













using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

public partial class NewUser : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=G:\New Project 2016\Holy\_Cross\ECCWebsite\App\_Data\eccdb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

cmd = new SqlCommand("insert into regtb values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + RadioButtonList1.SelectedItem.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "','" + TextBox7.Text + "','" + TextBox8.Text + "')", con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Write("<Script> alert('New User Info saved!') </Script>");

}

protected void Button2\_Click(object sender, EventArgs e)

{

TextBox1.Text = "";

TextBox2.Text = "";

TextBox3.Text = "";

TextBox4.Text = "";

TextBox5.Text = "";

TextBox6.Text = "";

TextBox7.Text = "";

TextBox8.Text = "";

TextBox9.Text = "";

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

public partial class UserLogin : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=G:\New Project 2016\Holy\_Cross\ECCWebsite\App\_Data\eccdb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

con.Open();

cmd = new SqlCommand("select \* from regtb where UserName='" + TextBox1.Text + "' and Password='" + TextBox2.Text + "'", con);

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

Session["uname"] = TextBox1.Text;

Response.Redirect("UserHome.aspx");

}

else

{

Response.Write("<Script> alert('Password incorrect!') </Script>");

}

con.Close();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

public partial class AdminHome : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=G:\New Project 2016\Holy\_Cross\ECCWebsite\App\_Data\eccdb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

protected void Page\_Load(object sender, EventArgs e)

{

cmd = new SqlCommand("select \* from regtb ", con);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

GridView1.DataSource = dt;

GridView1.DataBind();

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.IO;

using System.Security.Cryptography;

using System.Text;

public partial class Adminfileupload : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=G:\New Project 2016\Holy\_Cross\ECCWebsite\App\_Data\eccdb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public static byte[] alicePublicKey;

protected void Page\_Load(object sender, EventArgs e)

{

}

string filename;

protected void Button1\_Click(object sender, EventArgs e)

{

using (ECDiffieHellmanCng alice = new ECDiffieHellmanCng())

{

filename = System.IO.Path.GetFileName(FileUpload1.PostedFile.FileName);

FileUpload1.PostedFile.SaveAs(Server.MapPath("~/Upload/" + filename));

FileUpload1.PostedFile.SaveAs(Server.MapPath("~/split/" + filename));

string path = @"G:\New Project 2016\Holy\_Cross\ECCWebsite\Encrypt1\" + filename;

string path1 = @"G:\New Project 2016\Holy\_Cross\ECCWebsite\split\" + filename;

//string path5 = @"D:\New Project 2016\MNSK\Cloud Audit Control system\Cloud Owner\File\" + Label4.Text + ".txt";

string inputContent;

using (StreamReader inputStreamReader = new StreamReader(FileUpload1.PostedFile.InputStream))

{

inputContent = inputStreamReader.ReadToEnd();

}

alice.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;

alice.HashAlgorithm = CngAlgorithm.Sha256;

// ASCIIEncoding.ASCII.GetBytes(TextBox2.Text, 0, TextBox2.Text.Length, alicePublicKey, 3);

// alicePublicKey = byte.Parse(TextBox2.Text, NumberStyles.AllowHexSpecifier);

alicePublicKey = System.Text.Encoding.Default.GetBytes(TextBox1.Text);

alicePublicKey = alice.PublicKey.ToByteArray();

Bob bob = new Bob();

CngKey k = CngKey.Import(bob.bobPublicKey, CngKeyBlobFormat.EccPublicBlob);

byte[] aliceKey = alice.DeriveKeyMaterial(CngKey.Import(bob.bobPublicKey, CngKeyBlobFormat.EccPublicBlob));

byte[] encryptedMessage = null;

byte[] iv = null;

Send(aliceKey, inputContent, out encryptedMessage, out iv);

bob.Receive(encryptedMessage, iv);

//TextBox1.Text = encryptedMessage.ToString();

string result = System.Text.Encoding.UTF8.GetString(encryptedMessage);

TextBox1.Text = result;

ByteArrayToFile(path, encryptedMessage);

ByteArrayToFile(path1, encryptedMessage);

SplitFile(Server.MapPath(@"~/split/" + filename), Convert.ToInt32(2));

}

}

public bool ByteArrayToFile(string \_FileName, byte[] \_ByteArray)

{

try

{

// Open file for reading

System.IO.FileStream \_FileStream =

new System.IO.FileStream(\_FileName, System.IO.FileMode.Create,

System.IO.FileAccess.Write);

// Writes a block of bytes to this stream using data from

// a byte array.

\_FileStream.Write(\_ByteArray, 0, \_ByteArray.Length);

// close file stream

\_FileStream.Close();

return true;

}

catch (Exception \_Exception)

{

// Error

Console.WriteLine("Exception caught in process: {0}",

\_Exception.ToString());

}

// error occured, return false

return false;

}

protected void Button2\_Click(object sender, EventArgs e)

{

TextBox1.Text = "";

TextBox2.Text = "";

TextBox3.Text = "";

TextBox4.Text = "";

}

private static void Send(byte[] key, string secretMessage, out byte[] encryptedMessage, out byte[] iv)

{

using (Aes aes = new AesCryptoServiceProvider())

{

aes.Key = key;

iv = aes.IV;

// Encrypt the message

using (MemoryStream ciphertext = new MemoryStream())

using (CryptoStream cs = new CryptoStream(ciphertext, aes.CreateEncryptor(), CryptoStreamMode.Write))

{

byte[] plaintextMessage = Encoding.UTF8.GetBytes(secretMessage);

cs.Write(plaintextMessage, 0, plaintextMessage.Length);

cs.Close();

encryptedMessage = ciphertext.ToArray();

}

}

}

public FileStream fs;

string mergeFolder;

List<string> Packets = new List<string>();

public bool SplitFile(string SourceFile, int nNoofFiles)

{

bool Split = false;

try

{

FileStream fs = new FileStream(SourceFile, FileMode.Open, FileAccess.Read);

int SizeofEachFile = (int)Math.Ceiling((double)fs.Length / nNoofFiles);

for (int i = 0; i < nNoofFiles; i++)

{

string baseFileName = Path.GetFileNameWithoutExtension(SourceFile);

string Extension = Path.GetExtension(SourceFile);

FileStream outputFile = new FileStream(Path.GetDirectoryName(SourceFile) + "\\" + baseFileName + "." +

i.ToString().PadLeft(3, Convert.ToChar("0")) + Extension, FileMode.Create, FileAccess.Write);

//FileStream outputFile = new FileStream(Path.GetDirectoryName(SourceFile) + "\\" + baseFileName + "." +

// i.ToString().PadLeft(5, Convert.ToChar("0")) + Extension + ".txt", FileMode.Create, FileAccess.Write);

mergeFolder = Path.GetDirectoryName(SourceFile);

int bytesRead = 0;

byte[] buffer = new byte[SizeofEachFile];

if ((bytesRead = fs.Read(buffer, 0, SizeofEachFile)) > 0)

{

outputFile.Write(buffer, 0, bytesRead);

//outp.Write(buffer, 0, BytesRead);

string packet = baseFileName + "." + i.ToString().PadLeft(3, Convert.ToChar("0")) + Extension.ToString();

Packets.Add(packet);

}

outputFile.Close();

}

fs.Close();

}

catch (Exception Ex)

{

throw new ArgumentException(Ex.Message);

}

return Split;

}

}

public class Bob

{

public byte[] bobPublicKey;

private byte[] bobKey;

public Bob()

{

using (ECDiffieHellmanCng bob = new ECDiffieHellmanCng())

{

bob.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;

bob.HashAlgorithm = CngAlgorithm.Sha256;

bobPublicKey = bob.PublicKey.ToByteArray();

bobKey = bob.DeriveKeyMaterial(CngKey.Import(Adminfileupload.alicePublicKey, CngKeyBlobFormat.EccPublicBlob));

}

}

public void Receive(byte[] encryptedMessage, byte[] iv)

{

using (Aes aes = new AesCryptoServiceProvider())

{

aes.Key = bobKey;

aes.IV = iv;

// Decrypt the message

using (MemoryStream plaintext = new MemoryStream())

{

using (CryptoStream cs = new CryptoStream(plaintext, aes.CreateDecryptor(), CryptoStreamMode.Write))

{

cs.Write(encryptedMessage, 0, encryptedMessage.Length);

cs.Close();

string message = Encoding.UTF8.GetString(plaintext.ToArray());

Console.WriteLine(message);

}

}

}

}

}