public void SignOut()

{

try { FormsAuthentication.SignOut(); }

catch { }

try

{

Response.Cookies.Remove(FormsAuthentication.FormsCookieName);

Response.Cookies[FormsAuthentication.FormsCookieName].Expires = DateTime.Now;

Response.Cache.SetExpires(DateTime.Now.AddSeconds(-1));

HttpCookie cookie = Request.Cookies[FormsAuthentication.FormsCookieName];

Session.RemoveAll(); Session.Abandon();

if (cookie != null)

{

cookie.Expires = DateTime.Now.AddDays(-1);

Response.Cookies.Add(cookie);

}

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

//Prevents to prevoius authenticated after LogOut

Response.Cache.SetCacheability(HttpCacheability.NoCache);

Response.Cache.SetExpires(DateTime.Now.AddSeconds(-1));

Response.Cache.SetNoStore();

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

}

catch { }

}

<compilation debug="false" targetFramework="4.0">

protected void Application\_PreSendRequestHeaders(object sender, EventArgs e)

{

Response.AddHeader("X-Frame-Options", "SAMEORIGIN");

Response.Headers.Remove("X-Powered-By");

Response.Headers.Remove("Server");

Response.Headers.Remove("X-AspNet-Version");

Response.Headers.Remove("X-AspNetMvc-Version");

}

public static bool IsPasswordStrong(string password)

{

int TotVale = 0;

bool IsTrue = false;

int lowers = password.Count(p => char.IsLower(p));

int uppers = password.Count(p => char.IsUpper(p));

int digits = password.Count(p => char.IsDigit(p));

int specials = password.Count(p => !char.IsLetterOrDigit(p));

TotVale = ((lowers >= 3 ? 1 : 0) +

(uppers >= 1 ? 1 : 0) +

(digits >= 1 ? 1 : 0) +

(specials >= 1 ? 1 : 0));

if (TotVale > 3)

{ IsTrue = true; }

return IsTrue;

}

autocomplete="off"