROLESFORUSERS	130
8.52 - GETUSERSINROLES	131
8.53 - ISUSERINROLES	132
8.54 - REMOVEUSERSFROMROLES	133
8.55 - LOGEVENT	136
CHAPTER 9 : TEN TERMS	138
9.1 - You Look Smart	138
9.2 - Web Application	138
9.3 - Developer	138
9.4 - Data-Driven	139
9.5 - ASP.NET 2.0	139
9.6 - Visual Studio	139
9.7 - IDE	140
9.8 - Control	140
9.9 - Code	141
9.10 - Programmatic	141
9.11 - Database	142

CHAPTER 10 : TEN ALTERNATIVES TO BEING HELPLESS	144
10.1 - Microsoft Developer Network (MSDN)	144
10.2 - HTML Home Page.....	145
10.3 - Cascading Style Sheets Home Page	145
10.4 - ASP.NET Forums	146
10.5 - SQL Server Developer Center	146
10.6 - Dotnetjunkies	146
10.7 - Microsoft Technical Communities	147
CONCLUSION	148
REFERENCES	149

INTRODUCTION

The Objective of this project is to build a Web Based Online System. The Project consists introduction nine chapter nad conclusion.

Chapter one presents the architecture of MASTER PAGE

Chapter two presents the architecture of ALLUSER PAGES

Chapter three presents the architecture of ONLYADMIN PAGES

Chapter four presents the architecture of CONFIGURATION PAGES

Chapter five describes WORK ENVIRONMENT

Chapter six describes the TABLES

Chapter seven describes QUERIES AND DESIGNS AND RESULTS FOR VIEWS

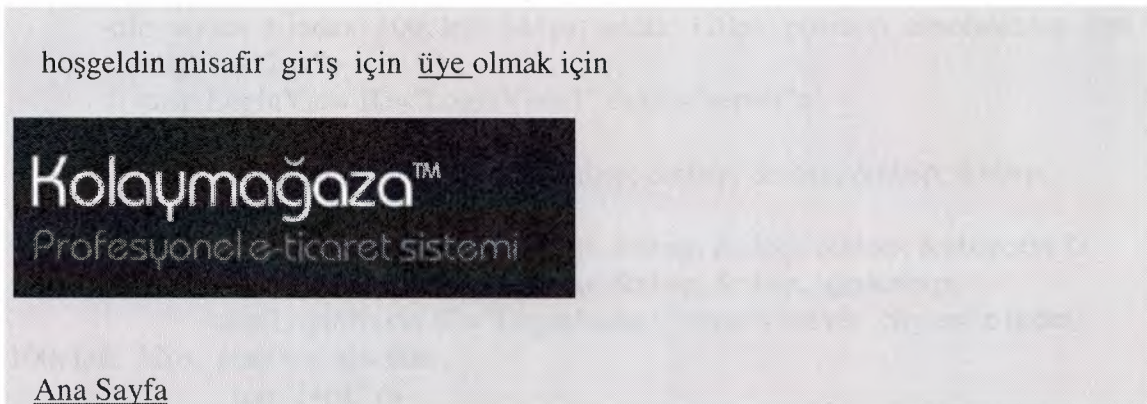
Chapter eight describes the PROCEDURES

Chapter nine describe some definition TEN TERMS

CHAPTER 1 : MASTER PAGES

1.1 - MASTER PAGE

1.1 -a) BaşSayfa.Master



1.1-b) CODE ASP FOR MASTER PAGE başsayfa.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="BaşSayfa.master.cs" Inherits="BaşSayfa" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
<script language="javascript" type="text/javascript">
<!--

function DIV1_onclick() {

}

// -->
</script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            &nbsp;<br />
            <br />
            <br />
            <br />
        </div>
    </form>
</body>
</html>
```


[illegible]

1.1 -c) CODE C# FOR MASTER PAGE başsayfa.master

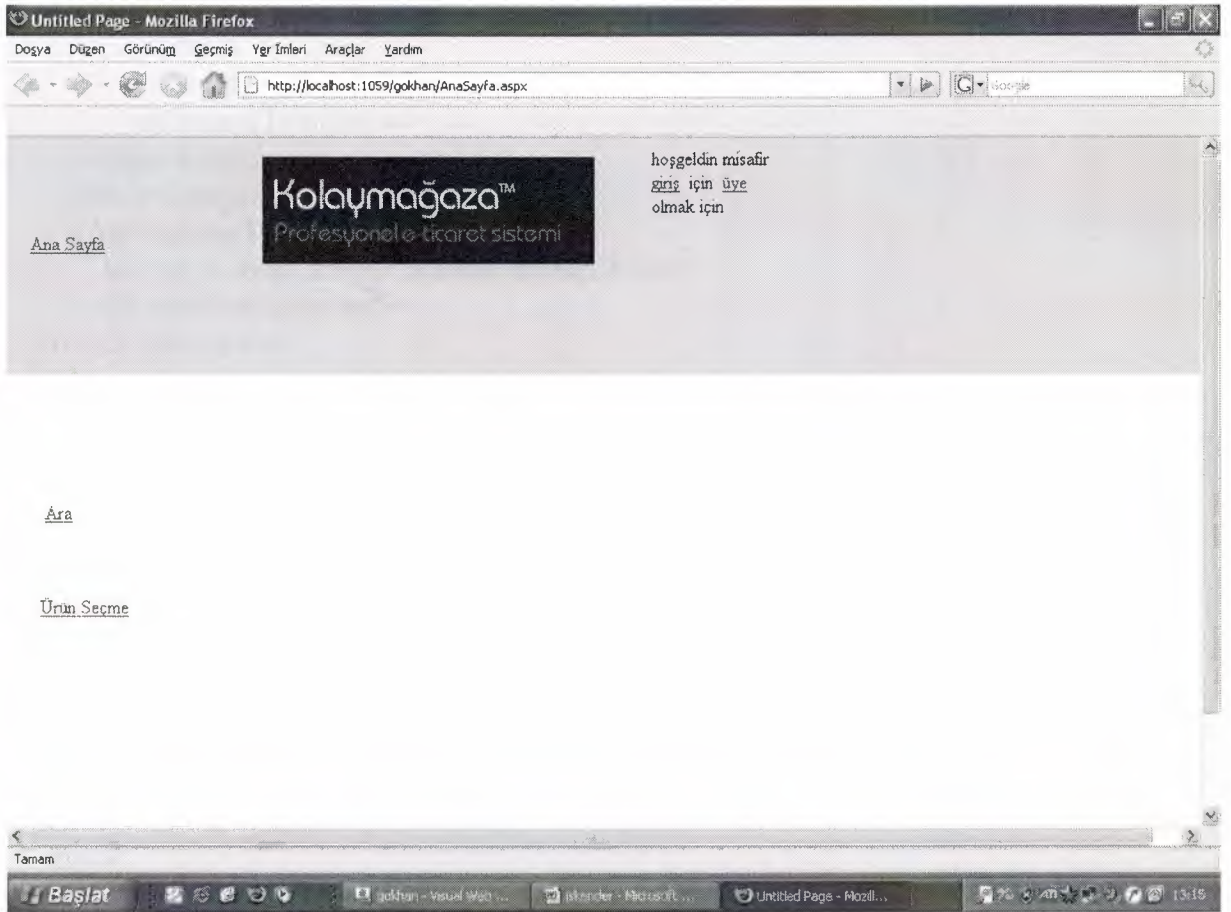
```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class BaşSayfa : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

CHAPTER 2 : ALLUSERS PAGES

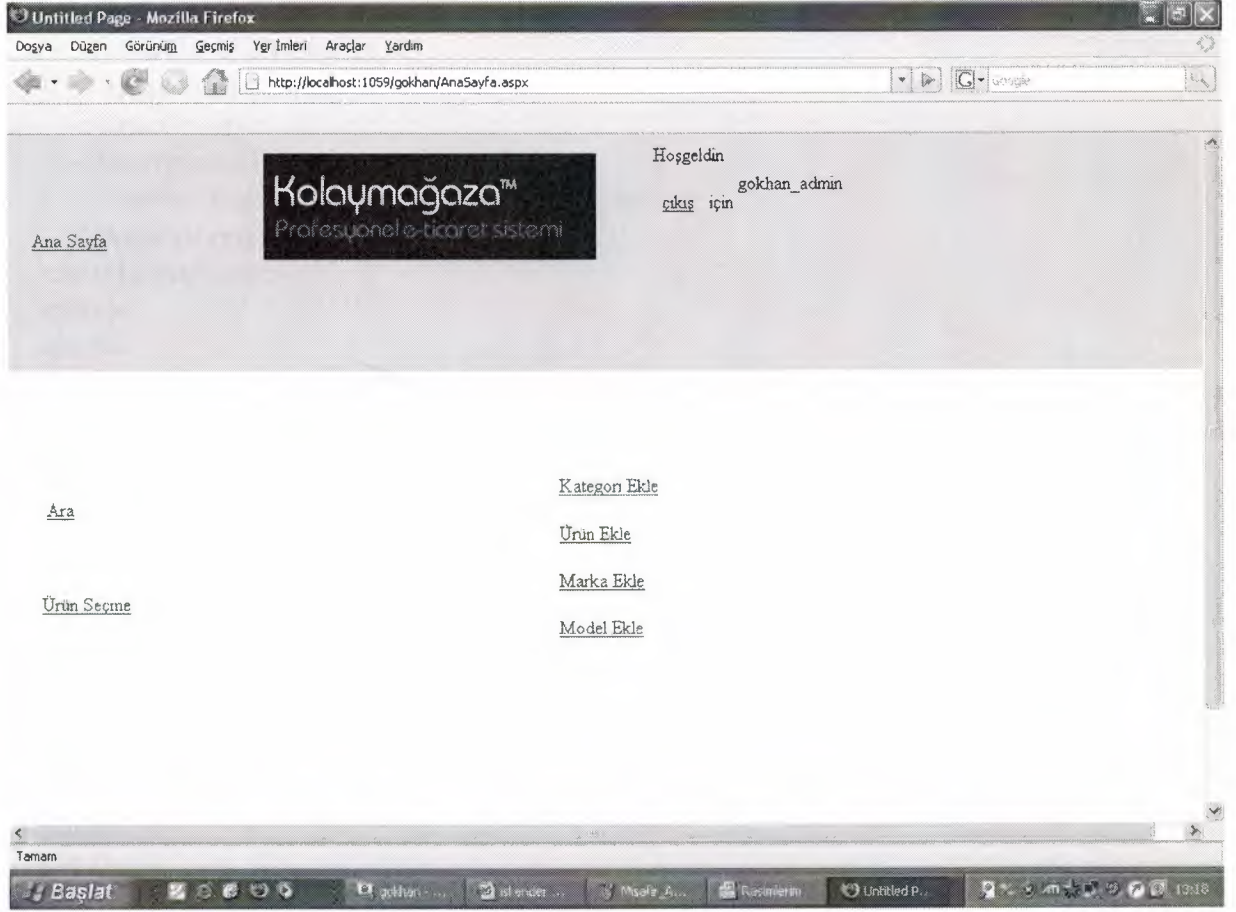
2.1 - FOR GUESTS Anasayfa.aspx



2.1 - a) ASP CODE FOR GUEST Anasayfa.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="AnaSayfa.aspx.cs" Inherits="AnaSayfa"
Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <br />
    &nbsp;
    <div style="z-index: 100; left: 464px; width: 100px; position: absolute; top: 288px;
    height: 100px">
    <asp:LoginView ID="LoginView1" runat="server">
        <RoleGroups>
            <asp:RoleGroup Roles="Admin">
                <ContentTemplate>
                    <a href="AdminPages/KategoriEkle.aspx">Kategori Ekle</a>
```


2.2 - FOR Admin Anasayfa.aspx



2.2 - a) ASP CODE FOR Admin Anasayfa.aspx

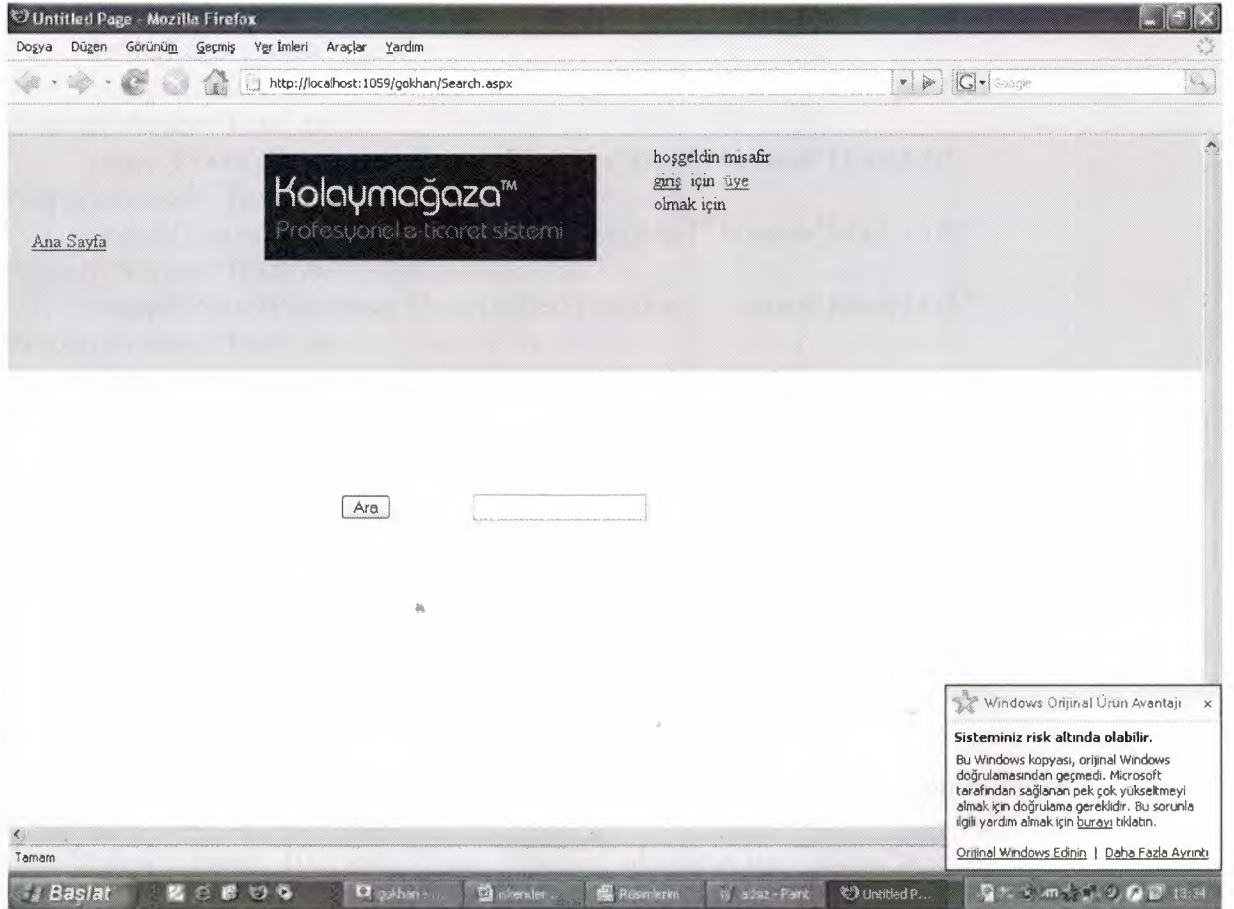
```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="AnaSayfa.aspx.cs" Inherits="AnaSayfa"
Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <br />
    &nbsp;
    <div style="z-index: 100; left: 464px; width: 100px; position: absolute; top: 288px;
    height: 100px">
    <asp:LoginView ID="LoginView1" runat="server">
        <RoleGroups>
            <asp:RoleGroup Roles="Admin">
                <ContentTemplate>
                    <a href="AdminPages/KategoriEkle.aspx">Kategori Ekle</a>
                    <br />
                    <br />
                    <a href="AdminPages/UrunEkle.aspx">Ürün Ekle</a>
                    <br />
                </ContentTemplate>
            </asp:RoleGroup>
        </RoleGroups>
    </asp:LoginView>
    </div>
</asp:Content>
```



```
using System.Web.UI.WebControls.WebParts;  
using System.Web.UI.HtmlControls;
```

```
public partial class AnaSayfa : System.Web.UI.Page  
{  
    protected void Page_Load(object sender, EventArgs e)  
    {  
  
    }  
}
```

2.3 - Search.aspx



2.3 - a) ASP CODE FOR Search.aspx

[illegible]


```

        <asp:BoundField DataField="ModelAdi" HeaderText="ModelAdi"
SortExpression="ModelAdi" />
    </Columns>
</asp:GridView>
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
</asp:Content>

```

2.3 - b) C# CODE FOR search.aspx

```

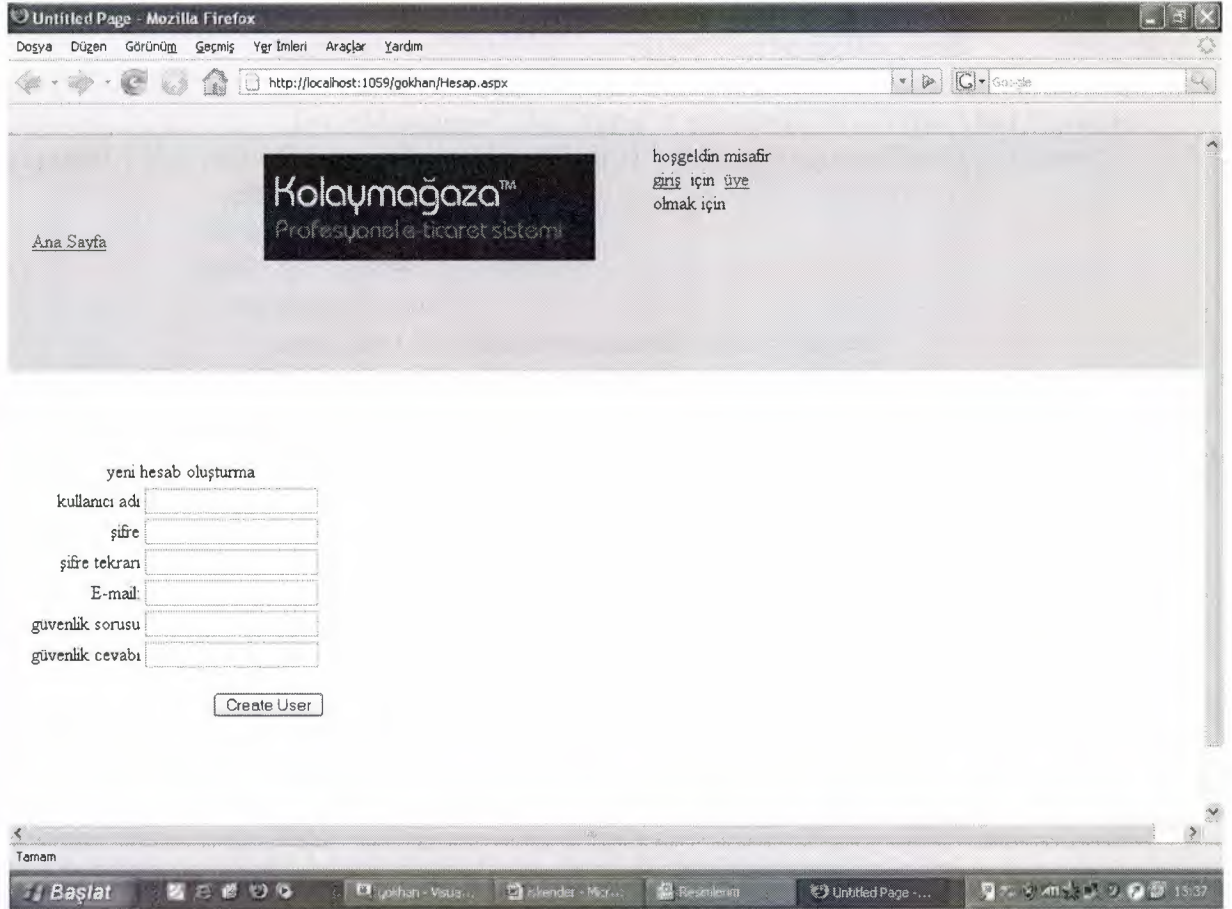
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Search : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        GridView1.DataBind();
    }
}

```

2.4 - Hesap.aspx



2.4 - a) ASP CODE FOR hesap.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="Hesap.aspx.cs" Inherits="Hesap" Title="Untitled
Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:CreateUserWizard ID="CreateUserWizard1" runat="server"
OnCreatedUser="CreateUserWizard1_CreatedUser"
    Style="z-index: 100; left: 16px; position: absolute; top: 272px">
    <WizardSteps>
        <asp:CreateUserWizardStep runat="server">
            <ContentTemplate>
                <table border="0">
                    <tr>
                        <td align="center" colspan="2" style="height: 21px">
```

```

        yeni hesap oluşturma</td>
    </tr>
    <tr>
        <td align="right">
            <asp:Label ID="UserNameLabel" runat="server"
AssociatedControlID="UserName">kullanıcı adı</asp:Label></td>
            <td>
                <asp:TextBox ID="UserName" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="UserNameRequired"
runat="server" ControlToValidate="UserName"
                ErrorMessage="User Name is required." ToolTip="User Name is
required." ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Label ID="PasswordLabel" runat="server"
AssociatedControlID="Password">şifre</asp:Label></td>
            <td>
                <asp:TextBox ID="Password" runat="server"
TextMode="Password"></asp:TextBox>
                <asp:RequiredFieldValidator ID="PasswordRequired"
runat="server" ControlToValidate="Password"
                ErrorMessage="Password is required." ToolTip="Password is
required." ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Label ID="ConfirmPasswordLabel" runat="server"
AssociatedControlID="ConfirmPassword">şifre tekrarı</asp:Label></td>
            <td>
                <asp:TextBox ID="ConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
                <asp:RequiredFieldValidator ID="ConfirmPasswordRequired"
runat="server" ControlToValidate="ConfirmPassword"
                ErrorMessage="Confirm Password is required."
                ToolTip="Confirm Password is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Label ID="EmailLabel" runat="server"
AssociatedControlID="Email">E-mail:</asp:Label></td>
            <td>
                <asp:TextBox ID="Email" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="EmailRequired" runat="server"
ControlToValidate="Email"
            </td>
        </tr>
    </table>

```

```

                ErrorMessage="E-mail is required." ToolTip="E-mail is
required." ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Label ID="QuestionLabel" runat="server"
AssociatedControlID="Question">güvenlik sorusu</asp:Label></td>
            <td>
                <asp:TextBox ID="Question" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="QuestionRequired"
runat="server" ControlToValidate="Question"
                ErrorMessage="Security question is required."
                ToolTip="Security question is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td align="right">
                <asp:Label ID="AnswerLabel" runat="server"
AssociatedControlID="Answer">güvenlik cevabı</asp:Label></td>
            <td>
                <asp:TextBox ID="Answer" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="AnswerRequired" runat="server"
ControlToValidate="Answer"
                ErrorMessage="Security answer is required." ToolTip="Security
answer is required."
                ValidationGroup="CreateUserWizard1">*</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td align="center" colspan="2">
                <asp:CompareValidator ID="PasswordCompare" runat="server"
ControlToCompare="Password"
                ControlToValidate="ConfirmPassword" Display="Dynamic"
                ErrorMessage="şifre ve şifre tekrarı uyumsuz"
                ValidationGroup="CreateUserWizard1"></asp:CompareValidator>
            </td>
        </tr>
        <tr>
            <td align="center" colspan="2" style="color: red">
                <asp:Literal ID="ErrorMessage" runat="server"
EnableViewState="False"></asp:Literal>
            </td>
        </tr>
    </table>
</ContentTemplate>

```



```

</asp:CreateUserWizardStep>
<asp:CompleteWizardStep runat="server">
  <ContentTemplate>
    <table border="0">
      <tr>
        <td align="center" colspan="2">
          tamamlandı</td>
        </tr>
      <tr>
        <td>
          hesabınız başarılı bir şekilde oluşturuldu.</td>
        </tr>
      <tr>
        <td align="right" colspan="2">
          <asp:Button ID="ContinueButton" runat="server"
CausesValidation="False" CommandName="Continue"
          PostBackUrl="~/AnaSayfa.aspx" Text="devam"
ValidationGroup="CreateUserWizard1" />
        </td>
      </tr>
    </table>
  </ContentTemplate>
</asp:CompleteWizardStep>
</WizardSteps>
</asp:CreateUserWizard>
<br />
<br />
<br />
<br />
</asp:Content>

```

2.4 - b) C# CODE FOR hesap.aspx

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Hesap : System.Web.UI.Page
{

```

```

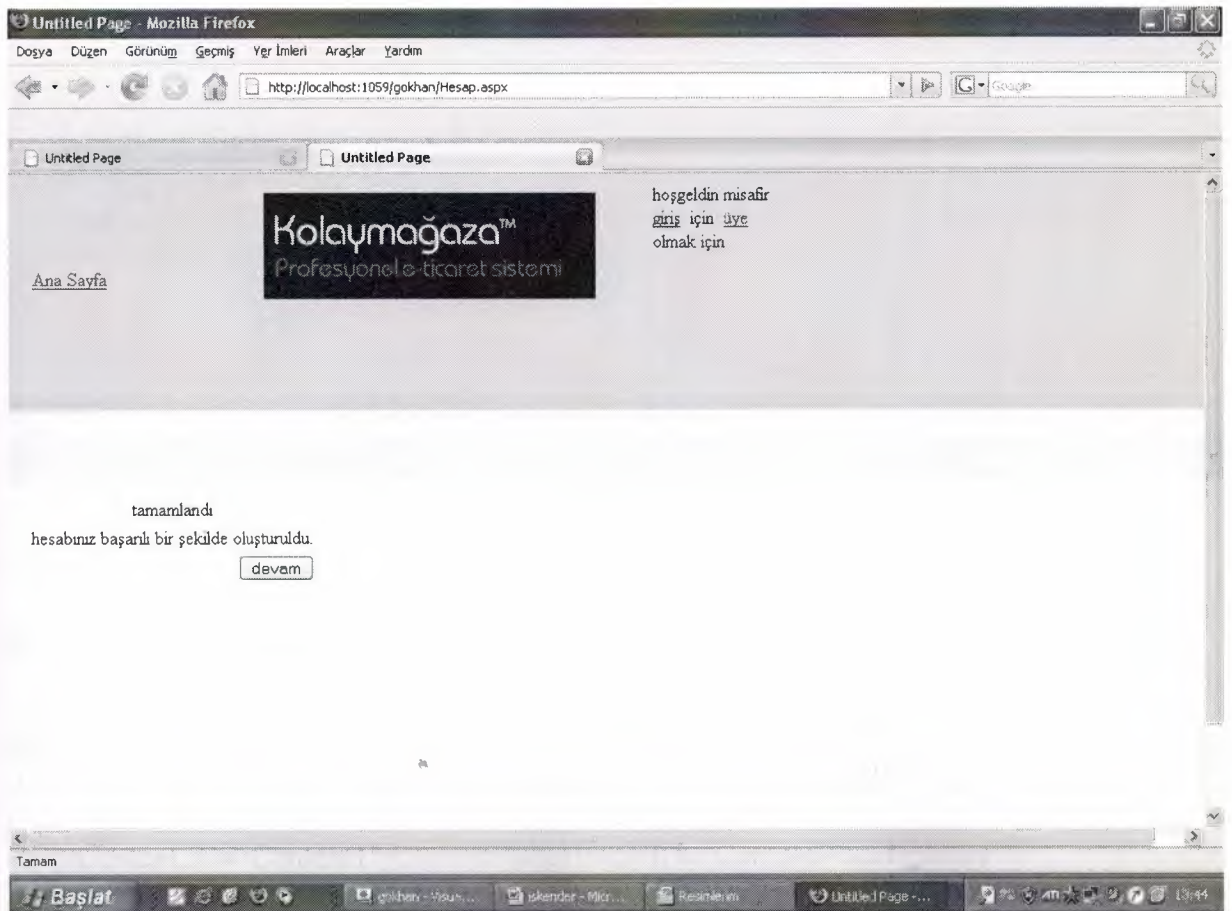
protected void Page_Load(object sender, EventArgs e)
{

}

protected void CreateUserWizard1_CreatedUser(object sender, EventArgs e)
{
    Roles.AddUserToRole(CreateUserWizard1.UserName, "SiteUye");
}
}

```

2.4 - c) Continue to hesap.aspx and Complete to create user



2.5 - Login Page



Giriş yapmak için lütfen kullanıcı adınızı ve şifrenizi doğru bir şekilde giriniz ...



2.5 - a) ASP CODE FOR Login.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="Login.aspx.cs" Inherits="Login" Title="Untitled
Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:Login ID="Login1" runat="server" BackColor="#EFF3FB"
BorderColor="#B5C7DE" BorderPadding="4"
    BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-
Size="0.8em"
    ForeColor="#333333" Style="z-index: 100; left: 40px; position: absolute; top:
360px">
        <TextBoxStyle Font-Size="0.8em" />
        <LoginButtonStyle BackColor="White" BorderColor="#507CD1"
BorderColor="Solid" BorderWidth="1px"
        Font-Names="Verdana" Font-Size="0.8em" ForeColor="#284E98" />
    </LayoutTemplate>
```



```

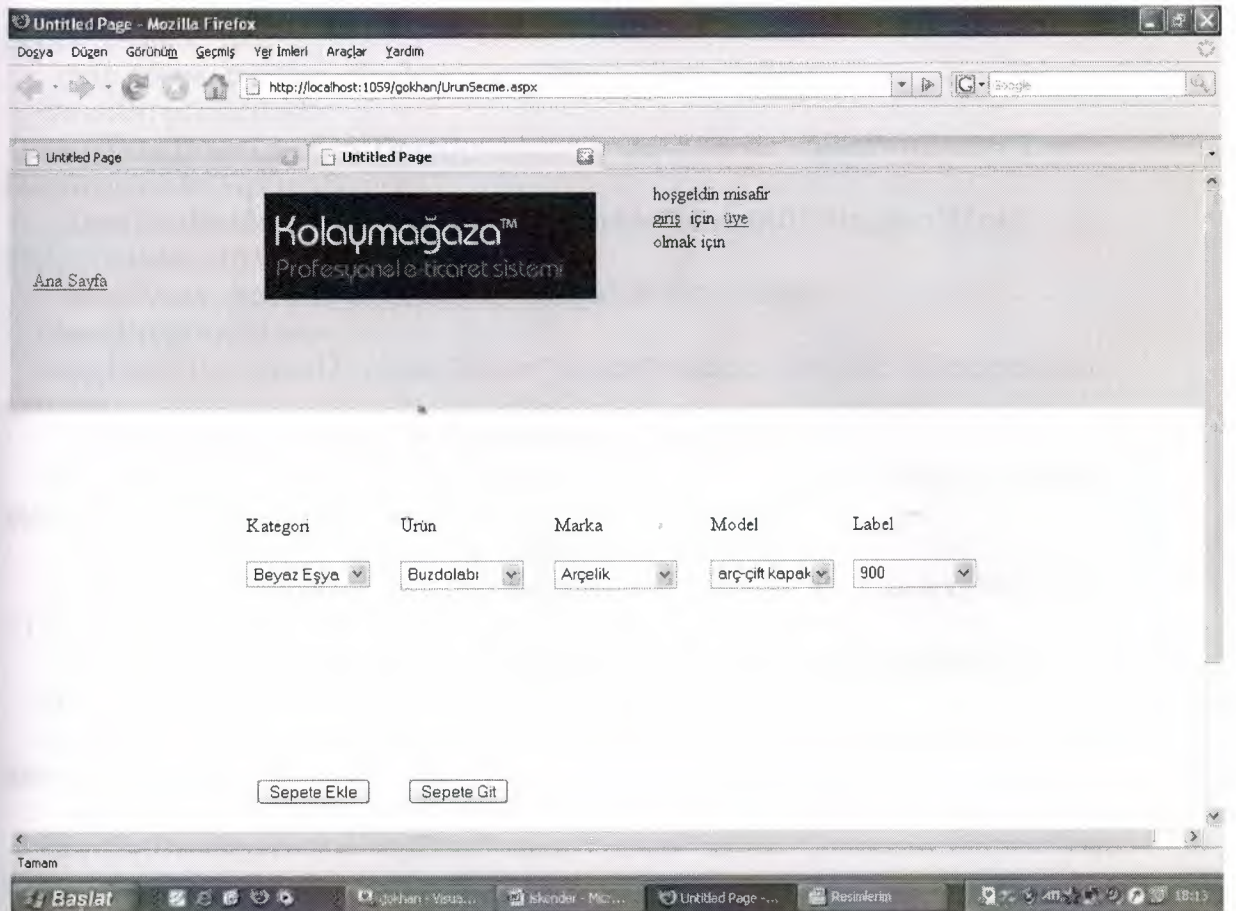
        <table border="0" cellpadding="4" cellspacing="0" style="border-collapse:
collapse">
            <tr>
                <td>
                    <table border="0" cellpadding="0">
                        <tr>
                            <td align="center" colspan="2" style="font-weight: bold; font-size:
0.9em; color: white;
                            background-color: #507cd1">
                                Log In</td>
                        </tr>
                        <tr>
                            <td align="right">
                                <asp:Label ID="UserNameLabel" runat="server"
AssociatedControlID="UserName">Kullanıcı Adı :</asp:Label></td>
                                <td>
                                    <asp:TextBox ID="UserName" runat="server" Font-
Size="0.8em"></asp:TextBox>
                                    <asp:RequiredFieldValidator ID="UserNameRequired"
runat="server" ControlToValidate="UserName"
                                    ErrorMessage="User Name is required." ToolTip="User Name
is required." ValidationGroup="Login1">*</asp:RequiredFieldValidator>
                                </td>
                            </tr>
                            <tr>
                                <td align="right">
                                    <asp:Label ID="PasswordLabel" runat="server"
AssociatedControlID="Password">Şifre :</asp:Label></td>
                                    <td>
                                        <asp:TextBox ID="Password" runat="server" Font-Size="0.8em"
TextMode="Password"></asp:TextBox>
                                        <asp:RequiredFieldValidator ID="PasswordRequired"
runat="server" ControlToValidate="Password"
                                        ErrorMessage="Password is required." ToolTip="Password is
required." ValidationGroup="Login1">*</asp:RequiredFieldValidator>
                                    </td>
                                </tr>
                                <tr>
                                    <td colspan="2">
                                        <asp:CheckBox ID="RememberMe" runat="server" Text="Beni
Hatırla" />
                                    </td>
                                </tr>
                                <tr>
                                    <td align="center" colspan="2" style="color: red">
                                        <asp:Literal ID="FailureText" runat="server"
EnableViewState="False"></asp:Literal>
                                    </td>
                                </tr>
                            </tr>
                    </td>
            </tr>

```


using System.Web.UI.HtmlControls;

```
public partial class Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

2.6 - ÜrünSeçme.aspx Page



2.6 - a) ASP CODE FOR UrunSecme.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="UrunSecme.aspx.cs" Inherits="UrunSecme"
Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource1"
    DataTextField="KategoriAdi" DataValueField="KategoriAdi" Style="z-index:
100;
    left: 200px; position: absolute; top: 328px" Width="104px">
</asp:DropDownList>
    <asp:DropDownList ID="DropDownList2" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource2"
    DataTextField="UrunAdi" DataValueField="UrunAdi" Height="22px" Style="z-
index: 101;
    left: 330px; position: absolute; top: 328px" Width="104px">
</asp:DropDownList>
    <asp:DropDownList ID="DropDownList3" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource3"
    DataTextField="MarkaAdi" DataValueField="MarkaAdi" Height="22px"
Style="z-index: 102;
    left: 460px; position: absolute; top: 328px" Width="104px">
</asp:DropDownList>
    <asp:DropDownList ID="DropDownList4" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource4"
    DataTextField="ModelAdi" DataValueField="ModelAdi" Height="22px"
Style="z-index: 103;
    left: 592px; position: absolute; top: 328px" Width="104px">
</asp:DropDownList>
    <asp:Label ID="Label2" runat="server" Style="z-index: 104; left: 330px; position:
absolute;
    top: 288px" Text="Ürün" Width="104px"></asp:Label>
    <asp:Label ID="Label3" runat="server" Style="z-index: 105; left: 460px; position:
absolute;
    top: 288px" Text="Marka" Width="104px"></asp:Label>
    <asp:Label ID="Label5" runat="server" Height="19px" Style="z-index: 106; left:
712px;
    position: absolute; top: 288px" Text="Label" Width="104px"></asp:Label>
<br />
    <asp:Label ID="Label4" runat="server" Style="z-index: 107; left: 200px; position:
absolute;
    top: 288px" Text="Kategori" Width="104px"></asp:Label>
    <asp:SqlDataSource ID="SqlDataSource4" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
```



```

        SelectCommand="SELECT [ModelAdi] FROM [DetayView] WHERE
        (([MarkaAdi] = @MarkaAdi) AND ([UrunAdi] = @UrunAdi)) ORDER BY
        [ModelAdi]">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList3" Name="MarkaAdi"
PropertyName="SelectedValue"
                Type="String" />
            <asp:ControlParameter ControlID="DropDownList2" Name="UrunAdi"
PropertyName="SelectedValue"
                Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <br />
    <asp:Label ID="Label1" runat="server" Style="z-index: 108; left: 592px; position:
absolute;
        top: 288px" Text="Model" Width="104px"></asp:Label>
    <br />
    <br />
    <asp:SqlDataSource ID="SqlDataSource3" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
        SelectCommand="SELECT DISTINCT [MarkaAdi] FROM [DetayView]
WHERE ([UrunAdi] = @UrunAdi) ORDER BY [MarkaAdi]">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList2" Name="UrunAdi"
PropertyName="SelectedValue"
                Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <br />
    <br />
    <asp:SqlDataSource ID="SqlDataSource2" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
        SelectCommand="SELECT DISTINCT [UrunAdi] FROM [DetayView] WHERE
([KategoriAdi] = @KategoriAdi) ORDER BY [UrunAdi]">
        <SelectParameters>
            <asp:ControlParameter ControlID="DropDownList1" Name="KategoriAdi"
PropertyName="SelectedValue"
                Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
    <br />
    <br />
    <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
        SelectCommand="SELECT DISTINCT [KategoriAdi] FROM [DetayView]
ORDER BY [KategoriAdi]">
    </asp:SqlDataSource>
    <br />
    <asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False"
DataKeyNames="ModelAdi"

```

```

DataSourceID="SqlDataSource5" Height="50px" Style="z-index: 109; left: 200px;
position: absolute; top: 392px" Width="504px">
<Fields>
    <asp:BoundField DataField="ModelAdi" HeaderText="ModelAdi"
ReadOnly="True" SortExpression="ModelAdi" />
    <asp:BoundField DataField="ModelOzellikleri"
HeaderText="ModelOzellikleri" SortExpression="ModelOzellikleri" />
    <asp:BoundField DataField="ModelFiyati" HeaderText="ModelFiyati"
SortExpression="ModelFiyati" />
    <asp:BoundField DataField="Taksit_ay" HeaderText="Taksit_ay"
SortExpression="Taksit_ay" />
</Fields>
</asp:DetailsView>
<asp:Button ID="Button2" runat="server" Style="z-index: 110; left: 208px; position:
absolute;
    top: 512px" Text="Sepete Ekle" OnClick="Button2_Click" />
<asp:DropDownList ID="DropDownList5" runat="server" AutoPostBack="True"
DataSourceID="SqlDataSource7"
    DataTextField="Fiyat" DataValueField="Fiyat" Style="z-index: 111; left: 712px;
    position: absolute; top: 328px" Height="22px" Width="104px">
</asp:DropDownList>
<br />
<br />
<asp:SqlDataSource ID="SqlDataSource5" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
    SelectCommand="SELECT * FROM [ModelOzellik] WHERE ([ModelAdi] =
@ModelAdi) ORDER BY [ModelAdi]"
    OldValuesParameterFormatString="original_{0}">
    <SelectParameters>
        <asp:ControlParameter ControlID="DropDownList4" Name="ModelAdi"
PropertyName="SelectedValue"
            Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
<asp:Button ID="Button3" runat="server" PostBackUrl="~/Sepet.aspx" Style="z-
index: 114;
    left: 336px; position: absolute; top: 512px" Text="Sepete Git" />
<br />
<br />
<asp:SqlDataSource ID="SqlDataSource6" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
    SelectCommand="SELECT * FROM Model" UpdateCommand="UPDATE
Model SET StokMiktari = StokMiktari - 1 WHERE (ModelAdi = @ModelAdi)">
    <UpdateParameters>
        <asp:ControlParameter ControlId="DropDownList4" Name="ModelAdi"
PropertyName="SelectedValue" Type="String" />
    </UpdateParameters>
</asp:SqlDataSource>
&nbsp;&nbsp;&nbsp;<br />
&nbsp;&nbsp;&nbsp;<br />

```

2.6 – b) C# CODE FOR UrunSecme.aspx

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

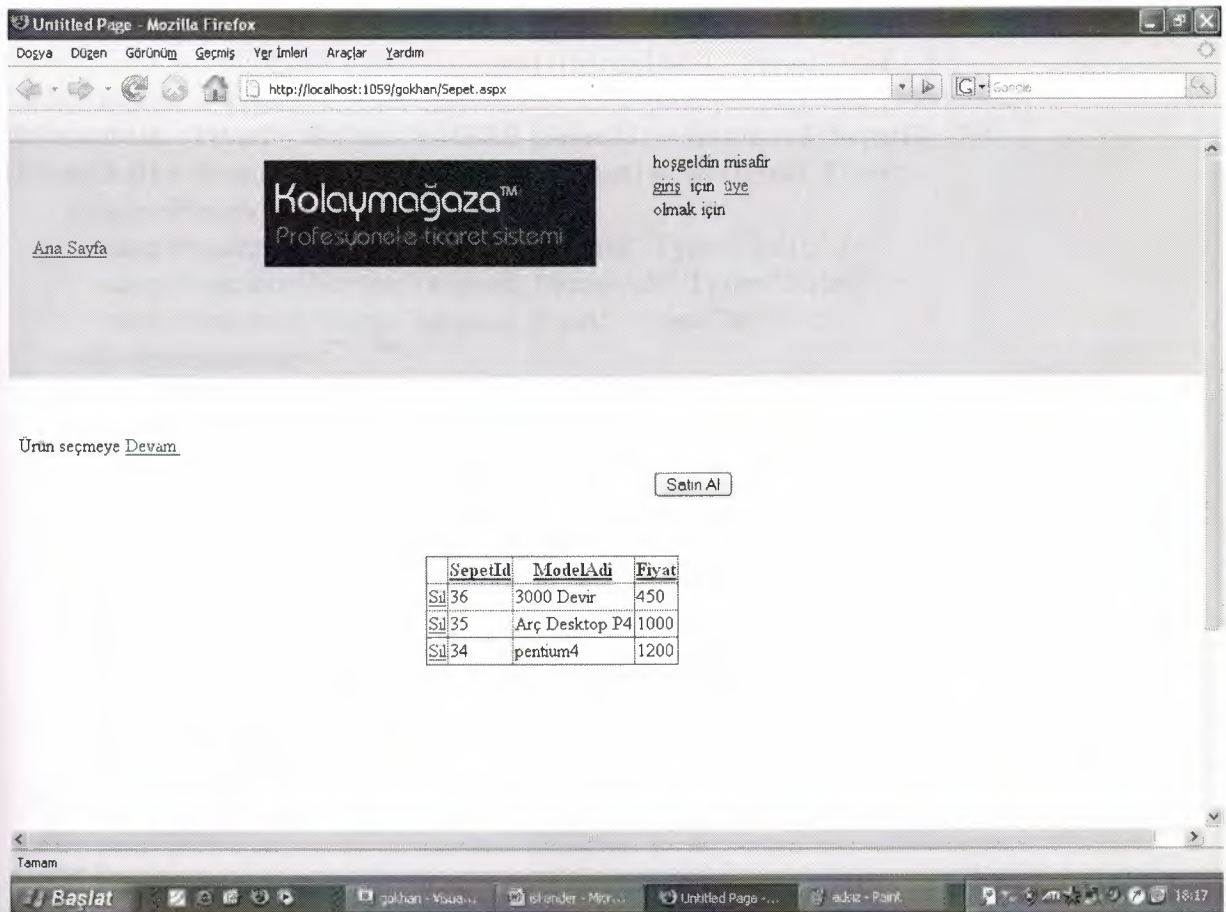
public partial class UrunSecme : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        SqlDataSource6.Update();
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        SqlDataSource8.Insert();
    }
}
```

2.7 - Sepet PAGE



2.7 - a) ASP CODE FOR Sepet.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"  
AutoEventWireup="true" CodeFile="Sepet.aspx.cs" Inherits="Sepet" Title="Untitled  
Page" %>  
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceholder1"  
Runat="Server">  
    &nbsp;&nbsp;&nbsp;<br />  
    Ürün seçmeye <a href="UrunSecme.aspx">Devam&nbsp;&nbsp;&</a><br />  
    &nbsp;&nbsp;& &nbsp;&nbsp;&
```



```

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%=
ConnectionStrings:MyConnection %>"
    OldValuesParameterFormatString="original_{0}" SelectCommand="SELECT *
FROM [Sepet] ORDER BY [ModelAdi]" ConflictDetection="CompareAllValues"
DeleteCommand="DELETE FROM [Sepet] WHERE [SepetId] = @original_SepetId
AND [ModelAdi] = @original_ModelAdi AND [Fiyat] = @original_Fiyat"
InsertCommand="INSERT INTO [Sepet] ([ModelAdi], [Fiyat]) VALUES
(@ModelAdi, @Fiyat)" UpdateCommand="UPDATE [Sepet] SET [ModelAdi] =
@ModelAdi, [Fiyat] = @Fiyat WHERE [SepetId] = @original_SepetId AND
[ModelAdi] = @original_ModelAdi AND [Fiyat] = @original_Fiyat">
    <DeleteParameters>
        <asp:Parameter Name="original_SepetId" Type="Int16" />
        <asp:Parameter Name="original_ModelAdi" Type="String" />
        <asp:Parameter Name="original_Fiyat" Type="Int32" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="ModelAdi" Type="String" />
        <asp:Parameter Name="Fiyat" Type="Int32" />
        <asp:Parameter Name="original_SepetId" Type="Int16" />
        <asp:Parameter Name="original_ModelAdi" Type="String" />
        <asp:Parameter Name="original_Fiyat" Type="Int32" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="ModelAdi" Type="String" />
        <asp:Parameter Name="Fiyat" Type="Int32" />
    </InsertParameters>
</asp:SqlDataSource>
&nbsp; &nbsp;
<br />
<br />
<br />
<br />
<br />
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
AllowSorting="True"
    AutoGenerateColumns="False" DataKeyNames="SepetId"
DataSourceID="SqlDataSource1"
    Style="z-index: 100; left: 352px; position: absolute; top: 352px">
    <Columns>
        <asp:CommandField ShowDeleteButton="True" DeleteText="Sil" />
        <asp:BoundField DataField="SepetId" HeaderText="SepetId"
InsertVisible="False" ReadOnly="True"
            SortExpression="SepetId" />
        <asp:BoundField DataField="ModelAdi" HeaderText="ModelAdi"
SortExpression="ModelAdi" />
        <asp:BoundField DataField="Fiyat" HeaderText="Fiyat"
SortExpression="Fiyat" />
    </Columns>
</asp:GridView>
<br />

```



```

{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}

```

2.8 - SatınAl PAGE

Untitled Page - Mozilla Firefox

Doğya Düzen Görünüm Geçmiş Yer İmleri Araçlar Yardım

http://localhost:1059/gokhan/SatinAl.aspx

Google

Untitled Page

Untitled Page

Lütfen Gerekli Alanları Doldurun ...

İsim ve Soyisim

Adres

Kart Tipi

Kart Numarası

CV Numarası

MasterCard

VISA

Satın Al

Tamam

Başlat

gokhan - Visual Web ...

Ekranlar - Microsoft ...

Untitled Page - Mozill...

18:25

2.8 - a) ASP CODE FOR SatınAl.aspx

```

<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="SatinAl.aspx.cs" Inherits="SatinAl"
Title="Untitled Page" %>

```



```

<br />
<br />
<br />
<br />
<br />
<br />
<asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Style="z-index:
111;
    left: 560px; position: absolute; top: 608px" Text="Satın Al" />
<br />
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%%$
ConnectionStrings:MyConnection %>"
    DeleteCommand="DELETE FROM Sepet" SelectCommand="SELECT Sepet.*
FROM Sepet" UpdateCommand="UPDATE Model SET StokMiktari = StokMiktari - 1
WHERE (ModelId IN (SELECT Model.ModelId FROM Sepet))">
</asp:SqlDataSource>
<br />
<br />
<br />
</asp:Content>

```

2.8 - b) CODE C# FOR SatınAl.aspx

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class SatınAl : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        SqlDataSource1.Update();
    }
}

```



```
        SqlDataSource1.Delete();  
    }  
}
```

CHAPTER 3 : ONLY ADMIN PAGES

3.1 - PAGE KategoriEkle.aspx



3.1 - a) ASP CODE FOR KategoriEkle.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="KategoriEkle.aspx.cs" Inherits="UrunEkle"
Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="True"
AutoGenerateRows="False"
```



```

        <asp:Parameter Name="KategoriAdi" Type="String" />
    </InsertParameters>
</asp:SqlDataSource>
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
</asp:Content>

```

3.1 - b) C# CODE FOR KategoriEkle.aspx

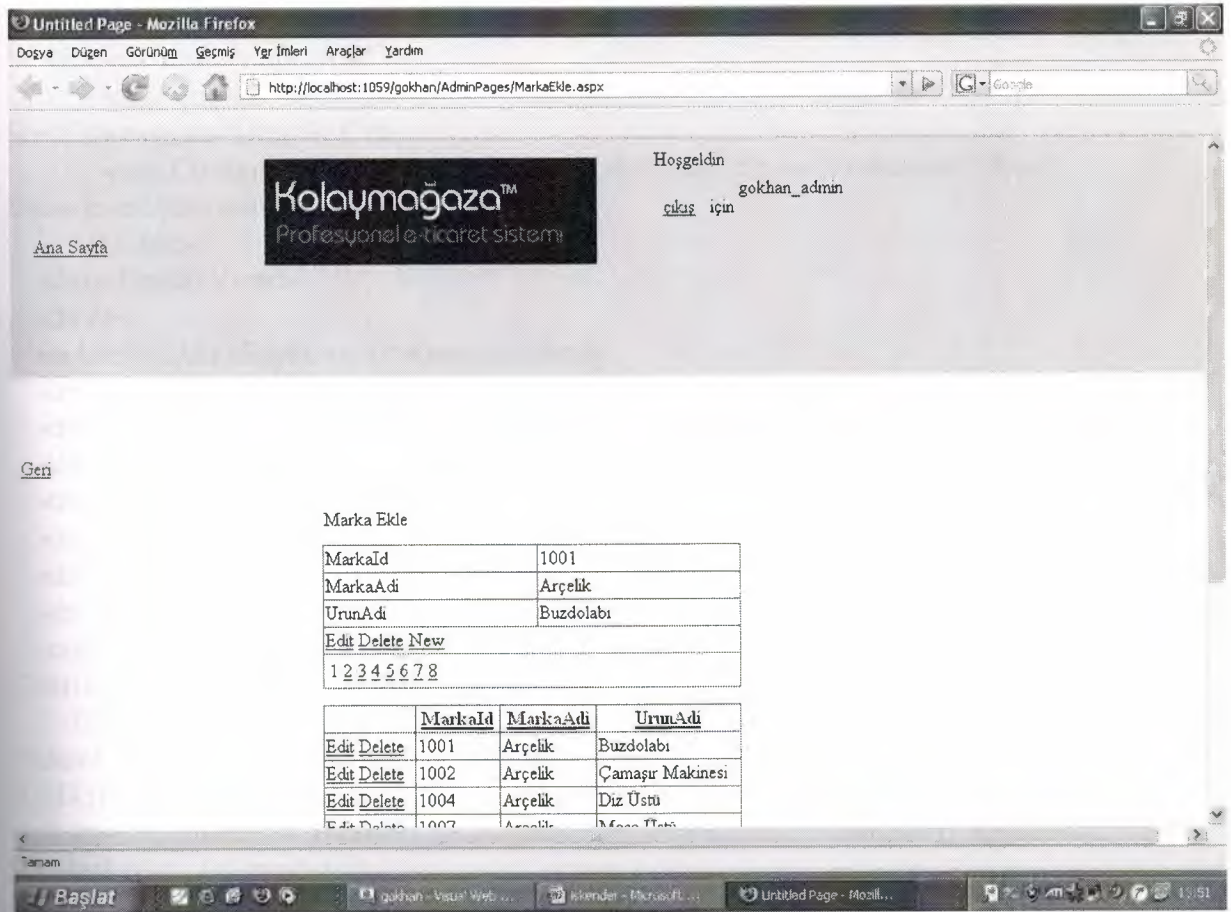
```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class UrunEkle : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // ...
    }
}

```

3.2 - PAGE MarkaEkle.aspx



3.2 - a) ASP CODE FOR MarkaEkle.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="MarkaEkle.aspx.cs"
Inherits="AdminPages_MarkaEkle" Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    &nbsp; &nbsp; &nbsp; &nbsp;
    <asp:Label ID="Label1" runat="server" Style="z-index: 100; left: 264px; position:
absolute;
    top: 312px" Text="Marka Ekle"></asp:Label>
    <asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="True"
AutoGenerateRows="False"
    DataKeyNames="MarkaId" DataSourceID="SqlDataSource1" Height="50px"
    Style="z-index: 101;
    left: 264px; position: absolute; top: 344px" Width="352px">
    <Fields>
```

```

        <asp:BoundField DataField="MarkaId" HeaderText="MarkaId"
InsertVisible="False" ReadOnly="True"
        SortExpression="MarkaId" />
        <asp:BoundField DataField="MarkaAdi" HeaderText="MarkaAdi"
SortExpression="MarkaAdi" />
        <asp:BoundField DataField="UrunAdi" HeaderText="UrunAdi"
SortExpression="UrunAdi" />
        <asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
ShowInsertButton="True" />
    </Fields>
</asp:DetailsView>
<br />
<a href=" ../AnaSayfa.aspx ">Geri</a><br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConflictDetection="CompareAllValues"
    ConnectionString="<%= $ ConnectionStrings:MyConnection %>"
DeleteCommand="DELETE FROM [Marka] WHERE [MarkaId] = @original_MarkaId
AND [MarkaAdi] = @original_MarkaAdi AND [UrunAdi] = @original_UrunAdi"
InsertCommand="INSERT INTO [Marka] ([MarkaAdi], [UrunAdi]) VALUES
(@MarkaAdi, @UrunAdi)"
OldValuesParameterFormatString="original_{0}" SelectCommand="SELECT *
FROM [Marka] ORDER BY [MarkaAdi]"
UpdateCommand="UPDATE [Marka] SET [MarkaAdi] = @MarkaAdi, [UrunAdi]
= @UrunAdi WHERE [MarkaId] = @original_MarkaId AND [MarkaAdi] =
@original_MarkaAdi AND [UrunAdi] = @original_UrunAdi">
    <DeleteParameters>
        <asp:Parameter Name="original_MarkaId" Type="Int16" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
        <asp:Parameter Name="original_MarkaId" Type="Int16" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
    </InsertParameters>
</asp:SqlDataSource>
<br />

```



```

<br />
<br />
<br />
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
AllowSorting="True"
    AutoGenerateColumns="False" DataKeyNames="MarkaId"
DataSourceID="SqlDataSource2"
    Style="z-index: 103; left: 264px; position: absolute; top: 480px" Width="352px">
    <Columns>
        <asp:CommandField ShowDeleteButton="True" ShowEditButton="True" />
        <asp:BoundField DataField="MarkaId" HeaderText="MarkaId"
InsertVisible="False" ReadOnly="True"
            SortExpression="MarkaId" />
        <asp:BoundField DataField="MarkaAdi" HeaderText="MarkaAdi"
SortExpression="MarkaAdi" />
        <asp:BoundField DataField="UrunAdi" HeaderText="UrunAdi"
SortExpression="UrunAdi" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
ConflictDetection="CompareAllValues"
    ConnectionString="<%= $ConnectionStrings:MyConnection %>"
DeleteCommand="DELETE FROM [Marka] WHERE [MarkaId] = @original_MarkaId
AND [MarkaAdi] = @original_MarkaAdi AND [UrunAdi] = @original_UrunAdi"
    InsertCommand="INSERT INTO [Marka] ([MarkaAdi], [UrunAdi]) VALUES
(@MarkaAdi, @UrunAdi)"
    OldValuesParameterFormatString="original_{0}" SelectCommand="SELECT *
FROM [Marka] ORDER BY [MarkaAdi]"
    UpdateCommand="UPDATE [Marka] SET [MarkaAdi] = @MarkaAdi, [UrunAdi]
= @UrunAdi WHERE [MarkaId] = @original_MarkaId AND [MarkaAdi] =
@original_MarkaAdi AND [UrunAdi] = @original_UrunAdi">
    <DeleteParameters>
        <asp:Parameter Name="original_MarkaId" Type="Int16" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
        <asp:Parameter Name="original_MarkaId" Type="Int16" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
    </InsertParameters>
</asp:SqlDataSource>
<br />
<br />

```

```
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
<br />
</asp:Content>
```

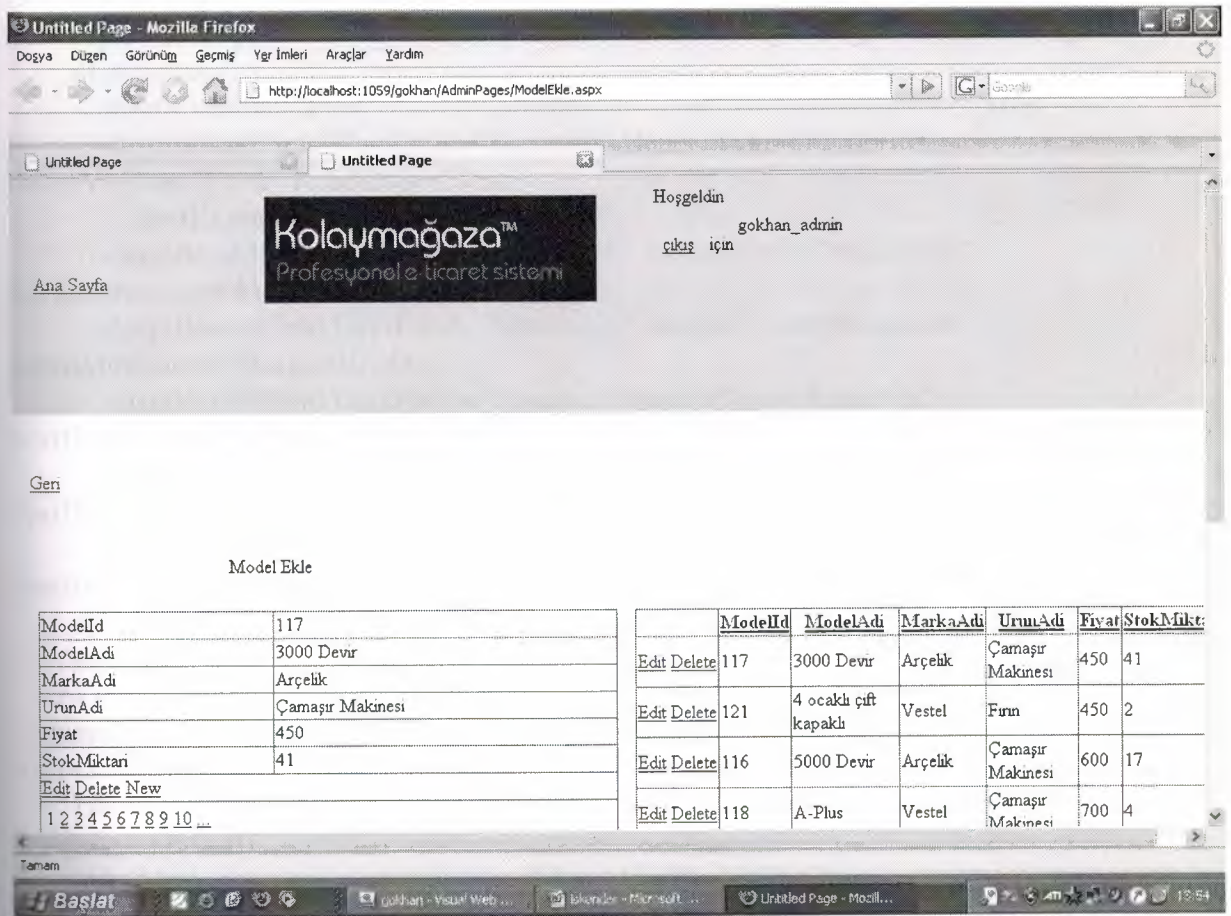
3.2 - b) C# CODE FOR MarkaEkle.aspx

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class AdminPages_MarkaEkle : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

3.3 - PAGE ModelEkle



3.3 – a) ASP CODE FOR ModelEkle.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="ModelEkle.aspx.cs"
Inherits="AdminPages_ModelEkle" Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    &nbsp; &nbsp; &nbsp; &nbsp;
    <asp:Label ID="Label1" runat="server" Style="z-index: 100; left: 184px; position:
absolute;
    top: 320px" Text="Model Ekle"></asp:Label>
    &nbsp;<br />
    &nbsp;<a href=" ../AnaSayfa.aspx">Geri</a><br />
<br />
<br />
<br />
<br />
```

```

<br />
<br />
<br />
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
AllowSorting="True"
    AutoGenerateColumns="False" DataKeyNames="ModelId"
DataSourceID="SqlDataSource2"
    Style="z-index: 103; left: 528px; position: absolute; top: 368px" Width="496px">
    <Columns>
        <asp:CommandField ShowDeleteButton="True" ShowEditButton="True" />
        <asp:BoundField DataField="ModelId" HeaderText="ModelId"
InsertVisible="False" ReadOnly="True"
    SortExpression="ModelId" />
        <asp:BoundField DataField="ModelAdi" HeaderText="ModelAdi"
SortExpression="ModelAdi" />
        <asp:BoundField DataField="MarkaAdi" HeaderText="MarkaAdi"
SortExpression="MarkaAdi" />
        <asp:BoundField DataField="UrunAdi" HeaderText="UrunAdi"
SortExpression="UrunAdi" />
        <asp:BoundField DataField="Fiyat" HeaderText="Fiyat"
SortExpression="Fiyat" />
        <asp:BoundField DataField="StokMiktari" HeaderText="StokMiktari"
SortExpression="StokMiktari" />
    </Columns>
</asp:GridView>
<br />
<asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="True"
AutoGenerateRows="False"
    DataKeyNames="ModelId" DataSourceID="SqlDataSource1" Height="50px"
Style="z-index: 102;
    left: 24px; position: absolute; top: 368px" Width="488px">
    <Fields>
        <asp:BoundField DataField="ModelId" HeaderText="ModelId"
InsertVisible="False" ReadOnly="True"
    SortExpression="ModelId" />
        <asp:BoundField DataField="ModelAdi" HeaderText="ModelAdi"
SortExpression="ModelAdi" />
        <asp:BoundField DataField="MarkaAdi" HeaderText="MarkaAdi"
SortExpression="MarkaAdi" />
        <asp:BoundField DataField="UrunAdi" HeaderText="UrunAdi"
SortExpression="UrunAdi" />
        <asp:BoundField DataField="Fiyat" HeaderText="Fiyat"
SortExpression="Fiyat" />
        <asp:BoundField DataField="StokMiktari" HeaderText="StokMiktari"
SortExpression="StokMiktari" />
        <asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
ShowInsertButton="True" />
    </Fields>
</asp:DetailsView>
<br />

```



```

<br />
<br />
    &nbsp;<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConflictDetection="CompareAllValues"
    ConnectionString="<%"$ ConnectionStrings:MyConnection %">
DeleteCommand="DELETE FROM [Model] WHERE [ModelId] = @original_ModelId
AND [ModelAdi] = @original_ModelAdi AND [MarkaAdi] = @original_MarkaAdi
AND [UrunAdi] = @original_UrunAdi AND [Fiyat] = @original_Fiyat AND
[StokMiktari] = @original_StokMiktari"
    InsertCommand="INSERT INTO [Model] ([ModelAdi], [MarkaAdi], [UrunAdi],
[Fiyat], [StokMiktari]) VALUES (@ModelAdi, @MarkaAdi, @UrunAdi, @Fiyat,
@StokMiktari)"
    OldValuesParameterFormatString="original_{0}" SelectCommand="SELECT *
FROM [Model] ORDER BY [ModelAdi]"
    UpdateCommand="UPDATE [Model] SET [ModelAdi] = @ModelAdi,
[MarkaAdi] = @MarkaAdi, [UrunAdi] = @UrunAdi, [Fiyat] = @Fiyat, [StokMiktari] =
@StokMiktari WHERE [ModelId] = @original_ModelId AND [ModelAdi] =
@original_ModelAdi AND [MarkaAdi] = @original_MarkaAdi AND [UrunAdi] =
@original_UrunAdi AND [Fiyat] = @original_Fiyat AND [StokMiktari] =
@original_StokMiktari">
    <DeleteParameters>
        <asp:Parameter Name="original_ModelId" Type="Int16" />
        <asp:Parameter Name="original_ModelAdi" Type="String" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
        <asp:Parameter Name="original_Fiyat" Type="Int32" />
        <asp:Parameter Name="original_StokMiktari" Type="Int16" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="ModelAdi" Type="String" />
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
        <asp:Parameter Name="Fiyat" Type="Int32" />
        <asp:Parameter Name="StokMiktari" Type="Int16" />
        <asp:Parameter Name="original_ModelId" Type="Int16" />
        <asp:Parameter Name="original_ModelAdi" Type="String" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
        <asp:Parameter Name="original_Fiyat" Type="Int32" />
        <asp:Parameter Name="original_StokMiktari" Type="Int16" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="ModelAdi" Type="String" />
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
        <asp:Parameter Name="Fiyat" Type="Int32" />
        <asp:Parameter Name="StokMiktari" Type="Int16" />
    </InsertParameters>
</asp:SqlDataSource>
&nbsp;&nbsp;&nbsp;

```

```

<asp:SqlDataSource ID="SqlDataSource2" runat="server"
ConflictDetection="CompareAllValues"
    ConnectionString="<%= $ ConnectionStrings:MyConnection %>"
DeleteCommand="DELETE FROM [Model] WHERE [ModelId] = @original_ModelId
AND [ModelAdi] = @original_ModelAdi AND [MarkaAdi] = @original_MarkaAdi
AND [UrunAdi] = @original_UrunAdi AND [Fiyat] = @original_Fiyat AND
[StokMiktari] = @original_StokMiktari"
    InsertCommand="INSERT INTO [Model] ([ModelAdi], [MarkaAdi], [UrunAdi],
[Fiyat], [StokMiktari]) VALUES (@ModelAdi, @MarkaAdi, @UrunAdi, @Fiyat,
@StokMiktari)"
    OldValuesParameterFormatString="original_{0}" SelectCommand="SELECT *
FROM [Model] ORDER BY [ModelAdi]"
    UpdateCommand="UPDATE [Model] SET [ModelAdi] = @ModelAdi,
[MarkaAdi] = @MarkaAdi, [UrunAdi] = @UrunAdi, [Fiyat] = @Fiyat, [StokMiktari] =
@StokMiktari WHERE [ModelId] = @original_ModelId AND [ModelAdi] =
@original_ModelAdi AND [MarkaAdi] = @original_MarkaAdi AND [UrunAdi] =
@original_UrunAdi AND [Fiyat] = @original_Fiyat AND [StokMiktari] =
@original_StokMiktari">
    <DeleteParameters>
        <asp:Parameter Name="original_ModelId" Type="Int16" />
        <asp:Parameter Name="original_ModelAdi" Type="String" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
        <asp:Parameter Name="original_Fiyat" Type="Int32" />
        <asp:Parameter Name="original_StokMiktari" Type="Int16" />
    </DeleteParameters>
    <UpdateParameters>
        <asp:Parameter Name="ModelAdi" Type="String" />
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
        <asp:Parameter Name="Fiyat" Type="Int32" />
        <asp:Parameter Name="StokMiktari" Type="Int16" />
        <asp:Parameter Name="original_ModelId" Type="Int16" />
        <asp:Parameter Name="original_ModelAdi" Type="String" />
        <asp:Parameter Name="original_MarkaAdi" Type="String" />
        <asp:Parameter Name="original_UrunAdi" Type="String" />
        <asp:Parameter Name="original_Fiyat" Type="Int32" />
        <asp:Parameter Name="original_StokMiktari" Type="Int16" />
    </UpdateParameters>
    <InsertParameters>
        <asp:Parameter Name="ModelAdi" Type="String" />
        <asp:Parameter Name="MarkaAdi" Type="String" />
        <asp:Parameter Name="UrunAdi" Type="String" />
        <asp:Parameter Name="Fiyat" Type="Int32" />
        <asp:Parameter Name="StokMiktari" Type="Int16" />
    </InsertParameters>
</asp:SqlDataSource>
<br />
<br />
<br />

```



```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class AdminPages_ModelEkle : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}

```

3.4 - PAGE UrunEkle.aspx

Hoşgeldin
gokhan_admin
çıkış için

Ana Sayfa

Ürün Düzenle

UrunId	1000
UrunAdi	Buzdolabı
KategoriAdi	Beyaz Eşya
Edit Delete New	
1 2 3 4 5	

	UrunId	UrunAdi	KategoriAdi
Edit Delete Select	1000	Buzdolabı	Beyaz Eşya
Edit Delete Select	1002	Çamaşır Makinesi	Beyaz Eşya
Edit Delete Select	1004	Diz Üstü	Elektronik
Edit Delete Select	1003	Fırın	Beyaz Eşya
Edit Delete Select	1005	Masa Üstü	Elektronik

Tamam

Başlat

3.4 - a) ASP CODE FOR UrunEkle.aspx

```
<%@ Page Language="C#" MasterPageFile="~/BaşSayfa.master"
AutoEventWireup="true" CodeFile="UrunEkle.aspx.cs"
Inherits="AdminPages_UrunEkle" Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:DetailsView ID="DetailsView1" runat="server" AllowPaging="True"
DataSourceID="SqlDataSource1"
    Height="50px" Style="z-index: 100; left: 120px; position: absolute; top: 344px"
    Width="256px">
        <Fields>
            <asp:CommandField ShowDeleteButton="True" ShowEditButton="True"
ShowInsertButton="True" />
        </Fields>
    </asp:DetailsView>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConflictDetection="CompareAllValues"
    ConnectionString="<%= $ConnectionStrings:MyConnection %>"
DeleteCommand="DELETE FROM [Urun] WHERE [UrunId] = @original_UrunId
AND [UrunAdi] = @original_UrunAdi AND [KategoriAdi] = @original_KategoriAdi"
    InsertCommand="INSERT INTO [Urun] ([UrunAdi], [KategoriAdi]) VALUES
(@UrunAdi, @KategoriAdi)"
    OldValuesParameterFormatString="original_{0}" SelectCommand="SELECT *
FROM [Urun] ORDER BY [UrunAdi]"
    UpdateCommand="UPDATE [Urun] SET [UrunAdi] = @UrunAdi, [KategoriAdi]
= @KategoriAdi WHERE [UrunId] = @original_UrunId AND [UrunAdi] =
@original_UrunAdi AND [KategoriAdi] = @original_KategoriAdi">
        <DeleteParameters>
            <asp:Parameter Name="original_UrunId" Type="Int16" />
            <asp:Parameter Name="original_UrunAdi" Type="String" />
            <asp:Parameter Name="original_KategoriAdi" Type="String" />
        </DeleteParameters>
        <UpdateParameters>
            <asp:Parameter Name="UrunAdi" Type="String" />
            <asp:Parameter Name="KategoriAdi" Type="String" />
            <asp:Parameter Name="original_UrunId" Type="Int16" />
            <asp:Parameter Name="original_UrunAdi" Type="String" />
            <asp:Parameter Name="original_KategoriAdi" Type="String" />
        </UpdateParameters>
        <InsertParameters>
            <asp:Parameter Name="UrunAdi" Type="String" />
            <asp:Parameter Name="KategoriAdi" Type="String" />
        </InsertParameters>
    </asp:SqlDataSource>
```



```

UpdateCommand="UPDATE [Urun] SET [UrunAdi] = @UrunAdi, [KategoriAdi]
= @KategoriAdi WHERE [UrunId] = @original_UrunId AND [UrunAdi] =
@original_UrunAdi AND [KategoriAdi] = @original_KategoriAdi">
<DeleteParameters>
  <asp:Parameter Name="original_UrunId" Type="Int16" />
  <asp:Parameter Name="original_UrunAdi" Type="String" />
  <asp:Parameter Name="original_KategoriAdi" Type="String" />
</DeleteParameters>
<UpdateParameters>
  <asp:Parameter Name="UrunAdi" Type="String" />
  <asp:Parameter Name="KategoriAdi" Type="String" />
  <asp:Parameter Name="original_UrunId" Type="Int16" />
  <asp:Parameter Name="original_UrunAdi" Type="String" />
  <asp:Parameter Name="original_KategoriAdi" Type="String" />
</UpdateParameters>
<InsertParameters>
  <asp:Parameter Name="UrunAdi" Type="String" />
  <asp:Parameter Name="KategoriAdi" Type="String" />
</InsertParameters>
</asp:SqlDataSource>
<br />
<br />
<br />
<br />
<br />
<br />
</asp:Content>

```

3.4 - b) C# CODE FOR UrunEkle.aspx

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class AdminPages_UrunEkle : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

```

}
}

CHAPTER 4 : CONFIGURATION PAGES

4.1 - CONFIGURATION FOR ADMIN PAGES

4.1 - a) PAGE Web.config

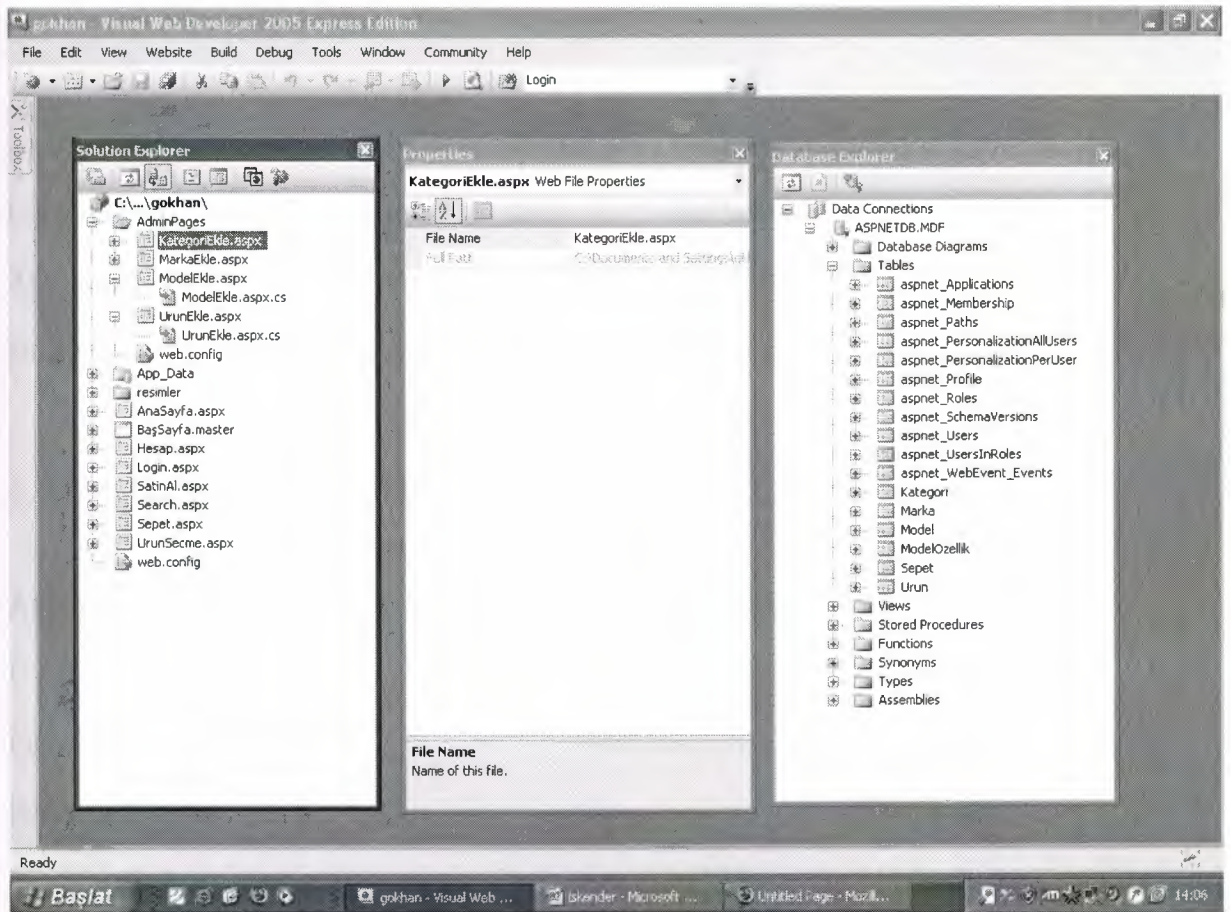
```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authorization>
      <deny roles="SiteUye" />
      <deny users="?" />
    </authorization>
  </system.web>
</configuration>
```

4.2 - CONFIGURATION FOR ALL USERS

4.2 - a) PAGE web.config

```
<?xml version="1.0"?>
<configuration>
  <connectionStrings>
    <add name="MyConnection" connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\ASPNETDB.MDF;Integr
ated Security=True;User Instance=True" providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <system.web>
    <authorization>
      <allow roles="Admin"/>
    </authorization>
    <roleManager enabled="true"/>
    <authentication mode="Forms"/>
    <compilation debug="true"/></system.web>
</configuration>
```

CHAPTER 5 : WORK ENVIRONMENT



VISUAL WEB DEVELOPER 2005 FOR INTERFACE

AND

C# AND ASP FOR PROGRAMMING

AND

SQLSERVER 2005 FOR DATABASE

CHAPTER 6 : TABLES

6.1 - APPLICATION TABLE

ApplicationName	nvarchar(256)	Unchecked
LoweredApplicationName	nvarchar(256)	Unchecked
ApplicationId	uniqueidentifier	Unchecked
Description	nvarchar(256)	Checked

6.2 - MEMBERSHIP TABLE

ApplicationId	uniqueidentifier	Unchecked
UserId	uniqueidentifier	Unchecked
Password	nvarchar(128)	Unchecked
PasswordFormat	int	Unchecked
PasswordSalt	nvarchar(128)	Unchecked
MobilePIN	nvarchar(16)	Checked
Email	nvarchar(256)	Checked
LoweredEmail	nvarchar(256)	Checked
PasswordQuestion	nvarchar(256)	Checked
PasswordAnswer	nvarchar(128)	Checked
IsApproved	bit	Unchecked
IsLockedOut	bit	Unchecked
CreateDate	datetime	Unchecked
LastLoginDate	datetime	Unchecked
LastPasswordChangedDate	datetime	Unchecked
LastLockoutDate	datetime	Unchecked
FailedPasswordAttemptCount	int	Unchecked

FailedPasswordAttemptWindowStart	datetime	Unchecked
FailedPasswordAnswerAttemptCount	int	Unchecked
FailedPasswordAnswerAttemptWindowStart	datetime	Unchecked
Comment	ntext	Checked

6.3 - PATH TABLE

ApplicationId	uniqueidentifier	Unchecked
PathId	uniqueidentifier	Unchecked
Path	nvarchar(256)	Unchecked
LoweredPath	nvarchar(256)	Unchecked

6.4 - PERSONALIZATION ALLUSER TABLE

PathId	uniqueidentifier	Unchecked
PageSettings	image	Unchecked
LastUpdatedDate	datetime	Unchecked

6.5 - PERSONALIZATION PER USER TABLE

Id	uniqueidentifier	Unchecked
PathId	uniqueidentifier	Checked
UserId	uniqueidentifier	Checked
PageSettings	image	Unchecked
LastUpdatedDate	datetime	Unchecked

6.6 - PROFILE TABLE

UserId	uniqueidentifier	Unchecked
Property	Names ntext	Unchecked
PropertyValuesString	ntext	Unchecked
PropertyValuesBinary	image	Unchecked
LastUpdatedDate	datetime	Unchecked

6.7 - ROLES TABLE

ApplicationId	uniqueidentifier	Unchecked
RoleId	uniqueidentifier	Unchecked
RoleName	nvarchar(256)	Unchecked
LoweredRoleName	nvarchar(256)	Unchecked
Description	nvarchar(256)	Checked

6.8 - SCHEMA TABLE

Feature	nvarchar(128)	Unchecked
CompatibleSchemaVersion	nvarchar(128)	Unchecked
IsCurrentVersion	bit	Unchecked

6.9 - USERS TABLE

ApplicationId	uniqueidentifier	Unchecked
UserId	uniqueidentifier	Unchecked
UserName	nvarchar(256)	Unchecked

LoweredUserName	nvarchar(256)	Unchecked
MobileAlias	nvarchar(16)	Checked
IsAnonymous	bit	Unchecked
LastActivityDate	datetime	Unchecked

6.10 - USERSINROLE TABLE

UserId	uniqueidentifier	Unchecked
RoleId	uniqueidentifier	Unchecked

6.11 - WEB EVENT TABLE

EventId	char(32)	Unchecked
EventTimeUtc	datetime	Unchecked
EventTime	datetime	Unchecked
EventType	nvarchar(256)	Unchecked
EventSequence	decimal(19, 0)	Unchecked
EventOccurrence	decimal(19, 0)	Unchecked
EventCode	int	Unchecked
EventDetailCode	int	Unchecked
Message	nvarchar(1024)	Checked
ApplicationPath	nvarchar(256)	Checked
ApplicationVirtualPath	nvarchar(256)	Checked
MachineName	nvarchar(256)	Unchecked
RequestUrl	nvarchar(1024)	Checked
ExceptionType	nvarchar(256)	Checked
Details	ntext	Checked

6.12 - KATEGORI TABLE

KategoriId	smallint	Unchecked
KategoriAdi	nvarchar(30)	Unchecked

6.13 - MARKA TABLE

MarkaId	smallint	Unchecked
MarkaAdi	nvarchar(30)	Unchecked
UrunAdi	nvarchar(30)	Unchecked

6.14 - MODEL TABLE

ModelId	smallint	Unchecked
ModelAdi	nvarchar(30)	Unchecked
MarkaAdi	nvarchar(30)	Unchecked
UrunAdi	nvarchar(30)	Unchecked
Fiyat	int	Unchecked
StokMiktari	smallint	Unchecked

6.15 - MODELOZELLIK TABLE

ModelAdi	nvarchar(30)	Unchecked
ModelOzellikleri	text	Unchecked
ModelFiyatı	int	Unchecked
Taksit_ay	tinyint	Unchecked



6.16 - SEPET TABLE

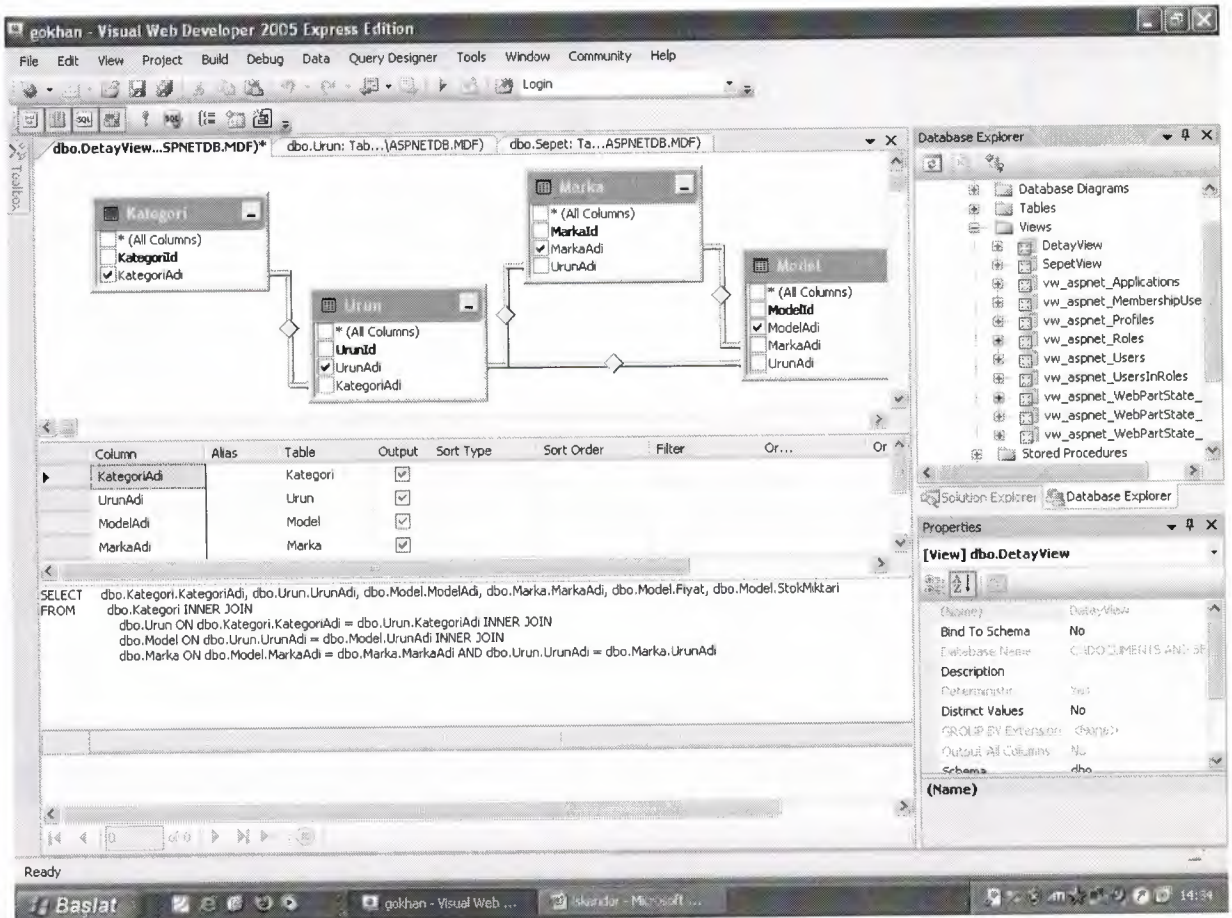
SepetId	smallint	Unchecked
ModelAdi	nvarchar(30)	Unchecked
Fiyat	int	Unchecked

6.17 - URUN TABLE

UrunId	smallint	Unchecked
UrunAdi	nvarchar(30)	Unchecked
KategoriAdi	nvarchar(30)	Unchecked

CHAPTER 7 : QUERIES AND DESIGNS AND RESULTS FOR VIEWS

7.1 - DETAY VIEWS



```
SELECT  dbo.Kategori.KategoriAdi, dbo.Urun.UrunAdi, dbo.Model.ModelAdi,
dbo.Marka.MarkaAdi, dbo.Model.Fiyat, dbo.Model.StokMiktari
```

```
FROM    dbo.Kategori INNER JOIN
```

```
        dbo.Urun ON dbo.Kategori.KategoriAdi = dbo.Urun.KategoriAdi INNER
JOIN
```

```
        dbo.Model ON dbo.Urun.UrunAdi = dbo.Model.UrunAdi INNER JOIN
```

```
        dbo.Marka ON dbo.Model.MarkaAdi = dbo.Marka.MarkaAdi AND
dbo.Urun.UrunAdi = dbo.Marka.UrunAdi
```


7.1 - a) RESULT

gokhan - Visual Web Developer 2005 Express Edition

File Edit View Project Build Debug Data Query Designer Tools Window Community Help

Change Type

DetayView: Qu...ASPNETDB.MDF) dbo.DetayView...SPNETDB.MDF)* dbo.Urun: Tab... (ASPNETDB.MDF)

KategoriAdi	UrunAdi	ModelAdi	MarkaAdi	Fiyat	StokMiktarı
Beyaz Eşya	Buzdolabı	arç-no-frost	Arçelik	800	10
Beyaz Eşya	Buzdolabı	arç-gift kapaklı	Arçelik	900	3
Beyaz Eşya	Buzdolabı	arç-tek kapaklı	Arçelik	750	-1
Beyaz Eşya	Buzdolabı	vest-no-frost	Vestel	500	16
Beyaz Eşya	Buzdolabı	vest-500 lt	Vestel	900	5
Beyaz Eşya	Çamaşır Makinesi	5000 Devir	Arçelik	600	17
Beyaz Eşya	Çamaşır Makinesi	3000 Devir	Arçelik	450	41
Beyaz Eşya	Çamaşır Makinesi	A-Plus	Vestel	700	4
Beyaz Eşya	Çamaşır Makinesi	C-Plus	Vestel	550	10
Elektronik	Diz Üstü	Pentium3 rx2	Arçelik	1000	1
Elektronik	Diz Üstü	Pentium Celeron	Arçelik	1600	-2
Elektronik	Diz Üstü	vest lptp celrn	Vestel	1500	14
Elektronik	Diz Üstü	vest lptp P4+	Vestel	1100	10
Elektronik	Masa Üstü	pentium4	Vestel	1200	11
Elektronik	Masa Üstü	Deskotp Celeron	Vestel	800	1
Elektronik	Masa Üstü	Arç Desktop P4	Arçelik	1000	0
Elektronik	Masa Üstü	Arç Desktop Cel3	Arçelik	1200	7
* NULL	NULL	NULL	NULL	NULL	NULL

Database Explorer

Sepet
Urun
Views
DetayView
SepetView
vw_aspnet_Applications
vw_aspnet_MembershipUse
vw_aspnet_Profiles
vw_aspnet_Roles
vw_aspnet_Users
vw_aspnet_UsersInRoles
vw_aspnet_WebPartState_
vw_aspnet_WebPartState_
vw_aspnet_WebPartState_
Stored Procedures

Solution Explorer Database Explorer

Properties

[Qry] Query

(Name) Query
Database Name: ASPNETDB.MDF
Destination Table:
Distinct Values: No
GROUP BY Extension: <Name>
Output All Columns: Yes
Query Parameter List: No parameters have been defined.
Server Name: localhost:54321
SQL Comment:
(Name)

Ready

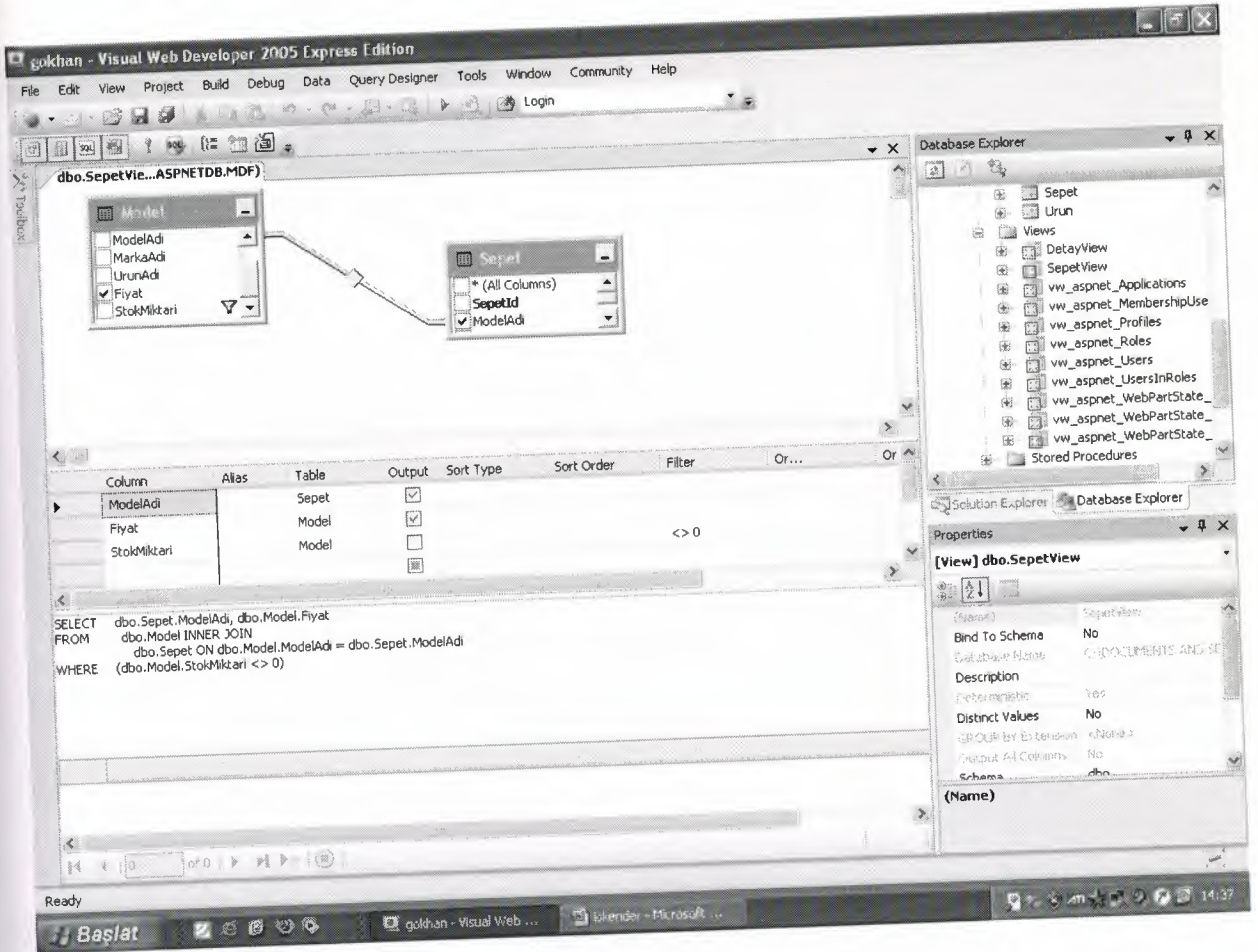
Başlat

gokhan - Visual Web ...

Microsoft ...

14:36

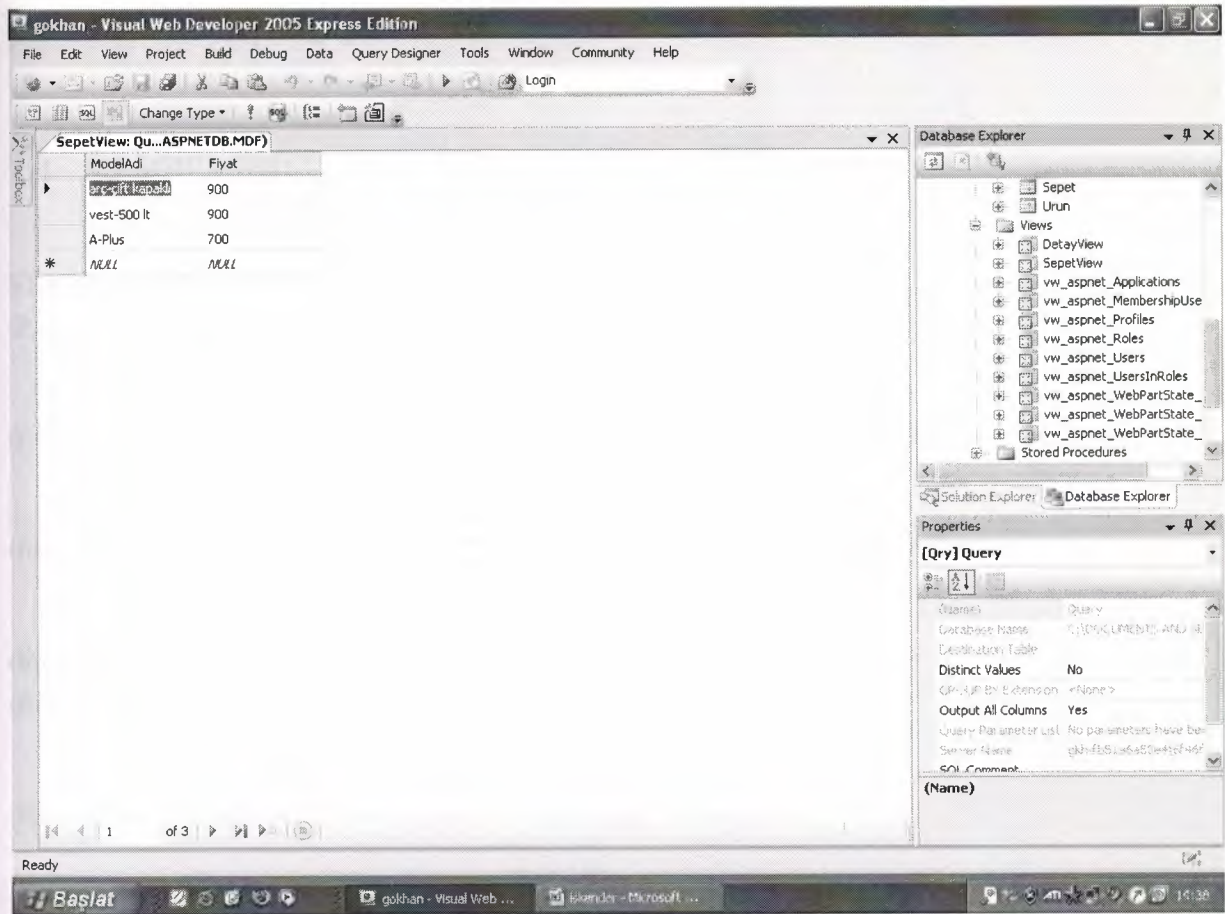
7.2 - SEPET VIEW



```

SELECT  dbo.Sepet.ModelAdi, dbo.Model.Fiyat
FROM    dbo.Model INNER JOIN
        dbo.Sepet ON dbo.Model.ModelAdi = dbo.Sepet.ModelAdi
WHERE   (dbo.Model.StokMiktari <> 0)
  
```

7.2 - a) RESULT



7.3 - APPLICATIONS VIEW

SELECT ApplicationName, LoweredApplicationName, ApplicationId, Description

FROM dbo.aspnet_Applications

7.4 - MEMBERSHIPUSERS VIEW

```
SELECT    dbo.aspnet_Membership.UserId, dbo.aspnet_Membership.PasswordFormat,
dbo.aspnet_Membership.MobilePIN, dbo.aspnet_Membership.Email,

          dbo.aspnet_Membership.LoweredEmail,
dbo.aspnet_Membership.PasswordQuestion, dbo.aspnet_Membership.PasswordAnswer,

          dbo.aspnet_Membership.IsApproved,
dbo.aspnet_Membership.IsLockedOut, dbo.aspnet_Membership.CreateDate,

          dbo.aspnet_Membership.LastLoginDate,
dbo.aspnet_Membership.LastPasswordChangedDate,
dbo.aspnet_Membership.LastLockoutDate,

          dbo.aspnet_Membership.FailedPasswordAttemptCount,
dbo.aspnet_Membership.FailedPasswordAttemptWindowStart,

          dbo.aspnet_Membership.FailedPasswordAnswerAttemptCount,
dbo.aspnet_Membership.FailedPasswordAnswerAttemptWindowStart,

          dbo.aspnet_Membership.Comment, dbo.aspnet_Users.ApplicationId,
dbo.aspnet_Users.UserName, dbo.aspnet_Users.MobileAlias,

          dbo.aspnet_Users.IsAnonymous, dbo.aspnet_Users.LastActivityDate

FROM      dbo.aspnet_Membership INNER JOIN

          dbo.aspnet_Users ON dbo.aspnet_Membership.UserId =
dbo.aspnet_Users.UserId
```

7.5 - PROFILES VIEW

```
SELECT    UserId, LastUpdatedDate, DATALENGTH(PropertyNames) +
DATALENGTH(PropertyValuesString) + DATALENGTH(PropertyValuesBinary)

          AS DataSize

FROM      dbo.aspnet_Profile
```


7.6 - ROLES VIEW

```
SELECT  ApplicationId, RoleId, RoleName, LoweredRoleName, Description
FROM    dbo.aspnet_Roles
```

7.7 - USERS VIEW

```
SELECT  ApplicationId, UserId, UserName, LoweredUserName, MobileAlias,
IsAnonymous, LastActivityDate
FROM    dbo.aspnet_Users
```

7.8 - USERSINROLES VIEW

```
SELECT  UserId, RoleId
FROM    dbo.aspnet_UsersInRoles
```

7.9 - PATHS VIEW

```
SELECT  ApplicationId, PathId, Path, LoweredPath
FROM    dbo.aspnet_Paths
```

7.10 - SHARED VIEW

```
SELECT  PathId, DATALENGTH(PageSettings) AS DataSize, LastUpdatedDate
FROM    dbo.aspnet_PersonalizationAllUsers
```

7.11 - WEBPARTUSERS VIEWS

```
SELECT  PathId, UserId, DATALENGTH(PageSettings) AS DataSize,  
        LastUpdatedDate  
  
FROM    dbo.aspnet_PersonalizationPerUser
```

CHAPTER 8 : STORED PROCEDURES

8.1 - ANYDATAINTABLES

```
ALTER PROCEDURE [dbo].aspnet_AnyDataInTables
    @TablesToCheck int
AS
BEGIN
    -- Check Membership table if (@TablesToCheck & 1) is set
    IF ((@TablesToCheck & 1) <> 0 AND
        (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_MembershipUsers') AND (type = 'V'))))
    BEGIN
        IF (EXISTS(SELECT TOP 1 UserId FROM dbo.aspnet_Membership))
        BEGIN
            SELECT N'aspnet_Membership'
            RETURN
        END
    END

    -- Check aspnet_Roles table if (@TablesToCheck & 2) is set
    IF ((@TablesToCheck & 2) <> 0 AND
        (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_Roles') AND (type = 'V'))))
    BEGIN
        IF (EXISTS(SELECT TOP 1 RoleId FROM dbo.aspnet_Roles))
        BEGIN
            SELECT N'aspnet_Roles'
            RETURN
        END
    END

    -- Check aspnet_Profile table if (@TablesToCheck & 4) is set
    IF ((@TablesToCheck & 4) <> 0 AND
        (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_Profiles') AND (type = 'V'))))
    BEGIN
        IF (EXISTS(SELECT TOP 1 UserId FROM dbo.aspnet_Profile))
        BEGIN
            SELECT N'aspnet_Profile'
            RETURN
        END
    END
END
```

```

-- Check aspnet_PersonalizationPerUser table if (@TablesToCheck & 8) is set
IF ((@TablesToCheck & 8) <> 0 AND
    (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_WebPartState_User') AND (type = 'V')))) )
BEGIN
    IF (EXISTS(SELECT TOP 1 UserId FROM dbo.aspnet_PersonalizationPerUser))
    BEGIN
        SELECT N'aspnet_PersonalizationPerUser'
        RETURN
    END
END

-- Check aspnet_PersonalizationPerUser table if (@TablesToCheck & 16) is set
IF ((@TablesToCheck & 16) <> 0 AND
    (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'aspnet_WebEvent_LogEvent') AND (type = 'P')))) )
BEGIN
    IF (EXISTS(SELECT TOP 1 * FROM dbo.aspnet_WebEvent_Events))
    BEGIN
        SELECT N'aspnet_WebEvent_Events'
        RETURN
    END
END

-- Check aspnet_Users table if (@TablesToCheck & 1,2,4 & 8) are all set
IF ((@TablesToCheck & 1) <> 0 AND
    (@TablesToCheck & 2) <> 0 AND
    (@TablesToCheck & 4) <> 0 AND
    (@TablesToCheck & 8) <> 0 AND
    (@TablesToCheck & 32) <> 0 AND
    (@TablesToCheck & 128) <> 0 AND
    (@TablesToCheck & 256) <> 0 AND
    (@TablesToCheck & 512) <> 0 AND
    (@TablesToCheck & 1024) <> 0)
BEGIN
    IF (EXISTS(SELECT TOP 1 UserId FROM dbo.aspnet_Users))
    BEGIN
        SELECT N'aspnet_Users'
        RETURN
    END
    IF (EXISTS(SELECT TOP 1 ApplicationId FROM dbo.aspnet_Applications))
    BEGIN
        SELECT N'aspnet_Applications'
        RETURN
    END
END
END

```


8.2 - CREATE APPLICATION

```
ALTER PROCEDURE [dbo].aspnet_Applications_CreateApplication
    @ApplicationName    nvarchar(256),
    @ApplicationId      uniqueidentifier OUTPUT
AS
BEGIN
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
    LOWER(@ApplicationName) = LoweredApplicationName

    IF(@ApplicationId IS NULL)
    BEGIN
        DECLARE @TranStarted bit
        SET @TranStarted = 0

        IF( @@TRANCOUNT = 0 )
        BEGIN
            BEGIN TRANSACTION
            SET @TranStarted = 1
        END
        ELSE
            SET @TranStarted = 0

        SELECT @ApplicationId = ApplicationId
        FROM dbo.aspnet_Applications WITH (UPDLOCK, HOLDLOCK)
        WHERE LOWER(@ApplicationName) = LoweredApplicationName

        IF(@ApplicationId IS NULL)
        BEGIN
            SELECT @ApplicationId = NEWID()
            INSERT dbo.aspnet_Applications (ApplicationId, ApplicationName,
            LoweredApplicationName)
            VALUES (@ApplicationId, @ApplicationName,
            LOWER(@ApplicationName))
        END

        IF( @TranStarted = 1 )
        BEGIN
            IF( @@ERROR = 0 )
            BEGIN
                SET @TranStarted = 0
                COMMIT TRANSACTION
            END
            ELSE
            BEGIN
                SET @TranStarted = 0
                ROLLBACK TRANSACTION
            END
        END
    END
```

```

        END
    END
END
END

```

8.3 - CHECHSCHEMAVERSION

```

ALTER PROCEDURE [dbo].aspnet_Applications_CreateApplication
    @ApplicationName    nvarchar(256),
    @ApplicationId       uniqueidentifier OUTPUT
AS
BEGIN
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
    LOWER(@ApplicationName) = LoweredApplicationName

    IF(@ApplicationId IS NULL)
    BEGIN
        DECLARE @TranStarted bit
        SET @TranStarted = 0

        IF( @@TRANCOUNT = 0 )
        BEGIN
            BEGIN TRANSACTION
            SET @TranStarted = 1
        END
        ELSE
            SET @TranStarted = 0

        SELECT @ApplicationId = ApplicationId
        FROM dbo.aspnet_Applications WITH (UPDLOCK, HOLDLOCK)
        WHERE LOWER(@ApplicationName) = LoweredApplicationName

        IF(@ApplicationId IS NULL)
        BEGIN
            SELECT @ApplicationId = NEWID()
            INSERT dbo.aspnet_Applications (ApplicationId, ApplicationName,
            LoweredApplicationName)
            VALUES (@ApplicationId, @ApplicationName,
            LOWER(@ApplicationName))
        END

        IF( @TranStarted = 1 )
    
```

```

BEGIN
    IF(@@ERROR = 0)
    BEGIN
        SET @TranStarted = 0
        COMMIT TRANSACTION
    END
    ELSE
    BEGIN
        SET @TranStarted = 0
        ROLLBACK TRANSACTION
    END
END
END
END

```

8.4 - EDITMEMBERSHIP

```

ALTER PROCEDURE [dbo].aspnet_Applications_CreateApplication
    @ApplicationName    nvarchar(256),
    @ApplicationId       uniqueidentifier OUTPUT
AS
BEGIN
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
    LOWER(@ApplicationName) = LoweredApplicationName

    IF(@ApplicationId IS NULL)
    BEGIN
        DECLARE @TranStarted bit
        SET @TranStarted = 0

        IF( @@TRANCOUNT = 0 )
        BEGIN
            BEGIN TRANSACTION
            SET @TranStarted = 1
        END
        ELSE
            SET @TranStarted = 0

        SELECT @ApplicationId = ApplicationId
        FROM dbo.aspnet_Applications WITH (UPDLOCK, HOLDLOCK)
        WHERE LOWER(@ApplicationName) = LoweredApplicationName

        IF(@ApplicationId IS NULL)
    
```

```

BEGIN
    SELECT @ApplicationId = NEWID()
    INSERT dbo.aspnet_Applications (ApplicationId, ApplicationName,
LoweredApplicationName)
        VALUES (@ApplicationId, @ApplicationName,
LOWER(@ApplicationName))
    END

IF( @TranStarted = 1 )
BEGIN
    IF( @@ERROR = 0)
    BEGIN
        SET @TranStarted = 0
        COMMIT TRANSACTION
    END
    ELSE
    BEGIN
        SET @TranStarted = 0
        ROLLBACK TRANSACTION
    END
END
END
END

```

8.5 - CREATEUSER

```

ALTER PROCEDURE dbo.aspnet_Membership_CreateUser
    @ApplicationName      nvarchar(256),
    @UserName              nvarchar(256),
    @Password              nvarchar(128),
    @PasswordSalt          nvarchar(128),
    @Email                 nvarchar(256),
    @PasswordQuestion      nvarchar(256),
    @PasswordAnswer        nvarchar(128),
    @IsApproved            bit,
    @CurrentTimeUtc        datetime,
    @CreateDate            datetime = NULL,
    @UniqueEmail            int    = 0,
    @PasswordFormat         int    = 0,
    @UserId                uniqueidentifier OUTPUT
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier

```



```

SELECT @ApplicationId = NULL

DECLARE @NewUserId uniqueidentifier
SELECT @NewUserId = NULL

DECLARE @IsLockedOut bit
SET @IsLockedOut = 0

DECLARE @LastLockoutDate datetime
SET @LastLockoutDate = CONVERT( datetime, '17540101', 112 )

DECLARE @FailedPasswordAttemptCount int
SET @FailedPasswordAttemptCount = 0

DECLARE @FailedPasswordAttemptWindowStart datetime
SET @FailedPasswordAttemptWindowStart = CONVERT( datetime, '17540101',
112 )

DECLARE @FailedPasswordAnswerAttemptCount int
SET @FailedPasswordAnswerAttemptCount = 0

DECLARE @FailedPasswordAnswerAttemptWindowStart datetime
SET @FailedPasswordAnswerAttemptWindowStart = CONVERT( datetime,
'17540101', 112 )

DECLARE @NewUserCreated bit
DECLARE @ReturnValue int
SET @ReturnValue = 0

DECLARE @ErrorCode int
SET @ErrorCode = 0

DECLARE @TranStarted bit
SET @TranStarted = 0

IF( @@TRANCOUNT = 0 )
BEGIN
    BEGIN TRANSACTION
    SET @TranStarted = 1
END
ELSE
    SET @TranStarted = 0

EXEC dbo.aspnet_Applications_CreateApplication @ApplicationName,
@ApplicationId OUTPUT

IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup

```

END

SET @CreateDate = @CurrentTimeUtc

SELECT @NewUserId = UserId FROM dbo.aspnet_Users WHERE
LOWER(@UserName) = LoweredUserName AND @ApplicationId = ApplicationId
IF (@NewUserId IS NULL)

BEGIN

SET @NewUserId = @UserId

EXEC @ReturnValue = dbo.aspnet_Users_CreateUser @ApplicationId,
@UserName, 0, @CreateDate, @NewUserId OUTPUT

SET @NewUserCreated = 1

END

ELSE

BEGIN

SET @NewUserCreated = 0

IF(@NewUserId <> @UserId AND @UserId IS NOT NULL)

BEGIN

SET @ErrorCode = 6

GOTO Cleanup

END

END

IF(@@ERROR <> 0)

BEGIN

SET @ErrorCode = -1

GOTO Cleanup

END

IF(@ReturnValue = -1)

BEGIN

SET @ErrorCode = 10

GOTO Cleanup

END

IF (EXISTS (SELECT UserId
FROM dbo.aspnet_Membership
WHERE @NewUserId = UserId))

BEGIN

SET @ErrorCode = 6

GOTO Cleanup

END

SET @UserId = @NewUserId

IF (@UniqueEmail = 1)

BEGIN

IF (EXISTS (SELECT *
FROM dbo.aspnet_Membership m WITH (UPDLOCK, HOLDLOCK)

```
WHERE ApplicationId = @ApplicationId AND LoweredEmail =  
LOWER(@Email)))
```

```
BEGIN  
    SET @ErrorCode = 7  
    GOTO Cleanup  
END  
END
```

```
IF (@NewUserCreated = 0)  
BEGIN  
    UPDATE dbo.aspnet_Users  
    SET LastActivityDate = @CreateDate  
    WHERE @UserId = UserId  
    IF( @@ERROR <> 0 )  
    BEGIN  
        SET @ErrorCode = -1  
        GOTO Cleanup  
    END  
END
```

```
INSERT INTO dbo.aspnet_Membership  
    ( ApplicationId,  
      UserId,  
      Password,  
      PasswordSalt,  
      Email,  
      LoweredEmail,  
      PasswordQuestion,  
      PasswordAnswer,  
      PasswordFormat,  
      IsApproved,  
      IsLockedOut,  
      CreateDate,  
      LastLoginDate,  
      LastPasswordChangedDate,  
      LastLockoutDate,  
      FailedPasswordAttemptCount,  
      FailedPasswordAttemptWindowStart,  
      FailedPasswordAnswerAttemptCount,  
      FailedPasswordAnswerAttemptWindowStart )  
VALUES ( @ApplicationId,  
        @UserId,  
        @Password,  
        @PasswordSalt,  
        @Email,  
        LOWER(@Email),  
        @PasswordQuestion,  
        @PasswordAnswer,  
        @PasswordFormat,  
        @IsApproved,
```

```

        @IsLockedOut,
        @CreateDate,
        @CreateDate,
        @CreateDate,
        @LastLockoutDate,
        @FailedPasswordAttemptCount,
        @FailedPasswordAttemptWindowStart,
        @FailedPasswordAnswerAttemptCount,
        @FailedPasswordAnswerAttemptWindowStart )

IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup
END

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    COMMIT TRANSACTION
END

RETURN 0

Cleanup:

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END

RETURN @ErrorCode

END

```

8.6 - FINDUSERBYEMAIL

```

ALTER PROCEDURE dbo.aspnet_Membership_FindUsersByEmail
    @ApplicationName    nvarchar(256),
    @EmailToMatch       nvarchar(256),
    @PageIndex          int,

```



```

    @PageSize          int
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN 0

    -- Set the page bounds
    DECLARE @PageLowerBound int
    DECLARE @PageUpperBound int
    DECLARE @TotalRecords int
    SET @PageLowerBound = @PageSize * @PageIndex
    SET @PageUpperBound = @PageSize - 1 + @PageLowerBound

    -- Create a temp table TO store the select results
    CREATE TABLE #PageIndexForUsers
    (
        IndexId int IDENTITY (0, 1) NOT NULL,
        UserId uniqueidentifier
    )

    -- Insert into our temp table
    IF( @EmailToMatch IS NULL )
        INSERT INTO #PageIndexForUsers (UserId)
            SELECT u.UserId
            FROM  dbo.aspnet_Users u, dbo.aspnet_Membership m
            WHERE u.ApplicationId = @ApplicationId AND m.UserId = u.UserId AND
m.Email IS NULL
            ORDER BY m.LoweredEmail
    ELSE
        INSERT INTO #PageIndexForUsers (UserId)
            SELECT u.UserId
            FROM  dbo.aspnet_Users u, dbo.aspnet_Membership m
            WHERE u.ApplicationId = @ApplicationId AND m.UserId = u.UserId AND
m.LoweredEmail LIKE LOWER(@EmailToMatch)
            ORDER BY m.LoweredEmail

    SELECT u.UserName, m.Email, m.PasswordQuestion, m.Comment, m.IsApproved,
        m.CreateDate,
        m.LastLoginDate,
        u.LastActivityDate,
        m.LastPasswordChangedDate,
        u.UserId, m.IsLockedOut,
        m.LastLockoutDate
    FROM  dbo.aspnet_Membership m, dbo.aspnet_Users u, #PageIndexForUsers p
    WHERE u.UserId = p.UserId AND u.UserId = m.UserId AND
        p.IndexId >= @PageLowerBound AND p.IndexId <= @PageUpperBound

```

```

ORDER BY m.LoweredEmail

SELECT @TotalRecords = COUNT(*)
FROM #PageIndexForUsers
RETURN @TotalRecords
END

```

8.7 - FINDUSERSBYNAME

```

ALTER PROCEDURE dbo.aspnet_Membership_FindUsersByName
    @ApplicationName    nvarchar(256),
    @UserNameToMatch    nvarchar(256),
    @PageIndex          int,
    @PageSize            int
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN 0

    -- Set the page bounds
    DECLARE @PageLowerBound int
    DECLARE @PageUpperBound int
    DECLARE @TotalRecords int
    SET @PageLowerBound = @PageSize * @PageIndex
    SET @PageUpperBound = @PageSize - 1 + @PageLowerBound

    -- Create a temp table TO store the select results
    CREATE TABLE #PageIndexForUsers
    (
        IndexId int IDENTITY (0, 1) NOT NULL,
        UserId uniqueidentifier
    )

    -- Insert into our temp table
    INSERT INTO #PageIndexForUsers (UserId)
        SELECT u.UserId

```

```

FROM dbo.aspnet_Users u, dbo.aspnet_Membership m
WHERE u.ApplicationId = @ApplicationId AND m.UserId = u.UserId AND
u.LoweredUserName LIKE LOWER(@UserNameToMatch)
ORDER BY u.UserName

```

```

SELECT u.UserName, m.Email, m.PasswordQuestion, m.Comment, m.IsApproved,
m.CreateDate,
m.LastLoginDate,
u.LastActivityDate,
m.LastPasswordChangedDate,
u.UserId, m.IsLockedOut,
m.LastLockoutDate
FROM dbo.aspnet_Membership m, dbo.aspnet_Users u, #PageIndexForUsers p
WHERE u.UserId = p.UserId AND u.UserId = m.UserId AND
p.IndexId >= @PageLowerBound AND p.IndexId <= @PageUpperBound
ORDER BY u.UserName

```

```

SELECT @TotalRecords = COUNT(*)
FROM #PageIndexForUsers
RETURN @TotalRecords
END

```

8.8 - GETALLUSERS

```

ALTER PROCEDURE dbo.aspnet_Membership_FindUsersByName
    @ApplicationName nvarchar(256),
    @UserNameToMatch nvarchar(256),
    @PageIndex int,
    @PageSize int
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
    LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN 0

    -- Set the page bounds
    DECLARE @PageLowerBound int
    DECLARE @PageUpperBound int

```

```

DECLARE @TotalRecords int
SET @PageLowerBound = @PageSize * @PageIndex
SET @PageUpperBound = @PageSize - 1 + @PageLowerBound

-- Create a temp table TO store the select results
CREATE TABLE #PageIndexForUsers
(
    IndexId int IDENTITY (0, 1) NOT NULL,
    UserId uniqueidentifier
)

-- Insert into our temp table
INSERT INTO #PageIndexForUsers (UserId)
    SELECT u.UserId
    FROM   dbo.aspnet_Users u, dbo.aspnet_Membership m
    WHERE  u.ApplicationId = @ApplicationId AND m.UserId = u.UserId AND
u.LoweredUserName LIKE LOWER(@UserNameToMatch)
    ORDER BY u.UserName

SELECT  u.UserName, m.Email, m.PasswordQuestion, m.Comment, m.IsApproved,
        m.CreateDate,
        m.LastLoginDate,
        u.LastActivityDate,
        m.LastPasswordChangedDate,
        u.UserId, m.IsLockedOut,
        m.LastLockoutDate
FROM    dbo.aspnet_Membership m, dbo.aspnet_Users u, #PageIndexForUsers p
WHERE   u.UserId = p.UserId AND u.UserId = m.UserId AND
        p.IndexId >= @PageLowerBound AND p.IndexId <= @PageUpperBound
ORDER BY u.UserName

SELECT  @TotalRecords = COUNT(*)
FROM    #PageIndexForUsers
RETURN @TotalRecords
END

```

8.9 - GETNUMBEROFUSERSONLINE

```

ALTER PROCEDURE dbo.aspnet_Membership_GetNumberOfUsersOnline
    @ApplicationName      nvarchar(256),
    @MinutesSinceLastInActive int,

```



```

    @CurrentTimeUtc      datetime
AS
BEGIN
    DECLARE @DateActive datetime
    SELECT @DateActive = DATEADD(minute, -(@MinutesSinceLastInActive),
@CurrentTimeUtc)

    DECLARE @NumOnline int
    SELECT @NumOnline = COUNT(*)
    FROM    dbo.aspnet_Users u(NOLOCK),
            dbo.aspnet_Applications a(NOLOCK),
            dbo.aspnet_Membership m(NOLOCK)
    WHERE   u.ApplicationId = a.ApplicationId          AND
            LastActivityDate > @DateActive            AND
            a.LoweredApplicationName = LOWER(@ApplicationName) AND
            u.UserId = m.UserId
    RETURN(@NumOnline)
END

```

8.10 - GETPASSWORD

```

ALTER PROCEDURE dbo.aspnet_Membership_GetNumberOfUsersOnline
    @ApplicationName      nvarchar(256),
    @MinutesSinceLastInActive int,
    @CurrentTimeUtc      datetime
AS
BEGIN
    DECLARE @DateActive datetime
    SELECT @DateActive = DATEADD(minute, -(@MinutesSinceLastInActive),
@CurrentTimeUtc)

    DECLARE @NumOnline int
    SELECT @NumOnline = COUNT(*)
    FROM    dbo.aspnet_Users u(NOLOCK),
            dbo.aspnet_Applications a(NOLOCK),
            dbo.aspnet_Membership m(NOLOCK)
    WHERE   u.ApplicationId = a.ApplicationId          AND
            LastActivityDate > @DateActive            AND
            a.LoweredApplicationName = LOWER(@ApplicationName) AND
            u.UserId = m.UserId
    RETURN(@NumOnline)
END

```

8.11 - GETPASSWORDWITHFORMAT

```
ALTER PROCEDURE dbo.aspnet_Membership_GetPasswordWithFormat
    @ApplicationName      nvarchar(256),
    @UserName              nvarchar(256),
    @UpdateLastLoginActivityDate bit,
    @CurrentTimeUtc        datetime
AS
BEGIN
    DECLARE @IsLockedOut      bit
    DECLARE @UserId            uniqueidentifier
    DECLARE @Password          nvarchar(128)
    DECLARE @PasswordSalt      nvarchar(128)
    DECLARE @PasswordFormat    int
    DECLARE @FailedPasswordAttemptCount int
    DECLARE @FailedPasswordAnswerAttemptCount int
    DECLARE @IsApproved        bit
    DECLARE @LastActivityDate   datetime
    DECLARE @LastLoginDate      datetime

    SELECT @UserId = NULL

    SELECT @UserId = u.UserId, @IsLockedOut = m.IsLockedOut,
    @Password=Password, @PasswordFormat=PasswordFormat,
    @PasswordSalt=PasswordSalt,
    @FailedPasswordAttemptCount=FailedPasswordAttemptCount,
    @FailedPasswordAnswerAttemptCount=FailedPasswordAnswerAttemptCount,
    @IsApproved=IsApproved,
    @LastActivityDate = LastActivityDate, @LastLoginDate = LastLoginDate
    FROM  dbo.aspnet_Applications a, dbo.aspnet_Users u, dbo.aspnet_Membership m
    WHERE LOWER(@ApplicationName) = a.LoweredApplicationName AND
    u.ApplicationId = a.ApplicationId AND
    u.UserId = m.UserId AND
    LOWER(@UserName) = u.LoweredUserName

    IF (@UserId IS NULL)
        RETURN 1

    IF (@IsLockedOut = 1)
        RETURN 99
```

```

SELECT  @Password, @PasswordFormat, @PasswordSalt,
@FailedPasswordAttemptCount,
        @FailedPasswordAnswerAttemptCount, @IsApproved, @LastLoginDate,
@LastActivityDate

```

```

IF ( @UpdateLastLoginActivityDate = 1 AND @IsApproved = 1)

```

```

BEGIN

```

```

    UPDATE dbo.aspnet_Membership
    SET     LastLoginDate = @CurrentTimeUtc
    WHERE  UserId = @UserId

```

```

    UPDATE dbo.aspnet_Users
    SET     LastActivityDate = @CurrentTimeUtc
    WHERE  @UserId = UserId

```

```

END

```

```

RETURN 0

```

```

END

```

8.12 - GETUSERSBYEMAIL

```

ALTER PROCEDURE dbo.aspnet_Membership_GetUserByEmail

```

```

    @ApplicationName nvarchar(256),

```

```

    @Email          nvarchar(256)

```

```

AS

```

```

BEGIN

```

```

    IF( @Email IS NULL )

```

```

        SELECT u.UserName

```

```

        FROM  dbo.aspnet_Applications a, dbo.aspnet_Users u, dbo.aspnet_Membership

```

```

m

```

```

        WHERE LOWER(@ApplicationName) = a.LoweredApplicationName AND

```

```

            u.ApplicationId = a.ApplicationId  AND

```

```

            u.UserId = m.UserId AND

```

```

            m.LoweredEmail IS NULL

```

```

    ELSE

```

```

        SELECT u.UserName

```

```

        FROM  dbo.aspnet_Applications a, dbo.aspnet_Users u, dbo.aspnet_Membership

```

```

m

```

```

        WHERE LOWER(@ApplicationName) = a.LoweredApplicationName AND

```

```

            u.ApplicationId = a.ApplicationId  AND

```

```

            u.UserId = m.UserId AND

```

```

            LOWER(@Email) = m.LoweredEmail

```

```

IF (@@rowcount = 0)
    RETURN(1)
RETURN(0)
END

```

8.13 - GETUSERBYNAME

```

ALTER PROCEDURE dbo.aspnet_Membership_GetUserByName
    @ApplicationName    nvarchar(256),
    @UserName           nvarchar(256),
    @CurrentTimeUtc     datetime,
    @UpdateLastActivity bit = 0
AS
BEGIN
    DECLARE @UserId uniqueidentifier

    IF (@UpdateLastActivity = 1)
    BEGIN
        SELECT TOP 1 m.Email, m.PasswordQuestion, m.Comment, m.IsApproved,
            m.CreateDate, m.LastLoginDate, @CurrentTimeUtc,
m.LastPasswordChangedDate,
            u.UserId, m.IsLockedOut, m.LastLockoutDate
        FROM    dbo.aspnet_Applications a, dbo.aspnet_Users u, dbo.aspnet_Membership
m
        WHERE   LOWER(@ApplicationName) = a.LoweredApplicationName AND
            u.ApplicationId = a.ApplicationId AND
            LOWER(@UserName) = u.LoweredUserName AND u.UserId = m.UserId

        IF (@@ROWCOUNT = 0) -- Username not found
            RETURN -1

        UPDATE  dbo.aspnet_Users
        SET      LastActivityDate = @CurrentTimeUtc
        WHERE    @UserId = UserId
    END
    ELSE
    BEGIN
        SELECT TOP 1 m.Email, m.PasswordQuestion, m.Comment, m.IsApproved,
            m.CreateDate, m.LastLoginDate, u.LastActivityDate,
m.LastPasswordChangedDate,
            u.UserId, m.IsLockedOut, m.LastLockoutDate
        FROM    dbo.aspnet_Applications a, dbo.aspnet_Users u, dbo.aspnet_Membership
m
        WHERE   LOWER(@ApplicationName) = a.LoweredApplicationName AND
            u.ApplicationId = a.ApplicationId AND

```



```

        LOWER(@UserName) = u.LoweredUserName AND u.UserId = m.UserId

    IF ( @@ROWCOUNT = 0 ) -- Username not found
        RETURN -1
    END

    RETURN 0
END

```

8.14 - GETUSERBYUSERID

```

ALTER PROCEDURE dbo.aspnet_Membership_GetUserByUserId
    @UserId          uniqueidentifier,
    @CurrentTimeUtc  datetime,
    @UpdateLastActivity bit = 0
AS
BEGIN
    IF ( @UpdateLastActivity = 1 )
    BEGIN
        UPDATE  dbo.aspnet_Users
        SET      LastActivityDate = @CurrentTimeUtc
        FROM      dbo.aspnet_Users
        WHERE     @UserId = UserId

        IF ( @@ROWCOUNT = 0 ) -- User ID not found
            RETURN -1
    END

    SELECT  m.Email, m.PasswordQuestion, m.Comment, m.IsApproved,
            m.CreateDate, m.LastLoginDate, u.LastActivityDate,
            m.LastPasswordChangedDate, u.UserName, m.IsLockedOut,
            m.LastLockoutDate
    FROM      dbo.aspnet_Users u, dbo.aspnet_Membership m
    WHERE     @UserId = u.UserId AND u.UserId = m.UserId

    IF ( @@ROWCOUNT = 0 ) -- User ID not found
        RETURN -1

    RETURN 0
END

```

8.15 - RESETPASSWORD

```
ALTER PROCEDURE dbo.aspnet_Membership_ResetPassword
    @ApplicationName    nvarchar(256),
    @UserName            nvarchar(256),
    @NewPassword         nvarchar(128),
    @MaxInvalidPasswordAttempts int,
    @PasswordAttemptWindow int,
    @PasswordSalt        nvarchar(128),
    @CurrentTimeUtc      datetime,
    @PasswordFormat      int = 0,
    @PasswordAnswer      nvarchar(128) = NULL
AS
BEGIN
    DECLARE @IsLockedOut bit
    DECLARE @LastLockoutDate datetime
    DECLARE @FailedPasswordAttemptCount int
    DECLARE @FailedPasswordAttemptWindowStart datetime
    DECLARE @FailedPasswordAnswerAttemptCount int
    DECLARE @FailedPasswordAnswerAttemptWindowStart datetime

    DECLARE @UserId uniqueidentifier
    SET @UserId = NULL

    DECLARE @ErrorCode int
    SET @ErrorCode = 0

    DECLARE @TranStarted bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END
    ELSE
        SET @TranStarted = 0

    SELECT @UserId = u.UserId
    FROM    dbo.aspnet_Users u, dbo.aspnet_Applications a, dbo.aspnet_Membership m
    WHERE   LoweredUserName = LOWER(@UserName) AND
            u.ApplicationId = a.ApplicationId AND
            LOWER(@ApplicationName) = a.LoweredApplicationName AND
            u.UserId = m.UserId

    IF ( @UserId IS NULL )
    BEGIN
        SET @ErrorCode = 1
        GOTO Cleanup
    END
```

END

```
SELECT @IsLockedOut = IsLockedOut,  
       @LastLockoutDate = LastLockoutDate,  
       @FailedPasswordAttemptCount = FailedPasswordAttemptCount,  
       @FailedPasswordAttemptWindowStart = FailedPasswordAttemptWindowStart,  
       @FailedPasswordAnswerAttemptCount = FailedPasswordAnswerAttemptCount,  
       @FailedPasswordAnswerAttemptWindowStart =  
FailedPasswordAnswerAttemptWindowStart  
FROM dbo.aspnet_Membership WITH ( UPDLOCK )  
WHERE @UserId = UserId
```

```
IF( @IsLockedOut = 1 )  
BEGIN  
    SET @ErrorCode = 99  
    GOTO Cleanup  
END
```

```
UPDATE dbo.aspnet_Membership  
SET Password = @NewPassword,  
    LastPasswordChangedDate = @CurrentTimeUtc,  
    PasswordFormat = @PasswordFormat,  
    PasswordSalt = @PasswordSalt  
WHERE @UserId = UserId AND  
    ( ( @PasswordAnswer IS NULL ) OR ( LOWER( PasswordAnswer ) =  
LOWER( @PasswordAnswer ) ) )
```

```
IF ( @@ROWCOUNT = 0 )  
BEGIN  
    IF( @CurrentTimeUtc > DATEADD( minute, @PasswordAttemptWindow,  
@FailedPasswordAnswerAttemptWindowStart ) )  
        BEGIN  
            SET @FailedPasswordAnswerAttemptWindowStart = @CurrentTimeUtc  
            SET @FailedPasswordAnswerAttemptCount = 1  
        END  
    ELSE  
        BEGIN  
            SET @FailedPasswordAnswerAttemptWindowStart = @CurrentTimeUtc  
            SET @FailedPasswordAnswerAttemptCount =  
@FailedPasswordAnswerAttemptCount + 1  
        END  
END
```

```
BEGIN  
    IF( @FailedPasswordAnswerAttemptCount >=  
@MaxInvalidPasswordAttempts )  
        BEGIN  
            SET @IsLockedOut = 1  
            SET @LastLockoutDate = @CurrentTimeUtc  
        END  
END
```

```

        SET @ErrorCode = 3
    END
ELSE
    BEGIN
        IF( @FailedPasswordAnswerAttemptCount > 0 )
            BEGIN
                SET @FailedPasswordAnswerAttemptCount = 0
                SET @FailedPasswordAnswerAttemptWindowStart = CONVERT( datetime,
'17540101', 112 )
            END
        END
    END

```

```

IF( NOT ( @PasswordAnswer IS NULL ) )
BEGIN
    UPDATE dbo.aspnet_Membership
    SET IsLockedOut = @IsLockedOut, LastLockoutDate = @LastLockoutDate,
        FailedPasswordAttemptCount = @FailedPasswordAttemptCount,
        FailedPasswordAttemptWindowStart = @FailedPasswordAttemptWindowStart,
        FailedPasswordAnswerAttemptCount =
@FailedPasswordAnswerAttemptCount,
        FailedPasswordAnswerAttemptWindowStart =
@FailedPasswordAnswerAttemptWindowStart
    WHERE @UserId = UserId

```

```

    IF( @@ERROR <> 0 )
        BEGIN
            SET @ErrorCode = -1
            GOTO Cleanup
        END
    END

```

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    COMMIT TRANSACTION
END

```

```

RETURN @ErrorCode

```

Cleanup:

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END

```

```

RETURN @ErrorCode

```


END

8.16 - SETPASSWORD

```
ALTER PROCEDURE dbo.aspnet_Membership_SetPassword
    @ApplicationName nvarchar(256),
    @UserName        nvarchar(256),
    @NewPassword     nvarchar(128),
    @PasswordSalt    nvarchar(128),
    @CurrentTimeUtc  datetime,
    @PasswordFormat  int = 0
AS
BEGIN
    DECLARE @UserId uniqueidentifier
    SELECT @UserId = NULL
    SELECT @UserId = u.UserId
    FROM    dbo.aspnet_Users u, dbo.aspnet_Applications a, dbo.aspnet_Membership m
    WHERE   LoweredUserName = LOWER(@UserName) AND
            u.ApplicationId = a.ApplicationId AND
            LOWER(@ApplicationName) = a.LoweredApplicationName AND
            u.UserId = m.UserId

    IF (@UserId IS NULL)
        RETURN(1)

    UPDATE dbo.aspnet_Membership
    SET Password = @NewPassword, PasswordFormat = @PasswordFormat,
        PasswordSalt = @PasswordSalt,
        LastPasswordChangedDate = @CurrentTimeUtc
    WHERE @UserId = UserId
    RETURN(0)
END
```

8.17 - UNLOCKUSERS

```

ALTER PROCEDURE dbo.aspnet_Membership_UnlockUser
    @ApplicationName      nvarchar(256),
    @UserName              nvarchar(256)
AS
BEGIN
    DECLARE @UserId uniqueidentifier
    SELECT @UserId = NULL
    SELECT @UserId = u.UserId
    FROM    dbo.aspnet_Users u, dbo.aspnet_Applications a, dbo.aspnet_Membership m
    WHERE   LoweredUserName = LOWER(@UserName) AND
            u.ApplicationId = a.ApplicationId AND
            LOWER(@ApplicationName) = a.LoweredApplicationName AND
            u.UserId = m.UserId

    IF ( @UserId IS NULL )
        RETURN 1

    UPDATE dbo.aspnet_Membership
    SET IsLockedOut = 0,
        FailedPasswordAttemptCount = 0,
        FailedPasswordAttemptWindowStart = CONVERT( datetime, '17540101', 112 ),
        FailedPasswordAnswerAttemptCount = 0,
        FailedPasswordAnswerAttemptWindowStart = CONVERT( datetime, '17540101',
112 ),
        LastLockoutDate = CONVERT( datetime, '17540101', 112 )
    WHERE @UserId = UserId

    RETURN 0
END

```

8.18 - UPDATEUSERS

```

ALTER PROCEDURE dbo.aspnet_Membership_UpdateUser
    @ApplicationName      nvarchar(256),
    @UserName              nvarchar(256),
    @Email                 nvarchar(256),
    @Comment               ntext,
    @IsApproved            bit,
    @LastLoginDate         datetime,
    @LastActivityDate      datetime,
    @UniqueEmail           int,
    @CurrentTimeUtc        datetime
AS
BEGIN

```

```

DECLARE @UserId uniqueidentifier
DECLARE @ApplicationId uniqueidentifier
SELECT @UserId = NULL
SELECT @UserId = u.UserId, @ApplicationId = a.ApplicationId
FROM   dbo.aspnet_Users u, dbo.aspnet_Applications a, dbo.aspnet_Membership m
WHERE  LoweredUserName = LOWER(@UserName) AND
       u.ApplicationId = a.ApplicationId AND
       LOWER(@ApplicationName) = a.LoweredApplicationName AND
       u.UserId = m.UserId

IF (@UserId IS NULL)
    RETURN(1)

IF (@UniqueEmail = 1)
BEGIN
    IF (EXISTS (SELECT *
                FROM   dbo.aspnet_Membership WITH (UPDLOCK, HOLDLOCK)
                WHERE  ApplicationId = @ApplicationId AND @UserId <> UserId AND
LoweredEmail = LOWER(@Email)))
        BEGIN
            RETURN(7)
        END
    END
END

DECLARE @TranStarted bit
SET @TranStarted = 0

IF( @@TRANCOUNT = 0 )
BEGIN
    BEGIN TRANSACTION
    SET @TranStarted = 1
END
ELSE
    SET @TranStarted = 0

UPDATE dbo.aspnet_Users WITH (ROWLOCK)
SET
    LastActivityDate = @LastActivityDate
WHERE
    @UserId = UserId

IF( @@ERROR <> 0 )
    GOTO Cleanup

UPDATE dbo.aspnet_Membership WITH (ROWLOCK)
SET
    Email           = @Email,
    LoweredEmail    = LOWER(@Email),
    Comment         = @Comment,
    IsApproved      = @IsApproved,

```

```

    LastLoginDate = @LastLoginDate
WHERE
    @UserId = UserId

IF( @@ERROR <> 0 )
    GOTO Cleanup

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    COMMIT TRANSACTION
END

RETURN 0

```

Cleanup:

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END

RETURN -1
END

```

8.19 - UPDATEUSERINFO

```

ALTER PROCEDURE dbo.aspnet_Membership_UpdateUserInfo
    @ApplicationName      nvarchar(256),
    @UserName              nvarchar(256),
    @IsPasswordCorrect     bit,
    @UpdateLastLoginActivityDate bit,
    @MaxInvalidPasswordAttempts int,
    @PasswordAttemptWindow int,
    @CurrentTimeUtc        datetime,
    @LastLoginDate         datetime,
    @LastActivityDate      datetime

```



```

AS
BEGIN
    DECLARE @UserId                uniqueidentifier
    DECLARE @IsApproved            bit
    DECLARE @IsLockedOut          bit
    DECLARE @LastLockoutDate       datetime
    DECLARE @FailedPasswordAttemptCount    int
    DECLARE @FailedPasswordAttemptWindowStart    datetime
    DECLARE @FailedPasswordAnswerAttemptCount    int
    DECLARE @FailedPasswordAnswerAttemptWindowStart    datetime

    DECLARE @ErrorCode    int
    SET @ErrorCode = 0

    DECLARE @TranStarted    bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END
    ELSE
        SET @TranStarted = 0

    SELECT @UserId = u.UserId,
           @IsApproved = m.IsApproved,
           @IsLockedOut = m.IsLockedOut,
           @LastLockoutDate = m.LastLockoutDate,
           @FailedPasswordAttemptCount = m.FailedPasswordAttemptCount,
           @FailedPasswordAttemptWindowStart =
m.FailedPasswordAttemptWindowStart,
           @FailedPasswordAnswerAttemptCount =
m.FailedPasswordAnswerAttemptCount,
           @FailedPasswordAnswerAttemptWindowStart =
m.FailedPasswordAnswerAttemptWindowStart
    FROM    dbo.aspnet_Applications a, dbo.aspnet_Users u, dbo.aspnet_Membership m
    WITH ( UPDLOCK )
    WHERE   LOWER(@ApplicationName) = a.LoweredApplicationName AND
           u.ApplicationId = a.ApplicationId AND
           u.UserId = m.UserId AND
           LOWER(@UserName) = u.LoweredUserName

    IF ( @@rowcount = 0 )
    BEGIN
        SET @ErrorCode = 1
        GOTO Cleanup
    END

    IF( @IsLockedOut = 1 )

```

```

BEGIN
    GOTO Cleanup
END

IF( @IsPasswordCorrect = 0 )
BEGIN
    IF( @CurrentTimeUtc > DATEADD( minute, @PasswordAttemptWindow,
@FailedPasswordAttemptWindowStart ) )
        BEGIN
            SET @FailedPasswordAttemptWindowStart = @CurrentTimeUtc
            SET @FailedPasswordAttemptCount = 1
        END
    ELSE
        BEGIN
            SET @FailedPasswordAttemptWindowStart = @CurrentTimeUtc
            SET @FailedPasswordAttemptCount = @FailedPasswordAttemptCount + 1
        END
    END

    BEGIN
        IF( @FailedPasswordAttemptCount >= @MaxInvalidPasswordAttempts )
        BEGIN
            SET @IsLockedOut = 1
            SET @LastLockoutDate = @CurrentTimeUtc
        END
    END

    END
ELSE
    BEGIN
        IF( @FailedPasswordAttemptCount > 0 OR
@FailedPasswordAnswerAttemptCount > 0 )
        BEGIN
            SET @FailedPasswordAttemptCount = 0
            SET @FailedPasswordAttemptWindowStart = CONVERT( datetime,
'17540101', 112 )
            SET @FailedPasswordAnswerAttemptCount = 0
            SET @FailedPasswordAnswerAttemptWindowStart = CONVERT( datetime,
'17540101', 112 )
            SET @LastLockoutDate = CONVERT( datetime, '17540101', 112 )
        END
    END

    END

    IF( @UpdateLastLoginActivityDate = 1 )
    BEGIN
        UPDATE dbo.aspnet_Users
        SET LastActivityDate = @LastActivityDate
        WHERE @UserId = UserId

        IF( @@ERROR <> 0 )
        BEGIN
            SET @ErrorCode = -1
        END
    END

```

```
GOTO Cleanup
END
```

```
UPDATE dbo.aspnet_Membership
SET LastLoginDate = @LastLoginDate
WHERE UserId = @UserId
```

```
IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup
END
END
```

```
UPDATE dbo.aspnet_Membership
SET IsLockedOut = @IsLockedOut, LastLockoutDate = @LastLockoutDate,
    FailedPasswordAttemptCount = @FailedPasswordAttemptCount,
    FailedPasswordAttemptWindowStart = @FailedPasswordAttemptWindowStart,
    FailedPasswordAnswerAttemptCount = @FailedPasswordAnswerAttemptCount,
    FailedPasswordAnswerAttemptWindowStart =
@FailedPasswordAnswerAttemptWindowStart
WHERE @UserId = UserId
```

```
IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup
END
```

```
IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    COMMIT TRANSACTION
END
```

```
RETURN @ErrorCode
```

Cleanup:

```
IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END
```

```
RETURN @ErrorCode
```

```
END
```

8.20 - CREATEPATH

```
ALTER PROCEDURE dbo.aspnet_Paths_CreatePath
    @ApplicationId UNIQUEIDENTIFIER,
    @Path          NVARCHAR(256),
    @PathId        UNIQUEIDENTIFIER OUTPUT
AS
BEGIN
    BEGIN TRANSACTION
    IF (NOT EXISTS(SELECT * FROM dbo.aspnet_Paths WHERE LoweredPath =
LOWER(@Path) AND ApplicationId = @ApplicationId))
        BEGIN
            INSERT dbo.aspnet_Paths (ApplicationId, Path, LoweredPath) VALUES
(@ApplicationId, @Path, LOWER(@Path))
        END
    COMMIT TRANSACTION
    SELECT @PathId = PathId FROM dbo.aspnet_Paths WHERE LOWER(@Path) =
LoweredPath AND ApplicationId = @ApplicationId
END
```

8.21 - GETAPPLICATIONID

```
ALTER PROCEDURE dbo.aspnet_Personalization_GetApplicationId (
    @ApplicationName NVARCHAR(256),
    @ApplicationId UNIQUEIDENTIFIER OUT)
AS
BEGIN
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
END
```


8.22 - DELETEALLSTATE

```
ALTER PROCEDURE dbo.aspnet_PersonalizationAdministration_DeleteAllState (
    @AllUsersScope bit,
    @ApplicationName NVARCHAR(256),
    @Count int OUT)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
    @ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
        SELECT @Count = 0
    ELSE
        BEGIN
            IF (@AllUsersScope = 1)
                DELETE FROM aspnet_PersonalizationAllUsers
                WHERE PathId IN
                (SELECT Paths.PathId
                 FROM dbo.aspnet_Paths Paths
                 WHERE Paths.ApplicationId = @ApplicationId)
            ELSE
                DELETE FROM aspnet_PersonalizationPerUser
                WHERE PathId IN
                (SELECT Paths.PathId
                 FROM dbo.aspnet_Paths Paths
                 WHERE Paths.ApplicationId = @ApplicationId)

            SELECT @Count = @@ROWCOUNT
        END
    END
END
```

8.23 - FINDSTATE

```
ALTER PROCEDURE dbo.aspnet_PersonalizationAdministration_FindState (
    @AllUsersScope bit,
    @ApplicationName NVARCHAR(256),
    @PageIndex      INT,
    @PageSize       INT,
    @Path NVARCHAR(256) = NULL,
    @UserName NVARCHAR(256) = NULL,
```

```

    @InactiveSinceDate DATETIME = NULL)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
    @ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
        RETURN

    -- Set the page bounds
    DECLARE @PageLowerBound INT
    DECLARE @PageUpperBound INT
    DECLARE @TotalRecords INT
    SET @PageLowerBound = @PageSize * @PageIndex
    SET @PageUpperBound = @PageSize - 1 + @PageLowerBound

    -- Create a temp table to store the selected results
    CREATE TABLE #PageIndex (
        IndexId int IDENTITY (0, 1) NOT NULL,
        ItemId UNIQUEIDENTIFIER
    )

    IF (@AllUsersScope = 1)
    BEGIN
        -- Insert into our temp table
        INSERT INTO #PageIndex (ItemId)
        SELECT Paths.PathId
        FROM dbo.aspnet_Paths Paths,
        ((SELECT Paths.PathId
        FROM dbo.aspnet_PersonalizationAllUsers AllUsers, dbo.aspnet_Paths Paths
        WHERE Paths.ApplicationId = @ApplicationId
            AND AllUsers.PathId = Paths.PathId
            AND (@Path IS NULL OR Paths.LoweredPath LIKE LOWER(@Path))
        ) AS SharedDataPerPath
        FULL OUTER JOIN
        (SELECT DISTINCT Paths.PathId
        FROM dbo.aspnet_PersonalizationPerUser PerUser, dbo.aspnet_Paths Paths
        WHERE Paths.ApplicationId = @ApplicationId
            AND PerUser.PathId = Paths.PathId
            AND (@Path IS NULL OR Paths.LoweredPath LIKE LOWER(@Path))
        ) AS UserDataPerPath
        ON SharedDataPerPath.PathId = UserDataPerPath.PathId
        )
        WHERE Paths.PathId = SharedDataPerPath.PathId OR Paths.PathId =
        UserDataPerPath.PathId
        ORDER BY Paths.Path ASC

    SELECT @TotalRecords = @@ROWCOUNT

    SELECT Paths.Path,

```

```

        SharedDataPerPath.LastUpdatedDate,
        SharedDataPerPath.SharedDataLength,
        UserDataPerPath.UserDataLength,
        UserDataPerPath.UserCount
    FROM dbo.aspnet_Paths Paths,
        ((SELECT PageIndex.ItemId AS PathId,
            AllUsers.LastUpdatedDate AS LastUpdatedDate,
            DATALENGTH(AllUsers.PageSettings) AS SharedDataLength
        FROM dbo.aspnet_PersonalizationAllUsers AllUsers, #PageIndex PageIndex
        WHERE AllUsers.PathId = PageIndex.ItemId
            AND PageIndex.IndexId >= @PageLowerBound AND PageIndex.IndexId
<= @PageUpperBound
        ) AS SharedDataPerPath
    FULL OUTER JOIN
        (SELECT PageIndex.ItemId AS PathId,
            SUM(DATALENGTH(PerUser.PageSettings)) AS UserDataLength,
            COUNT(*) AS UserCount
        FROM aspnet_PersonalizationPerUser PerUser, #PageIndex PageIndex
        WHERE PerUser.PathId = PageIndex.ItemId
            AND PageIndex.IndexId >= @PageLowerBound AND PageIndex.IndexId
<= @PageUpperBound
        GROUP BY PageIndex.ItemId
        ) AS UserDataPerPath
    ON SharedDataPerPath.PathId = UserDataPerPath.PathId
    )
    WHERE Paths.PathId = SharedDataPerPath.PathId OR Paths.PathId =
UserDataPerPath.PathId
    ORDER BY Paths.Path ASC
END
ELSE
BEGIN
    -- Insert into our temp table
    INSERT INTO #PageIndex (ItemId)
    SELECT PerUser.Id
    FROM dbo.aspnet_PersonalizationPerUser PerUser, dbo.aspnet_Users Users,
dbo.aspnet_Paths Paths
    WHERE Paths.ApplicationId = @ApplicationId
        AND PerUser.UserId = Users.UserId
        AND PerUser.PathId = Paths.PathId
        AND (@Path IS NULL OR Paths.LoweredPath LIKE LOWER(@Path))
        AND (@UserName IS NULL OR Users.LoweredUserName LIKE
LOWER(@UserName))
        AND (@InactiveSinceDate IS NULL OR Users.LastActivityDate <=
@InactiveSinceDate)
    ORDER BY Paths.Path ASC, Users.UserName ASC

    SELECT @TotalRecords = @@ROWCOUNT

    SELECT Paths.Path, PerUser.LastUpdatedDate,
DATALENGTH(PerUser.PageSettings), Users.UserName, Users.LastActivityDate

```

```

FROM dbo.aspnet_PersonalizationPerUser PerUser, dbo.aspnet_Users Users,
dbo.aspnet_Paths Paths, #PageIndex PageIndex
WHERE PerUser.Id = PageIndex.ItemId
AND PerUser.UserId = Users.UserId
AND PerUser.PathId = Paths.PathId
AND PageIndex.IndexId >= @PageLowerBound AND PageIndex.IndexId <=
@PageUpperBound
ORDER BY Paths.Path ASC, Users.UserName ASC
END

RETURN @TotalRecords
END

```

8.24 - GETCOUNTOFSTATE

```

ALTER PROCEDURE dbo.aspnet_PersonalizationAdministration_GetCountOfState (
    @Count int OUT,
    @AllUsersScope bit,
    @ApplicationName NVARCHAR(256),
    @Path NVARCHAR(256) = NULL,
    @UserName NVARCHAR(256) = NULL,
    @InactiveSinceDate DATETIME = NULL)
AS
BEGIN

    DECLARE @ApplicationId UNIQUEIDENTIFIER
    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
    @ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
        SELECT @Count = 0
    ELSE
        IF (@AllUsersScope = 1)
            SELECT @Count = COUNT(*)
            FROM dbo.aspnet_PersonalizationAllUsers AllUsers, dbo.aspnet_Paths Paths
            WHERE Paths.ApplicationId = @ApplicationId
            AND AllUsers.PathId = Paths.PathId
            AND (@Path IS NULL OR Paths.LoweredPath LIKE LOWER(@Path))
        ELSE
            SELECT @Count = COUNT(*)
            FROM dbo.aspnet_PersonalizationPerUser PerUser, dbo.aspnet_Users Users,
            dbo.aspnet_Paths Paths
            WHERE Paths.ApplicationId = @ApplicationId
            AND PerUser.UserId = Users.UserId
            AND PerUser.PathId = Paths.PathId

```



```

        AND (@Path IS NULL OR Paths.LoweredPath LIKE LOWER(@Path))
        AND (@UserName IS NULL OR Users.LoweredUserName LIKE
LOWER(@UserName))
        AND (@InactiveSinceDate IS NULL OR Users.LastActivityDate <=
@InactiveSinceDate)
END

```

8.25 - RESETSHAREDSTATE

```

ALTER PROCEDURE dbo.aspnet_PersonalizationAdministration_ResetSharedState (
    @Count int OUT,
    @ApplicationName NVARCHAR(256),
    @Path NVARCHAR(256))
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
@ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
        SELECT @Count = 0
    ELSE
        BEGIN
            DELETE FROM dbo.aspnet_PersonalizationAllUsers
            WHERE PathId IN
                (SELECT AllUsers.PathId
                FROM dbo.aspnet_PersonalizationAllUsers AllUsers, dbo.aspnet_Paths Paths
                WHERE Paths.ApplicationId = @ApplicationId
                AND AllUsers.PathId = Paths.PathId
                AND Paths.LoweredPath = LOWER(@Path))

            SELECT @Count = @@ROWCOUNT
        END
END

```

8.26 - RESETUSERSTATE

```

ALTER PROCEDURE dbo.aspnet_PersonalizationAdministration_ResetUserState (

```

```

@Count          int          OUT,
@ApplicationName NVARCHAR(256),
@InactiveSinceDate DATETIME   = NULL,
@UserName        NVARCHAR(256) = NULL,
@Path           NVARCHAR(256) = NULL)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
@ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
        SELECT @Count = 0
    ELSE
        BEGIN
            DELETE FROM dbo.aspnet_PersonalizationPerUser
            WHERE Id IN (SELECT PerUser.Id
                        FROM dbo.aspnet_PersonalizationPerUser PerUser, dbo.aspnet_Users
                        Users, dbo.aspnet_Paths Paths
                        WHERE Paths.ApplicationId = @ApplicationId
                           AND PerUser.UserId = Users.UserId
                           AND PerUser.PathId = Paths.PathId
                           AND (@InactiveSinceDate IS NULL OR Users.LastActivityDate <=
@InactiveSinceDate)
                           AND (@UserName IS NULL OR Users.LoweredUserName =
LOWER(@UserName))
                           AND (@Path IS NULL OR Paths.LoweredPath = LOWER(@Path)))

            SELECT @Count = @@ROWCOUNT
        END
    END
END

```

8.27 - GETPAGESETTINGS

```

ALTER PROCEDURE dbo.aspnet_PersonalizationAllUsers_GetPageSettings (
    @ApplicationName NVARCHAR(256),
    @Path           NVARCHAR(256))
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    DECLARE @PathId UNIQUEIDENTIFIER

    SELECT @ApplicationId = NULL
    SELECT @PathId = NULL

```

```

EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
@ApplicationId OUTPUT
IF (@ApplicationId IS NULL)
BEGIN
    RETURN
END

SELECT @PathId = u.PathId FROM dbo.aspnet_Paths u WHERE u.ApplicationId =
@ApplicationId AND u.LoweredPath = LOWER(@Path)
IF (@PathId IS NULL)
BEGIN
    RETURN
END

SELECT p.PageSettings FROM dbo.aspnet_PersonalizationAllUsers p WHERE
p.PathId = @PathId
END

```

8.28 - RESETPAGESETTINGS

```

ALTER PROCEDURE dbo.aspnet_PersonalizationAllUsers_ResetPageSettings (
    @ApplicationName NVARCHAR(256),
    @Path            NVARCHAR(256))
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    DECLARE @PathId UNIQUEIDENTIFIER

    SELECT @ApplicationId = NULL
    SELECT @PathId = NULL

    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
@ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
    BEGIN
        RETURN
    END

    SELECT @PathId = u.PathId FROM dbo.aspnet_Paths u WHERE u.ApplicationId =
@ApplicationId AND u.LoweredPath = LOWER(@Path)
    IF (@PathId IS NULL)
    BEGIN
        RETURN
    END
END

```

```

DELETE FROM dbo.aspnet_PersonalizationAllUsers WHERE PathId = @PathId
RETURN 0
END

```

8.29 - SETPAGESETTINGS

```

ALTER PROCEDURE dbo.aspnet_PersonalizationAllUsers_SetPageSettings (
    @ApplicationName NVARCHAR(256),
    @Path            NVARCHAR(256),
    @PageSettings    IMAGE,
    @CurrentTimeUtc  DATETIME)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    DECLARE @PathId UNIQUEIDENTIFIER

    SELECT @ApplicationId = NULL
    SELECT @PathId = NULL

    EXEC dbo.aspnet_Applications_CreateApplication @ApplicationName,
    @ApplicationId OUTPUT

    SELECT @PathId = u.PathId FROM dbo.aspnet_Paths u WHERE u.ApplicationId =
    @ApplicationId AND u.LoweredPath = LOWER(@Path)
    IF (@PathId IS NULL)
    BEGIN
        EXEC dbo.aspnet_Paths_CreatePath @ApplicationId, @Path, @PathId OUTPUT
    END

    IF (EXISTS(SELECT PathId FROM dbo.aspnet_PersonalizationAllUsers WHERE
    PathId = @PathId))
        UPDATE dbo.aspnet_PersonalizationAllUsers SET PageSettings =
    @PageSettings, LastUpdatedDate = @CurrentTimeUtc WHERE PathId = @PathId
    ELSE
        INSERT INTO dbo.aspnet_PersonalizationAllUsers(PathId, PageSettings,
    LastUpdatedDate) VALUES (@PathId, @PageSettings, @CurrentTimeUtc)
    RETURN 0
END

```


8.30 - PERUSERGETPAGESETTINGS

```
ALTER PROCEDURE dbo.aspnet_PersonalizationPerUser_GetPageSettings (
    @ApplicationName NVARCHAR(256),
    @UserName        NVARCHAR(256),
    @Path            NVARCHAR(256),
    @CurrentTimeUtc  DATETIME)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    DECLARE @PathId UNIQUEIDENTIFIER
    DECLARE @UserId UNIQUEIDENTIFIER

    SELECT @ApplicationId = NULL
    SELECT @PathId = NULL
    SELECT @UserId = NULL

    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
    @ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
    BEGIN
        RETURN
    END

    SELECT @PathId = u.PathId FROM dbo.aspnet_Paths u WHERE u.ApplicationId =
    @ApplicationId AND u.LoweredPath = LOWER(@Path)
    IF (@PathId IS NULL)
    BEGIN
        RETURN
    END

    SELECT @UserId = u.UserId FROM dbo.aspnet_Users u WHERE u.ApplicationId
    = @ApplicationId AND u.LoweredUserName = LOWER(@UserName)
    IF (@UserId IS NULL)
    BEGIN
        RETURN
    END

    UPDATE  dbo.aspnet_Users WITH (ROWLOCK)
    SET     LastActivityDate = @CurrentTimeUtc
    WHERE   UserId = @UserId
    IF (@@ROWCOUNT = 0) -- Username not found
        RETURN
```

```

SELECT p.PageSettings FROM dbo.aspnet_PersonalizationPerUser p WHERE
p.PathId = @PathId AND p.UserId = @UserId
END

```

8.31 - PERUSERRESETPAGESETTINGS

```

ALTER PROCEDURE dbo.aspnet_PersonalizationPerUser_ResetPageSettings (
    @ApplicationName NVARCHAR(256),
    @UserName        NVARCHAR(256),
    @Path            NVARCHAR(256),
    @CurrentTimeUtc  DATETIME)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    DECLARE @PathId UNIQUEIDENTIFIER
    DECLARE @UserId UNIQUEIDENTIFIER

    SELECT @ApplicationId = NULL
    SELECT @PathId = NULL
    SELECT @UserId = NULL

    EXEC dbo.aspnet_Personalization_GetApplicationId @ApplicationName,
    @ApplicationId OUTPUT
    IF (@ApplicationId IS NULL)
    BEGIN
        RETURN
    END

    SELECT @PathId = u.PathId FROM dbo.aspnet_Paths u WHERE u.ApplicationId =
    @ApplicationId AND u.LoweredPath = LOWER(@Path)
    IF (@PathId IS NULL)
    BEGIN
        RETURN
    END

    SELECT @UserId = u.UserId FROM dbo.aspnet_Users u WHERE u.ApplicationId
    = @ApplicationId AND u.LoweredUserName = LOWER(@UserName)
    IF (@UserId IS NULL)
    BEGIN
        RETURN
    END
END

```

```

UPDATE  dbo.aspnet_Users WITH (ROWLOCK)
SET     LastActivityDate = @CurrentTimeUtc
WHERE   UserId = @UserId
IF (@@ROWCOUNT = 0) -- Username not found
    RETURN

```

```

DELETE FROM dbo.aspnet_PersonalizationPerUser WHERE PathId = @PathId
AND UserId = @UserId
RETURN 0
END

```

8.32 - PERUSERSETPAGESETTINGS

```

ALTER PROCEDURE dbo.aspnet_PersonalizationPerUser_SetPageSettings (
    @ApplicationName NVARCHAR(256),
    @UserName        NVARCHAR(256),
    @Path            NVARCHAR(256),
    @PageSettings    IMAGE,
    @CurrentTimeUtc  DATETIME)
AS
BEGIN
    DECLARE @ApplicationId UNIQUEIDENTIFIER
    DECLARE @PathId UNIQUEIDENTIFIER
    DECLARE @UserId UNIQUEIDENTIFIER

    SELECT @ApplicationId = NULL
    SELECT @PathId = NULL
    SELECT @UserId = NULL

    EXEC dbo.aspnet_Applications_CreateApplication @ApplicationName,
    @ApplicationId OUTPUT

    SELECT @PathId = u.PathId FROM dbo.aspnet_Paths u WHERE u.ApplicationId =
    @ApplicationId AND u.LoweredPath = LOWER(@Path)
    IF (@PathId IS NULL)
    BEGIN
        EXEC dbo.aspnet_Paths_CreatePath @ApplicationId, @Path, @PathId OUTPUT
    END

    SELECT @UserId = u.UserId FROM dbo.aspnet_Users u WHERE u.ApplicationId =
    @ApplicationId AND u.LoweredUserName = LOWER(@UserName)
    IF (@UserId IS NULL)
    BEGIN
        EXEC dbo.aspnet_Users_CreateUser @ApplicationId, @UserName, 0,
    @CurrentTimeUtc, @UserId OUTPUT
    END
END

```

```

UPDATE  dbo.aspnet_Users WITH (ROWLOCK)
SET     LastActivityDate = @CurrentTimeUtc
WHERE   UserId = @UserId
IF (@@ROWCOUNT = 0) -- Username not found
    RETURN

IF (EXISTS(SELECT PathId FROM dbo.aspnet_PersonalizationPerUser WHERE
UserId = @UserId AND PathId = @PathId))
    UPDATE dbo.aspnet_PersonalizationPerUser SET PageSettings = @PageSettings,
LastUpdatedDate = @CurrentTimeUtc WHERE UserId = @UserId AND PathId =
@PathId
ELSE
    INSERT INTO dbo.aspnet_PersonalizationPerUser(UserId, PathId, PageSettings,
LastUpdatedDate) VALUES (@UserId, @PathId, @PageSettings, @CurrentTimeUtc)
    RETURN 0
END

```

8.33 - DELETEINACTIVEUSERS

```

ALTER PROCEDURE dbo.aspnet_Profile_DeleteInactiveProfiles
    @ApplicationName    nvarchar(256),
    @ProfileAuthOptions int,
    @InactiveSinceDate  datetime
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
    BEGIN
        SELECT 0
        RETURN
    END

    DELETE
    FROM  dbo.aspnet_Profile
    WHERE UserId IN
        ( SELECT UserId
          FROM  dbo.aspnet_Users u
          WHERE ApplicationId = @ApplicationId
                AND (LastActivityDate <= @InactiveSinceDate)

```



```

        AND (
            (@ProfileAuthOptions = 2)
            OR (@ProfileAuthOptions = 0 AND IsAnonymous = 1)
            OR (@ProfileAuthOptions = 1 AND IsAnonymous = 0)
        )
    )

    SELECT @@ROWCOUNT
END

```

8.34 - DELETEPROFILES

```

ALTER PROCEDURE dbo.aspnet_Profile_DeleteProfiles
    @ApplicationName    nvarchar(256),
    @UserNames          nvarchar(4000)
AS
BEGIN
    DECLARE @UserName    nvarchar(256)
    DECLARE @CurrentPos  int
    DECLARE @NextPos     int
    DECLARE @NumDeleted  int
    DECLARE @DeletedUser int
    DECLARE @TranStarted bit
    DECLARE @ErrorCode   int

    SET @ErrorCode = 0
    SET @CurrentPos = 1
    SET @NumDeleted = 0
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END
    ELSE
        SET @TranStarted = 0

    WHILE ( @CurrentPos <= LEN(@UserNames))
    BEGIN
        SELECT @NextPos = CHARINDEX(N',', @UserNames, @CurrentPos)
    
```

```

IF (@NextPos = 0 OR @NextPos IS NULL)
    SELECT @NextPos = LEN(@UserNames) + 1

SELECT @UserName = SUBSTRING(@UserNames, @CurrentPos, @NextPos -
@CurrentPos)
SELECT @CurrentPos = @NextPos+1

IF (LEN(@UserName) > 0)
BEGIN
    SELECT @DeletedUser = 0
    EXEC dbo.aspnet_Users_DeleteUser @ApplicationName, @UserName, 4,
@DeletedUser OUTPUT
    IF( @@ERROR <> 0 )
    BEGIN
        SET @ErrorCode = -1
        GOTO Cleanup
    END
    IF (@DeletedUser <> 0)
        SELECT @NumDeleted = @NumDeleted + 1
    END
END
SELECT @NumDeleted
IF (@TranStarted = 1)
BEGIN
    SET @TranStarted = 0
    COMMIT TRANSACTION
END
SET @TranStarted = 0

RETURN 0

Cleanup:
IF (@TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END
RETURN @ErrorCode
END

```

8.35 - GETNUMBEROFINACTIVEUSERS

```

ALTER PROCEDURE dbo.aspnet_Profile_GetNumberOfInactiveProfiles

```

```

    @ApplicationName    nvarchar(256),
    @ProfileAuthOptions int,
    @InactiveSinceDate  datetime
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
    BEGIN
        SELECT 0
        RETURN
    END

    SELECT COUNT(*)
    FROM    dbo.aspnet_Users u, dbo.aspnet_Profile p
    WHERE   ApplicationId = @ApplicationId
        AND u.UserId = p.UserId
        AND (LastActivityDate <= @InactiveSinceDate)
        AND (
            (@ProfileAuthOptions = 2)
            OR (@ProfileAuthOptions = 0 AND IsAnonymous = 1)
            OR (@ProfileAuthOptions = 1 AND IsAnonymous = 0)
        )
END

```

8.36 - GETPROFILES

```

ALTER PROCEDURE dbo.aspnet_Profile_GetProfiles
    @ApplicationName    nvarchar(256),
    @ProfileAuthOptions int,
    @PageIndex          int,
    @PageSize           int,
    @UserNameToMatch    nvarchar(256) = NULL,
    @InactiveSinceDate  datetime      = NULL
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)

```

```

RETURN

-- Set the page bounds
DECLARE @PageLowerBound int
DECLARE @PageUpperBound int
DECLARE @TotalRecords int
SET @PageLowerBound = @PageSize * @PageIndex
SET @PageUpperBound = @PageSize - 1 + @PageLowerBound

-- Create a temp table TO store the select results
CREATE TABLE #PageIndexForUsers
(
    IndexId int IDENTITY (0, 1) NOT NULL,
    UserId uniqueidentifier
)

-- Insert into our temp table
INSERT INTO #PageIndexForUsers (UserId)
SELECT u.UserId
FROM    dbo.aspnet_Users u, dbo.aspnet_Profile p
WHERE   ApplicationId = @ApplicationId
        AND u.UserId = p.UserId
        AND (@InactiveSinceDate IS NULL OR LastActivityDate <=
@InactiveSinceDate)
        AND ( (@ProfileAuthOptions = 2)
            OR (@ProfileAuthOptions = 0 AND IsAnonymous = 1)
            OR (@ProfileAuthOptions = 1 AND IsAnonymous = 0)
        )
        AND (@UserNameToMatch IS NULL OR LoweredUserName LIKE
LOWER(@UserNameToMatch))
ORDER BY UserName

SELECT u.UserName, u.IsAnonymous, u.LastActivityDate, p.LastUpdatedDate,
        DATALENGTH(p.PropertyNames) + DATALENGTH(p.PropertyValuesString)
+ DATALENGTH(p.PropertyValuesBinary)
FROM    dbo.aspnet_Users u, dbo.aspnet_Profile p, #PageIndexForUsers i
WHERE   u.UserId = p.UserId AND p.UserId = i.UserId AND i.IndexId >=
@PageLowerBound AND i.IndexId <= @PageUpperBound

SELECT COUNT(*)
FROM    #PageIndexForUsers

DROP TABLE #PageIndexForUsers
END

```


8.37 - GETPROPERTIES

```
ALTER PROCEDURE dbo.aspnet_Profile_GetProperties
    @ApplicationName    nvarchar(256),
    @UserName           nvarchar(256),
    @CurrentTimeUtc     datetime
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM dbo.aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN

    DECLARE @UserId uniqueidentifier
    SELECT @UserId = NULL

    SELECT @UserId = UserId
    FROM    dbo.aspnet_Users
    WHERE   ApplicationId = @ApplicationId AND LoweredUserName =
LOWER(@UserName)

    IF (@UserId IS NULL)
        RETURN
    SELECT TOP 1 PropertyNames, PropertyValuesString, PropertyValuesBinary
    FROM    dbo.aspnet_Profile
    WHERE   UserId = @UserId

    IF (@@ROWCOUNT > 0)
    BEGIN
        UPDATE dbo.aspnet_Users
        SET    LastActivityDate=@CurrentTimeUtc
        WHERE  UserId = @UserId
    END
END
```

8.38 - SETPROPERTIES

```

ALTER PROCEDURE dbo.aspnet_Profile_SetProperties
    @ApplicationName    nvarchar(256),
    @PropertyNames      ntext,
    @PropertyValuesString ntext,
    @PropertyValuesBinary image,
    @UserName           nvarchar(256),
    @IsUserAnonymous    bit,
    @CurrentTimeUtc     datetime
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL

    DECLARE @ErrorCode    int
    SET @ErrorCode = 0

    DECLARE @TranStarted bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END
    ELSE
        SET @TranStarted = 0

    EXEC dbo.aspnet_Applications_CreateApplication @ApplicationName,
    @ApplicationId OUTPUT

    IF( @@ERROR <> 0 )
    BEGIN
        SET @ErrorCode = -1
        GOTO Cleanup
    END

    DECLARE @UserId uniqueidentifier
    DECLARE @LastActivityDate datetime
    SELECT @UserId = NULL
    SELECT @LastActivityDate = @CurrentTimeUtc

    SELECT @UserId = UserId
    FROM   dbo.aspnet_Users
    WHERE  ApplicationId = @ApplicationId AND LoweredUserName =
    LOWER(@UserName)
    IF (@UserId IS NULL)
        EXEC dbo.aspnet_Users_CreateUser @ApplicationId, @UserName,
    @IsUserAnonymous, @LastActivityDate, @UserId OUTPUT

```

```

IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup
END

```

```

UPDATE dbo.aspnet_Users
SET LastActivityDate=@CurrentTimeUtc
WHERE UserId = @UserId

```

```

IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup
END

```

```

IF (EXISTS( SELECT *
            FROM  dbo.aspnet_Profile
            WHERE UserId = @UserId))
    UPDATE dbo.aspnet_Profile
    SET PropertyNames=@PropertyNames, PropertyValuesString =
@PropertyValuesString,
        PropertyValuesBinary = @PropertyValuesBinary,
LastUpdatedDate=@CurrentTimeUtc
    WHERE UserId = @UserId
ELSE
    INSERT INTO dbo.aspnet_Profile(UserId, PropertyNames, PropertyValuesString,
PropertyValuesBinary, LastUpdatedDate)
        VALUES (@UserId, @PropertyNames, @PropertyValuesString,
@PropertyValuesBinary, @CurrentTimeUtc)

```

```

IF( @@ERROR <> 0 )
BEGIN
    SET @ErrorCode = -1
    GOTO Cleanup
END

```

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    COMMIT TRANSACTION
END

```

```

RETURN 0

```

Cleanup:

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0

```

```
        ROLLBACK TRANSACTION
    END
```

```
    RETURN @ErrorCode
```

```
END
```

8.39 - REGISTERSCHEMAVERSIONS

```
ALTER PROCEDURE [dbo].aspnet_RegisterSchemaVersion
    @Feature          nvarchar(128),
    @CompatibleSchemaVersion nvarchar(128),
    @IsCurrentVersion  bit,
    @RemoveIncompatibleSchema bit
AS
BEGIN
    IF( @RemoveIncompatibleSchema = 1 )
    BEGIN
        DELETE FROM dbo.aspnet_SchemaVersions WHERE Feature = LOWER(
@Feature )
    END
    ELSE
    BEGIN
        IF( @IsCurrentVersion = 1 )
        BEGIN
            UPDATE dbo.aspnet_SchemaVersions
            SET IsCurrentVersion = 0
            WHERE Feature = LOWER( @Feature )
        END
    END

    INSERT dbo.aspnet_SchemaVersions( Feature, CompatibleSchemaVersion,
IsCurrentVersion )
    VALUES( LOWER( @Feature ), @CompatibleSchemaVersion, @IsCurrentVersion
)
END
```

8.40 - CRETAEROLE


```

ALTER PROCEDURE dbo.aspnet_Roles_CreateRole
    @ApplicationName nvarchar(256),
    @RoleName        nvarchar(256)
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL

    DECLARE @ErrorCode    int
    SET @ErrorCode = 0

    DECLARE @TranStarted  bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END
    ELSE
        SET @TranStarted = 0

    EXEC dbo.aspnet_Applications_CreateApplication @ApplicationName,
    @ApplicationId OUTPUT

    IF( @@ERROR <> 0 )
    BEGIN
        SET @ErrorCode = -1
        GOTO Cleanup
    END

    IF (EXISTS(SELECT RoleId FROM dbo.aspnet_Roles WHERE LoweredRoleName
= LOWER(@RoleName) AND ApplicationId = @ApplicationId))
    BEGIN
        SET @ErrorCode = 1
        GOTO Cleanup
    END

    INSERT INTO dbo.aspnet_Roles
        (ApplicationId, RoleName, LoweredRoleName)
        VALUES (@ApplicationId, @RoleName, LOWER(@RoleName))

    IF( @@ERROR <> 0 )
    BEGIN
        SET @ErrorCode = -1
        GOTO Cleanup
    END

    IF( @TranStarted = 1 )
    BEGIN

```

```

    SET @TranStarted = 0
    COMMIT TRANSACTION
END

```

```

RETURN(0)

```

Cleanup:

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END

```

```

RETURN @ErrorCode

```

```

END

```

8.41 - DELETEROLES

```

ALTER PROCEDURE dbo.aspnet_Roles_DeleteRole
    @ApplicationName      nvarchar(256),
    @RoleName             nvarchar(256),
    @DeleteOnlyIfRoleIsEmpty bit
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN(1)

    DECLARE @ErrorCode int
    SET @ErrorCode = 0

    DECLARE @TranStarted bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END

```

```

ELSE
    SET @TranStarted = 0

    DECLARE @RoleId uniqueidentifier
    SELECT @RoleId = NULL
    SELECT @RoleId = RoleId FROM dbo.aspnet_Roles WHERE LoweredRoleName
= LOWER(@RoleName) AND ApplicationId = @ApplicationId

    IF (@RoleId IS NULL)
    BEGIN
        SELECT @ErrorCode = 1
        GOTO Cleanup
    END
    IF (@DeleteOnlyIfRoleIsEmpty <> 0)
    BEGIN
        IF (EXISTS (SELECT RoleId FROM dbo.aspnet_UsersInRoles WHERE
@RoleId = RoleId))
        BEGIN
            SELECT @ErrorCode = 2
            GOTO Cleanup
        END
    END
    END

    DELETE FROM dbo.aspnet_UsersInRoles WHERE @RoleId = RoleId

    IF( @@ERROR <> 0 )
    BEGIN
        SET @ErrorCode = -1
        GOTO Cleanup
    END

    DELETE FROM dbo.aspnet_Roles WHERE @RoleId = RoleId AND ApplicationId
= @ApplicationId

    IF( @@ERROR <> 0 )
    BEGIN
        SET @ErrorCode = -1
        GOTO Cleanup
    END

    IF( @TranStarted = 1 )
    BEGIN
        SET @TranStarted = 0
        COMMIT TRANSACTION
    END

    RETURN(0)

```

Cleanup:

```

IF( @TranStarted = 1 )
BEGIN
    SET @TranStarted = 0
    ROLLBACK TRANSACTION
END

RETURN @ErrorCode
END

```

8.42 - GETALLROLES

```

ALTER PROCEDURE dbo.aspnet_Roles_GetAllRoles (
    @ApplicationName      nvarchar(256))
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN
    SELECT RoleName
    FROM  dbo.aspnet_Roles WHERE ApplicationId = @ApplicationId
    ORDER BY RoleName
END

```

8.43 - ROLEEXISTS

```

ALTER PROCEDURE dbo.aspnet_Roles_RoleExists
    @ApplicationName nvarchar(256),
    @RoleName        nvarchar(256)
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier

```



```

SELECT @ApplicationId = NULL
SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
IF (@ApplicationId IS NULL)
    RETURN(0)
IF (EXISTS (SELECT RoleName FROM dbo.aspnet_Roles WHERE
LOWER(@RoleName) = LoweredRoleName AND ApplicationId = @ApplicationId ))
    RETURN(1)
ELSE
    RETURN(0)
END

```

8.44 - REMOVEALLROLEMEMBERS

```

ALTER PROCEDURE [dbo].aspnet_Setup_RemoveAllRoleMembers
    @name sysname
AS
BEGIN
    CREATE TABLE #aspnet_RoleMembers
    (
        Group_name sysname,
        Group_id smallint,
        Users_in_group sysname,
        User_id smallint
    )

    INSERT INTO #aspnet_RoleMembers
    EXEC sp_helpuser @name

    DECLARE @user_id smallint
    DECLARE @cmd nvarchar(500)
    DECLARE c1 cursor FORWARD_ONLY FOR
        SELECT User_id FROM #aspnet_RoleMembers

    OPEN c1

    FETCH c1 INTO @user_id
    WHILE (@@fetch_status = 0)
    BEGIN
        SET @cmd = 'EXEC sp_droprolemember ' + '''' + @name + ''', '''' +
USER_NAME(@user_id) + ''''
        EXEC (@cmd)
        FETCH c1 INTO @user_id
    END

```

```

END

CLOSE c1
DEALLOCATE c1
END

```

8.45 - RESTOREPERMISSIONS

```

ALTER PROCEDURE [dbo].aspnet_Setup_RestorePermissions
    @name sysname
AS
BEGIN
    DECLARE @object sysname
    DECLARE @protectType char(10)
    DECLARE @action varchar(60)
    DECLARE @grantee sysname
    DECLARE @cmd nvarchar(500)
    DECLARE c1 cursor FORWARD_ONLY FOR
        SELECT Object, ProtectType, [Action], Grantee FROM #aspnet_Permissions
        where Object = @name

    OPEN c1

    FETCH c1 INTO @object, @protectType, @action, @grantee
    WHILE (@@fetch_status = 0)
    BEGIN
        SET @cmd = @protectType + ' ' + @action + ' on ' + @object + ' TO [' +
@grantee + ']'
        EXEC (@cmd)
        FETCH c1 INTO @object, @protectType, @action, @grantee
    END

    CLOSE c1
    DEALLOCATE c1
END

```

8.46 - UNREGISTERSCHEMAVERSION

```

ALTER PROCEDURE [dbo].aspnet_UnRegisterSchemaVersion
    @Feature          nvarchar(128),
    @CompatibleSchemaVersion nvarchar(128)
AS
BEGIN
    DELETE FROM dbo.aspnet_SchemaVersions
        WHERE Feature = LOWER(@Feature) AND @CompatibleSchemaVersion =
CompatibleSchemaVersion
END

```

8.47 - CREATEUSER

```

ALTER PROCEDURE [dbo].aspnet_Users_CreateUser
    @ApplicationId  uniqueidentifier,
    @UserName       nvarchar(256),
    @IsUserAnonymous bit,
    @LastActivityDate DATETIME,
    @UserId         uniqueidentifier OUTPUT
AS
BEGIN
    IF( @UserId IS NULL )
        SELECT @UserId = NEWID()
    ELSE
        BEGIN
            IF( EXISTS( SELECT UserId FROM dbo.aspnet_Users
                        WHERE @UserId = UserId ) )
                RETURN -1
        END
    END

    INSERT dbo.aspnet_Users (ApplicationId, UserId, UserName, LoweredUserName,
IsAnonymous, LastActivityDate)
    VALUES (@ApplicationId, @UserId, @UserName, LOWER(@UserName),
@IsUserAnonymous, @LastActivityDate)

    RETURN 0
END

```

8.48 - DELETEUSER

```
ALTER PROCEDURE [dbo].aspnet_Users_DeleteUser
    @ApplicationName nvarchar(256),
    @UserName        nvarchar(256),
    @TablesToDeleteFrom int,
    @NumTablesDeletedFrom int OUTPUT
AS
BEGIN
    DECLARE @UserId          uniqueidentifier
    SELECT @UserId          = NULL
    SELECT @NumTablesDeletedFrom = 0

    DECLARE @TranStarted bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END
    ELSE
        SET @TranStarted = 0

    DECLARE @ErrorCode int
    DECLARE @RowCount int

    SET @ErrorCode = 0
    SET @RowCount = 0

    SELECT @UserId = u.UserId
    FROM    dbo.aspnet_Users u, dbo.aspnet_Applications a
    WHERE   u.LoweredUserName = LOWER(@UserName)
           AND u.ApplicationId = a.ApplicationId
           AND LOWER(@ApplicationName) = a.LoweredApplicationName

    IF (@UserId IS NULL)
    BEGIN
        GOTO Cleanup
    END

    -- Delete from Membership table if (@TablesToDeleteFrom & 1) is set
    IF ((@TablesToDeleteFrom & 1) <> 0 AND
        (EXISTS (SELECT name FROM sysobjects WHERE (name =
        N'vw_aspnet_MembershipUsers') AND (type = 'V'))))
```



```

BEGIN
    DELETE FROM dbo.aspnet_Membership WHERE @UserId = UserId

    SELECT @ErrorCode = @@ERROR,
           @RowCount = @@ROWCOUNT

    IF( @ErrorCode <> 0 )
        GOTO Cleanup

    IF ( @RowCount <> 0 )
        SELECT @NumTablesDeletedFrom = @NumTablesDeletedFrom + 1
END

-- Delete from aspnet_UsersInRoles table if ( @TablesToDeleteFrom & 2 ) is set
IF ((@TablesToDeleteFrom & 2) <> 0 AND
    (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_UsersInRoles') AND (type = 'V')))) )
BEGIN
    DELETE FROM dbo.aspnet_UsersInRoles WHERE @UserId = UserId

    SELECT @ErrorCode = @@ERROR,
           @RowCount = @@ROWCOUNT

    IF( @ErrorCode <> 0 )
        GOTO Cleanup

    IF ( @RowCount <> 0 )
        SELECT @NumTablesDeletedFrom = @NumTablesDeletedFrom + 1
END

-- Delete from aspnet_Profile table if ( @TablesToDeleteFrom & 4 ) is set
IF ((@TablesToDeleteFrom & 4) <> 0 AND
    (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_Profiles') AND (type = 'V')))) )
BEGIN
    DELETE FROM dbo.aspnet_Profile WHERE @UserId = UserId

    SELECT @ErrorCode = @@ERROR,
           @RowCount = @@ROWCOUNT

    IF( @ErrorCode <> 0 )
        GOTO Cleanup

    IF ( @RowCount <> 0 )
        SELECT @NumTablesDeletedFrom = @NumTablesDeletedFrom + 1
END

-- Delete from aspnet_PersonalizationPerUser table if ( @TablesToDeleteFrom & 8 )
is set
IF ((@TablesToDeleteFrom & 8) <> 0 AND

```

```

        (EXISTS (SELECT name FROM sysobjects WHERE (name =
N'vw_aspnet_WebPartState_User') AND (type = 'V')))) )
    BEGIN
        DELETE FROM dbo.aspnet_PersonalizationPerUser WHERE @UserId = UserId

        SELECT @ErrorCode = @@ERROR,
               @RowCount = @@ROWCOUNT

        IF( @ErrorCode <> 0 )
            GOTO Cleanup

        IF (@RowCount <> 0)
            SELECT @NumTablesDeletedFrom = @NumTablesDeletedFrom + 1
    END

    -- Delete from aspnet_Users table if (@TablesToDeleteFrom & 1,2,4 & 8) are all set
    IF ((@TablesToDeleteFrom & 1) <> 0 AND
        (@TablesToDeleteFrom & 2) <> 0 AND
        (@TablesToDeleteFrom & 4) <> 0 AND
        (@TablesToDeleteFrom & 8) <> 0 AND
        (EXISTS (SELECT UserId FROM dbo.aspnet_Users WHERE @UserId =
UserId)))
    BEGIN
        DELETE FROM dbo.aspnet_Users WHERE @UserId = UserId

        SELECT @ErrorCode = @@ERROR,
               @RowCount = @@ROWCOUNT

        IF( @ErrorCode <> 0 )
            GOTO Cleanup

        IF (@RowCount <> 0)
            SELECT @NumTablesDeletedFrom = @NumTablesDeletedFrom + 1
    END

    IF( @TranStarted = 1 )
    BEGIN
        SET @TranStarted = 0
        COMMIT TRANSACTION
    END

    RETURN 0

Cleanup:
    SET @NumTablesDeletedFrom = 0

    IF( @TranStarted = 1 )
    BEGIN
        SET @TranStarted = 0
        ROLLBACK TRANSACTION
    
```

END

RETURN @ErrorCode

END

8.49 - ADDUSERSTOROLES

```
ALTER PROCEDURE dbo.aspnet_UsersInRoles_AddUsersToRoles
    @ApplicationName nvarchar(256),
    @UserNames       nvarchar(4000),
    @RoleNames       nvarchar(4000),
    @CurrentTimeUtc  datetime
AS
BEGIN
    DECLARE @AppId uniqueidentifier
    SELECT @AppId = NULL
    SELECT @AppId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@AppId IS NULL)
        RETURN(2)
    DECLARE @TranStarted bit
    SET @TranStarted = 0

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END

    DECLARE @tbNames      table(Name nvarchar(256) NOT NULL PRIMARY
KEY)
    DECLARE @tbRolestable(RoleId uniqueidentifier NOT NULL PRIMARY
KEY)
    DECLARE @tbUserstable(UserId uniqueidentifier NOT NULL PRIMARY
KEY)
    DECLARE @Num          int
    DECLARE @Pos          int
    DECLARE @NextPos      int
    DECLARE @Name         nvarchar(256)

    SET @Num = 0
    SET @Pos = 1
```

```

WHILE(@Pos <= LEN(@RoleNames))
BEGIN
    SELECT @NextPos = CHARINDEX(N',', @RoleNames, @Pos)
    IF (@NextPos = 0 OR @NextPos IS NULL)
        SELECT @NextPos = LEN(@RoleNames) + 1
    SELECT @Name = RTRIM(LTRIM(SUBSTRING(@RoleNames,
@Pos, @NextPos - @Pos)))
    SELECT @Pos = @NextPos+1

    INSERT INTO @tbNames VALUES (@Name)
    SET @Num = @Num + 1
END

INSERT INTO @tbRoles
SELECT RoleId
FROM   dbo.aspnet_Roles ar, @tbNames t
WHERE  LOWER(t.Name) = ar.LoweredRoleName AND ar.ApplicationId =
@AppId

IF (@@ROWCOUNT <> @Num)
BEGIN
    SELECT TOP 1 Name
    FROM   @tbNames
    WHERE  LOWER(Name) NOT IN (SELECT ar.LoweredRoleName
FROM   dbo.aspnet_Roles ar, @tbRoles r WHERE r.RoleId = ar.RoleId)
    IF( @TranStarted = 1 )
        ROLLBACK TRANSACTION
    RETURN(2)
END

DELETE FROM @tbNames WHERE 1=1
SET @Num = 0
SET @Pos = 1

WHILE(@Pos <= LEN(@UserNames))
BEGIN
    SELECT @NextPos = CHARINDEX(N',', @UserNames, @Pos)
    IF (@NextPos = 0 OR @NextPos IS NULL)
        SELECT @NextPos = LEN(@UserNames) + 1
    SELECT @Name = RTRIM(LTRIM(SUBSTRING(@UserNames,
@Pos, @NextPos - @Pos)))
    SELECT @Pos = @NextPos+1

    INSERT INTO @tbNames VALUES (@Name)
    SET @Num = @Num + 1
END

INSERT INTO @tbUsers
SELECT UserId
FROM   dbo.aspnet_Users ar, @tbNames t

```



```
WHERE LOWER(t.Name) = ar.LoweredUserName AND ar.ApplicationId =  
@AppId
```

```
IF ( @@ROWCOUNT <> @Num)  
BEGIN  
    DELETE FROM @tbNames  
    WHERE LOWER(Name) IN (SELECT LoweredUserName FROM  
dbo.aspnet_Users au, @tbUsers u WHERE au.UserId = u.UserId)
```

```
INSERT dbo.aspnet_Users (ApplicationId, UserId, UserName,  
LoweredUserName, IsAnonymous, LastActivityDate)  
    SELECT @AppId, NEWID(), Name, LOWER(Name), 0,  
@CurrentTimeUtc  
    FROM @tbNames
```

```
INSERT INTO @tbUsers  
    SELECT UserId  
    FROM dbo.aspnet_Users au, @tbNames t  
    WHERE LOWER(t.Name) = au.LoweredUserName AND  
au.ApplicationId = @AppId  
END
```

```
IF (EXISTS (SELECT * FROM dbo.aspnet_UsersInRoles ur, @tbUsers tu,  
@tbRoles tr WHERE tu.UserId = ur.UserId AND tr.RoleId = ur.RoleId))  
BEGIN  
    SELECT TOP 1 UserName, RoleName  
    FROM dbo.aspnet_UsersInRoles ur, @tbUsers tu, @tbRoles tr,  
aspnet_Users u, aspnet_Roles r  
    WHERE u.UserId = tu.UserId AND r.RoleId = tr.RoleId  
AND tu.UserId = ur.UserId AND tr.RoleId = ur.RoleId
```

```
IF( @TranStarted = 1 )  
    ROLLBACK TRANSACTION  
RETURN(3)  
END
```

```
INSERT INTO dbo.aspnet_UsersInRoles (UserId, RoleId)  
SELECT UserId, RoleId  
FROM @tbUsers, @tbRoles
```

```
IF( @TranStarted = 1 )  
    COMMIT TRANSACTION  
RETURN(0)  
END
```

8.50 - FINDUSERSINROLES

```
ALTER PROCEDURE dbo.aspnet_UsersInRoles_FindUsersInRole
    @ApplicationName nvarchar(256),
    @RoleName        nvarchar(256),
    @UserNameToMatch nvarchar(256)
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN(1)
    DECLARE @RoleId uniqueidentifier
    SELECT @RoleId = NULL

    SELECT @RoleId = RoleId
    FROM    dbo.aspnet_Roles
    WHERE   LOWER(@RoleName) = LoweredRoleName AND ApplicationId =
@ApplicationId

    IF (@RoleId IS NULL)
        RETURN(1)

    SELECT u.UserName
    FROM    dbo.aspnet_Users u, dbo.aspnet_UsersInRoles ur
    WHERE   u.UserId = ur.UserId AND @RoleId = ur.RoleId AND u.ApplicationId =
@ApplicationId AND LoweredUserName LIKE LOWER(@UserNameToMatch)
    ORDER BY u.UserName
    RETURN(0)
END
```

8.51 - GETROLESFORUSERS

```
ALTER PROCEDURE dbo.aspnet_UsersInRoles_GetRolesForUser
    @ApplicationName nvarchar(256),
    @UserName        nvarchar(256)
AS
```

```

BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
    LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN(1)
    DECLARE @UserId uniqueidentifier
    SELECT @UserId = NULL

    SELECT @UserId = UserId
    FROM   dbo.aspnet_Users
    WHERE  LoweredUserName = LOWER(@UserName) AND ApplicationId =
    @ApplicationId

    IF (@UserId IS NULL)
        RETURN(1)

    SELECT r.RoleName
    FROM   dbo.aspnet_Roles r, dbo.aspnet_UsersInRoles ur
    WHERE  r.RoleId = ur.RoleId AND r.ApplicationId = @ApplicationId AND
    ur.UserId = @UserId
    ORDER BY r.RoleName
    RETURN (0)
END

```

8.52 - GETUSERSINROLES

```

ALTER PROCEDURE dbo.aspnet_UsersInRoles_GetUsersInRoles
    @ApplicationName nvarchar(256),
    @RoleName        nvarchar(256)
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
    LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN(1)
    DECLARE @RoleId uniqueidentifier
    SELECT @RoleId = NULL

    SELECT @RoleId = RoleId

```

```

FROM  dbo.aspnet_Roles
WHERE LOWER(@RoleName) = LoweredRoleName AND ApplicationId =
@ApplicationId

IF (@RoleId IS NULL)
    RETURN(1)

SELECT u.UserName
FROM  dbo.aspnet_Users u, dbo.aspnet_UsersInRoles ur
WHERE u.UserId = ur.UserId AND @RoleId = ur.RoleId AND u.ApplicationId =
@ApplicationId
ORDER BY u.UserName
RETURN(0)
END

```

8.53 - ISUSERINROLES

```

ALTER PROCEDURE dbo.aspnet_UsersInRoles_IsUserInRole
    @ApplicationName nvarchar(256),
    @UserName        nvarchar(256),
    @RoleName        nvarchar(256)
AS
BEGIN
    DECLARE @ApplicationId uniqueidentifier
    SELECT @ApplicationId = NULL
    SELECT @ApplicationId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@ApplicationId IS NULL)
        RETURN(2)
    DECLARE @UserId uniqueidentifier
    SELECT @UserId = NULL
    DECLARE @RoleId uniqueidentifier
    SELECT @RoleId = NULL

    SELECT @UserId = UserId
    FROM  dbo.aspnet_Users
    WHERE LoweredUserName = LOWER(@UserName) AND ApplicationId =
@ApplicationId

    IF (@UserId IS NULL)
        RETURN(2)

    SELECT @RoleId = RoleId

```



```

FROM   dbo.aspnet_Roles
WHERE  LoweredRoleName = LOWER(@RoleName) AND ApplicationId =
@ApplicationId

```

```

IF (@RoleId IS NULL)
    RETURN(3)

```

```

IF (EXISTS( SELECT * FROM dbo.aspnet_UsersInRoles WHERE UserId =
@UserId AND RoleId = @RoleId))
    RETURN(1)
ELSE
    RETURN(0)
END

```

8.54 - REMOVEUSERSFROMROLES

```

ALTER PROCEDURE dbo.aspnet_UsersInRoles_RemoveUsersFromRoles
    @ApplicationName nvarchar(256),
    @UserNames       nvarchar(4000),
    @RoleNames       nvarchar(4000)
AS
BEGIN
    DECLARE @AppId uniqueidentifier
    SELECT @AppId = NULL
    SELECT @AppId = ApplicationId FROM aspnet_Applications WHERE
LOWER(@ApplicationName) = LoweredApplicationName
    IF (@AppId IS NULL)
        RETURN(2)

```

```

    DECLARE @TranStarted bit
    SET @TranStarted = 0

```

```

    IF( @@TRANCOUNT = 0 )
    BEGIN
        BEGIN TRANSACTION
        SET @TranStarted = 1
    END

```

```

    DECLARE @tbNames table(Name nvarchar(256) NOT NULL PRIMARY
KEY)
    DECLARE @tbRoles table(RoleId uniqueidentifier NOT NULL PRIMARY
KEY)

```

```

DECLARE @tbUsers table(UserId uniqueidentifier NOT NULL PRIMARY
KEY)
DECLARE @Num      int
DECLARE @Pos      int
DECLARE @NextPos  int
DECLARE @Name     nvarchar(256)
DECLARE @CountAll int
DECLARE @CountU   int
DECLARE @CountR   int

SET @Num = 0
SET @Pos = 1
WHILE(@Pos <= LEN(@RoleNames))
BEGIN
    SELECT @NextPos = CHARINDEX(N',', @RoleNames, @Pos)
    IF (@NextPos = 0 OR @NextPos IS NULL)
        SELECT @NextPos = LEN(@RoleNames) + 1
    SELECT @Name = RTRIM(LTRIM(SUBSTRING(@RoleNames,
@Pos, @NextPos - @Pos)))
    SELECT @Pos = @NextPos+1

    INSERT INTO @tbNames VALUES (@Name)
    SET @Num = @Num + 1
END

INSERT INTO @tbRoles
SELECT RoleId
FROM   dbo.aspnet_Roles ar, @tbNames t
WHERE  LOWER(t.Name) = ar.LoweredRoleName AND ar.ApplicationId =
@AppId
SELECT @CountR = @@ROWCOUNT

IF (@CountR <> @Num)
BEGIN
    SELECT TOP 1 N'', Name
    FROM   @tbNames
    WHERE  LOWER(Name) NOT IN (SELECT ar.LoweredRoleName
FROM   dbo.aspnet_Roles ar, @tbRoles r WHERE r.RoleId = ar.RoleId)
    IF( @TranStarted = 1 )
        ROLLBACK TRANSACTION
    RETURN(2)
END

DELETE FROM @tbNames WHERE 1=1
SET @Num = 0
SET @Pos = 1

```

```

WHILE(@Pos <= LEN(@UserNames))
BEGIN
    SELECT @NextPos = CHARINDEX(N',', @UserNames, @Pos)
    IF (@NextPos = 0 OR @NextPos IS NULL)
        SELECT @NextPos = LEN(@UserNames) + 1
    SELECT @Name = RTRIM(LTRIM(SUBSTRING(@UserNames,
@Pos, @NextPos - @Pos)))
    SELECT @Pos = @NextPos+1

    INSERT INTO @tbNames VALUES (@Name)
    SET @Num = @Num + 1
END

INSERT INTO @tbUsers
SELECT UserId
FROM dbo.aspnet_Users ar, @tbNames t
WHERE LOWER(t.Name) = ar.LoweredUserName AND ar.ApplicationId =
@AppId

SELECT @CountU = @@ROWCOUNT
IF (@CountU <> @Num)
BEGIN
    SELECT TOP 1 Name, N"
    FROM @tbNames
    WHERE LOWER(Name) NOT IN (SELECT au.LoweredUserName
FROM dbo.aspnet_Users au, @tbUsers u WHERE u.UserId = au.UserId)

    IF( @TranStarted = 1 )
        ROLLBACK TRANSACTION
    RETURN(1)
END

SELECT @CountAll = COUNT(*)
FROM dbo.aspnet_UsersInRoles ur, @tbUsers u, @tbRoles r
WHERE ur.UserId = u.UserId AND ur.RoleId = r.RoleId

IF (@CountAll <> @CountU * @CountR)
BEGIN
    SELECT TOP 1 UserName, RoleName
    FROM @tbUsers tu, @tbRoles tr, dbo.aspnet_Users u,
dbo.aspnet_Roles r
    WHERE u.UserId = tu.UserId AND r.RoleId = tr.RoleId
    AND tu.UserId NOT IN (SELECT ur.UserId FROM
dbo.aspnet_UsersInRoles ur WHERE ur.RoleId = tr.RoleId) AND
tr.RoleId NOT IN (SELECT ur.RoleId FROM
dbo.aspnet_UsersInRoles ur WHERE ur.UserId = tu.UserId)
    IF( @TranStarted = 1 )
        ROLLBACK TRANSACTION
    RETURN(3)

```

```

END

DELETE FROM dbo.aspnet_UsersInRoles
WHERE UserId IN (SELECT UserId FROM @tbUsers)
AND RoleId IN (SELECT RoleId FROM @tbRoles)
IF( @TranStarted = 1 )
    COMMIT TRANSACTION
RETURN(0)
END

```

8.55 - LOGEVENT

```

ALTER PROCEDURE dbo.aspnet_WebEvent_LogEvent
    @EventId      char(32),
    @EventTimeUtc  datetime,
    @EventTime     datetime,
    @EventType     nvarchar(256),
    @EventSequence decimal(19,0),
    @EventOccurrence decimal(19,0),
    @EventCode     int,
    @EventDetailCode int,
    @Message       nvarchar(1024),
    @ApplicationPath nvarchar(256),
    @ApplicationVirtualPath nvarchar(256),
    @MachineName   nvarchar(256),
    @RequestUrl    nvarchar(1024),
    @ExceptionType nvarchar(256),
    @Details       ntext
AS
BEGIN
    INSERT
        dbo.aspnet_WebEvent_Events
    (
        EventId,
        EventTimeUtc,
        EventTime,
        EventType,
        EventSequence,
        EventOccurrence,
        EventCode,
        EventDetailCode,
        Message,
        ApplicationPath,
        ApplicationVirtualPath,

```



```

MachineName,
RequestUrl,
ExceptionType,
Details
)
VALUES
(
  @EventId,
  @EventTimeUtc,
  @EventTime,
  @EventType,
  @EventSequence,
  @EventOccurrence,
  @EventCode,
  @EventDetailCode,
  @Message,
  @ApplicationPath,
  @ApplicationVirtualPath,
  @MachineName,
  @RequestUrl,
  @ExceptionType,
  @Details
)
END

```

CHAPTER 9 : TEN TERMS

9.1 - You Look Smart

Even if this book were 3,000 pages long, it still wouldn't be long enough to cover everything there is to know about Visual Web Developer, HTML, CSS, ASP.NET, and programming. You'll often need to refer to online resources or books for more information.

Given that Visual Web Developer is a tool for software developers, you'll have to get accustomed to terminology that developers use. This chapter presents the top ten terms that everyone will assume you already know and understand.

9.2 - Web Application

Familiar computer terms can change when they venture onto the World Wide Web. For instance, you may already know application as a program such as a word processor or spreadsheet, but the term Web application basically means a Web site or Web page that contains more than just static, unchanging HTML and text that looks the same to all visitors. A Web application is more dynamic than that; the page's exact content is created just before the page is sent. Exactly what the visitor sees depends on who that visitor is, or the specific information the user requested.

9.3 - Developer

There are two main types of computer users in the world, those who use software, and those who create software. The people who create software are called developers. Everyone else is called a user.

It's a lot easier to be a user than a developer. As a user, you can often figure out how to do things just by guessing and hacking away until you get it right. Not so for a developer. You have to actually know how the technology works — as well as what you're doing with it — to create software. Guessing and hacking away at

random generally leads to nothing but endless hours of hair-pulling frustration.

9.4 - Data-Driven

A data-driven Web site consists of Web pages whose content depends upon (“is driven by”) the data stored in a database. For example, when you search the Internet with a search engine such as Google, you’re basically sending a query string to Google’s database. What you get back is a dynamic page created in response to your query; exactly what appears on it depends on what you searched for. The page’s specific content is “driven” (determined) by what’s in Google’s database that matches what you searched for.

9.5 - ASP.NET 2.0

ASP stands for Active Server Pages, and is Microsoft’s fundamental technology for building dynamic, data-driven Web sites. As with all technologies, ASP has evolved over the years to become ever more powerful and easy to use.

As I write this book, ASP.NET 2.0 is the latest version of ASP. It’s also the version of ASP used by Visual Web Developer to create dynamic data-driven Web sites. To publish a site you created with Visual Web Developer, you must find a hosting provider that supports ASP.NET 2.0.

9.6 - Visual Studio

Visual Studio is Microsoft’s main software-development tool. It has all kinds of functionality for creating all kinds of programs. It’s big, complex, and easy to get lost in because Visual Studio provides a seemingly-endless array of tools and options.

Even so, Visual Web Developer is (believe it or not) a greatly uncomplicated version of Visual Studio; it focuses on developing one type of application: dynamic, data-driven Web sites. In effect, Visual Web Developer just “hides”

the parts of Visual Studio that aren't relevant to developing Web applications.

9.7 - IDE

IDE acronym stands for Integrated Development Environment. Both Visual Studio and Visual Web Developer are IDEs. They're called integrated because they give you access to all the various tools and technologies you need to get the job done.

To create data-driven Web sites, for example, you need a pretty specific list of things: HTML, CSS, ASP.NET, a data source (such as SQL Server), and at least one programming language (such as C#). An IDE like Visual Web developer brings all those technologies together under one roof, so to speak, so you can get all your work done in one program window. (What a concept.)

You've seen a similar (but unrelated) term before: In hard drives, IDE stands for Integrated Device Electronics. But that concept is in no way related to software development. In hard drives, it just means that all the electronics the disk needs to operate are contained within the drive's case.

9.8 - Control

When used as a noun within the context of programming, the term control refers to things in a program, or on a page, that allow users to control the action. For example, open any Windows dialog box and you see a bunch of text boxes, drop-down menus, check boxes, and buttons. All of these are relatively small, simple controls.

In a more general sense, a control can be anything on the page that a user interacts with. In Visual Web Developer's Toolbox, each item actually represents a control that you can drag and drop onto a page. As a developer, you

choose which controls users will see, and can program what happens when a user interacts with a control.

9.9 - Code

Code is the text that forms the instructions for the computer, written in some programming language such as C# or Visual Basic. It has a different function from text that's typed in a human language. Unlike regular text, code uses no sentences or paragraphs. Instead, code consists of statements (lines) written in the programming language rather than in a human language.

When you write in (say) English, you can get away with murder in terms of spelling and punctuation; even if the text would make your composition teacher cry, your human recipient will probably still be able to figure out what you mean — simply by studying the context of the message. (If you've spent any time in online forums, you've no doubt seen the extent to which people can get away with worse-than-iffy spelling and punctuation.)

When you write code, there is no such margin for error. Every character of every word counts; if it isn't just so, you'll likely get an error message rather than code that works. Unlike a human being who can figure out (at least) the intent of some dreadfully composed written message, a computer can't "figure out" diddley squat. You either write the code correctly and it works, or you write it incorrectly and it doesn't work.

9.10 - Programmatic

There are two ways to interact with controls. One way is manual: You actually use the finished control yourself. For example, when you click a button in a Windows dialog box, you are manually operating the control with your mouse and keyboard.

As a programmer, you can also write code to specify the exact appearance

and behavior of a control, as well as what happens after a user interacts with the control. That sort of code is generally executed in response to some event, as when the user clicks a button or makes a selection from a drop-down menu.

In essence, a user who is viewing a page and makes a selection from the options provided is manually using the control. But exactly what happens after the user makes a selection is generally handled programmatically — the control runs some code you've written, and that code tells the machine exactly what to do.

9.11 - Database

The term database is often used casually to refer to any body of knowledge or collection of information. In programming terms, it's more specific — referring to data that's organized into tables that consist of columns and rows so it can be stored and accessed consistently. The database for a Web site is made available through the Database Explorer tab in Visual Web Developer, as shown in Figure 13-2.

A database management system (DBMS) is a program that allows you to organize data into tables, and retrieve data programmatically. Microsoft Access and SQL Server are examples of database-management systems. Microsoft Access is a self-contained DBMS; it contains all the tools you need to create tables, queries, forms, reports, and other aspects of an application — in one program.

Server isn't so much an application as it is a server — a single-minded program obsessed with serving up data. Unlike an application program, which has its own program window, a server works behind the scenes; it has no particular interface or program window of its own. Instead, the server just

“serves up” data to some external application — such as (for a convenient example) a dynamic data-driven Web site you create in Visual Web Developer

CHAPTER 10 : TEN ALTERNATIVES TO BEING HELPLESS

Visual Web Developer brings together all the technologies you need to create dynamic, data-driven Web sites. Those technologies include HTML, CSS, ASP.NET, C#, the .NET Framework, and SQL Server — each and every one of which is an enormous topic in itself. (For example, the first printed documentation for the .NET Framework alone consisted of over 8,000 pages, bound into several volumes.)

To deal with such a gigantic body of knowledge, you must find ways to get exactly the information you need, when you need it. Sending desperate “PLEASE HELP!!!!” messages generally won’t cut it. You have to be more resourceful — and know where the people you’re asking are getting their information, so you can go get that information too.

The sites listed in this chapter get you in the ballpark. Of course, Web sites being what they are, some of the URLs listed here may change between now and the time you actually read this. That’s why I keep an up-to-date list of links on www.coolnerds.com/vwd just in case any change occurs. If you have any problems finding a page, try looking there.

10.1 - Microsoft Developer Network (MSDN)

The Microsoft Developer Network at <http://msdn.microsoft.com> is home to all Microsoft software-development tools and technologies. This is where the hardcore computer geeks go to get information about Microsoft developer products.

10.2 - HTML Home Page

All Web pages are formatted using HyperText Markup Language (HTML). The W3C (Word Wide Web Consortium) is the “official” site for all things HTML (including that useful new mutant, XHTML). To go straight to documentation on HTML or XHTML, use the link www.w3.org/MarkUp/.

10.3 - Cascading Style Sheets Home Page

Cascading Style Sheets (CSS) are a must if you intend to give your Web site a consistent look and feel. They’re even more indispensable if you intend to use themes, because every theme includes some CSS. The home page for CSS is at www.w3.org/Style/CSS/.

XML Home Page

XML (eXtensible Markup Language) is a standardized means of transmitting raw data across the Internet via the Word Wide Web. (By “raw” I mean unformatted, like the data in a database — useful even if it isn’t pretty to look at.)

Whether your site needs (or can even use) XML depends on many factors.

In Visual Web Developer, you may be able to get by with little or no XML at all — that depends on your site. If you do need XML info, the official site at www.w3.org/XML/ contains all the specs.

ASP.NET

If I was forced to pin down my choice for the main technology behind Visual Web Developer, ASP.NET 2.0 would be the one I’d choose. Although there are plenty of Web sites that address this topic, the most “official” would have to be the one whose URL is also the easiest in the world to remember:

<http://ASP.NET>.

ASP.NET Starter Kits

Clicking the Starter Kits tab at the ASP.NET Web site takes you to a page of

sample Web applications that you can download for free. There are different kinds of Starter Kits for different types of Web sites. If you have a particular type of site in mind, but don't quite know how to organize things, a Starter Kit could be a great way to get started.

10.4 - ASP.NET Forums

The ASP.NET Forums at <http://forums.asp.net> give you a handy place to go for questions and answers. There are a lot of discussions to choose from. You may have to scroll down to find Visual Web Developer and related discussions. You can discover a lot about what the technology can do just by reading through some of the questions and answers that have already been posted.

10.5 - SQL Server Developer Center

The subtitle to this Web page, "Enabling a world of data-driven applications," is what SQL Server is all about. In a Visual Web Developer site, SQL Server contains all the data that drives the Web site. As I write this, the URL for SQL Server 2005 is <http://msdn.microsoft.com/SQL/2005>. If that doesn't work, try the more general URL <http://msdn.microsoft.com/SQL>.

10.6 - Dotnetjunkies

The Microsoft .NET Framework is a huge collection of pre-written software that most modern programmers use to create modern programs. In fact, all the Using statements you see at the top of a code-behind page in VWD are references to namespaces within the .NET Framework. Each namespace is like a folder containing usable code organized into classes.

The dotnetjunkies site (at www.dotnetjunkies.com) is a place for people who are into the whole .NET way of doing things. Though it spans the whole

of .NET Framework development tools, there's always info on ASP.NET 2.0 and Visual Web Developer.

10.7 - Microsoft Technical Communities

This site encompasses all of Microsoft's software-development tools — and discusses them in blogs, technical chats, newsgroups, webcasts, communities, user groups, and forums. There are all sorts of ways to interact with other people who use the same tools you're using. The URL for this site is www.microsoft.com/communities.

CONCLUSION

We are in the 21st century and the time is short, business is expanding jobs are much likely to be done with the least time and power resources. The technology out of all is the web which is burning the mid night out in this fast world.

To sum up I have been using such a fast and brilliant technology of web and internet facilities to get the best out of it.

The most vital factor considered while building this project was the dynamic Driven, simply not just being awesome but being proficient in the task.

Security is one the basic and very much to be faced out in the whole cyber world every single day. To get sorted out in a fabulous way I used GUI the high level security identification system.

REFERENCES

[1] Alan Simpson , Visual Web Developer 2005 Express Edition

Publishing : WROX 2005

[2] David Sussman Alex Hommer , ASP.NET Visual Web Developer 2005 Express Edition STARTER KIT

Publishing WROX 2005

[3] A guide for writing asp codes retrived from the world wide web <http://www.asp.net>
ASP FORUMS

[4]Tutorials for writing asp and c# retrieved from world wide web
<http://www.w3schools.com/asp/default.asp> TUTORIAL OF ASP

[5] Tutotials and code examples retrieved from world wide web
<http://www.aspfree.com> ALL ABOUT ASP