<script type = "text/javascript" >

function preventBack() { window.history.forward(); }

setTimeout("preventBack()", 0);

window.onunload = function () { null };

</script>

Application["UsersObj"] = users;

Session.Clear();

Session.Abandon();

Session.RemoveAll();

HttpContext.Current.Application.Clear();

if (Request.Cookies["ASP.NET\_SessionId"] != null)

{

Response.Cookies["ASP.NET\_SessionId"].Value = string.Empty;

Response.Cookies["ASP.NET\_SessionId"].Value = "";

Response.Cookies["ASP.NET\_SessionId"].Expires = DateTime.Now.AddYears(-1);

Response.Cookies.Add(new HttpCookie("ASP.NET\_SessionId", ""));

}

if (Request.Cookies["AuthToken"] != null)

{

Response.Cookies["AuthToken"].Value = string.Empty;

Response.Cookies["AuthToken"].Value = "";

Response.Cookies["AuthToken"].Expires = DateTime.Now.AddYears(-1);

Response.Cookies.Add(new HttpCookie("AuthToken", ""));

}

HttpCookie CheckLogin = new HttpCookie("UserInfo");

CheckLogin["UserType"] = null;

CheckLogin.Expires = DateTime.Now.AddDays(-2);

Response.Cookies.Add(CheckLogin);

Response.Redirect("~/Mainpage.aspx", false);

Masterafterlogin

#region For Session Fixation

if (Session["AuthToken"] != null && Request.Cookies["AuthToken"] != null && Application["UsersObj"] != null)

{

if (Session["AuthToken"].ToString().Equals(Request.Cookies["AuthToken"].Value))

{

// createa a new GUID and save into the session

string guid = Guid.NewGuid().ToString();

Session["AuthToken"] = guid;

// now create a new cookie with this guid value

Response.Cookies.Add(new HttpCookie("AuthToken", guid));

}

else

{

Session.Abandon();

Session.RemoveAll();

Response.Redirect("~/MainPage.aspx");

}

}

else

{

Session.Abandon();

Session.RemoveAll();

Response.Redirect("~/MainPage.aspx");

}

#endregion

Response.Cache.SetNoStore();

Response.AppendHeader("Pragma", "no-cache");

Response.Expires = 0;

<meta charset="utf-8" />

<meta http-equiv="X-Frame-Options" content="deny" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

if (Session["UsersObj"] != null && Session["UserTypeId"] != null)

{

if (Session["UserTypeId"].ToString() != "ad")

{

Response.Redirect("~/Mainpage.aspx");

}

else

{

if (Page.IsPostBack == false)

{

users = (Users)Session["UsersObj"];

loadStates();

BindGridView();

}

}

}

else

{

Response.Redirect("~/Mainpage.aspx");

}

if (Page.IsPostBack == false)

{

string CSRFToken = CheckSecurity.GenerateRandomCode();

Session["CSRFToken"] = CSRFToken;

ViewState["HdnCSRF"] = CSRFToken;

}

else

{

if (Session["CSRFToken"].ToString() == Convert.ToString(ViewState["HdnCSRF"]))

{

string CSRFToken = CheckSecurity.GenerateRandomCode();

Session["CSRFToken"] = CSRFToken;

ViewState["HdnCSRF"] = CSRFToken;

}

else

{

Response.Redirect("~/MainPage.aspx");

return;

}

}

#endregion

users = (Users)Session["UsersObj"];

**New Password**

<asp:RegularExpressionValidator ID="rfex\_password" ForeColor="Red" runat="server" ErrorMessage="Ex-\*(aA@1\*\*\*\*)min 8 Char "

ValidationExpression="^.\*(?=.{8,})(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?=.\*[@#$%^&+=]).\*$"

ControlToValidate="txtPwd"></asp:RegularExpressionValidator>

Password change submit button

Session.Abandon();

Session.Clear();

Response.Cookies.Clear();

HttpContext.Current.Application.Clear();

Session.Clear();

Session.Abandon();

Session.RemoveAll();

HttpContext.Current.Application.Clear();

if (Request.Cookies["ASP.NET\_SessionId"] != null)

{

Response.Cookies["ASP.NET\_SessionId"].Value = string.Empty;

Response.Cookies["ASP.NET\_SessionId"].Value = "";

Response.Cookies["ASP.NET\_SessionId"].Expires = DateTime.Now.AddYears(-1);

Response.Cookies.Add(new HttpCookie("ASP.NET\_SessionId", ""));

}

if (Request.Cookies["AuthToken"] != null)

{

Response.Cookies["AuthToken"].Value = string.Empty;

Response.Cookies["AuthToken"].Value = "";

Response.Cookies["AuthToken"].Expires = DateTime.Now.AddYears(-1);

Response.Cookies.Add(new HttpCookie("AuthToken", ""));

}

HttpCookie CheckLogin = new HttpCookie("UserInfo");

CheckLogin["UserType"] = null;

CheckLogin.Expires = DateTime.Now.AddDays(-2);

Response.Cookies.Add(CheckLogin);

Response.Redirect("~/Mainpage.aspx", false);