**Sample code:**

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.IO;

using System.Text;

using System.Security.Cryptography;

using System.Data.SqlClient;

using System.Threading;

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

{

SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["PODcon"].ConnectionString);

public FileStream fs;

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

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

{

Label35.Text = Session["uname"].ToString();

if (!IsPostBack)

{

Label17.Text = Request.QueryString["fid"].ToString();

TextBox1.Text = Request.QueryString["fsub"].ToString();

TextBox2.Text = Request.QueryString["fname"].ToString();

string fname = Server.MapPath("~/Upload/") + TextBox2.Text;

string fnamwoex = Path.GetFileNameWithoutExtension(fname);

string ext = Path.GetExtension(fname);

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

int FileLength = (int)fs.Length / 1024;

fs.Flush();

fs.Dispose();

fs.Close();

string name = Path.GetFileName(fname);

SplitFile(fname, Convert.ToInt32(3));

ListBox1.Items.Add(Packets[0].ToString());

ListBox1.Items.Add(Packets[1].ToString());

ListBox1.Items.Add(Packets[2].ToString());

TextBox3.Text = File.ReadAllText(Server.MapPath("~/Upload/Split/") + fnamwoex + ".0" + ext);

TextBox4.Text = File.ReadAllText(Server.MapPath("~/Upload/Split/") + fnamwoex + ".1" + ext);

TextBox5.Text = File.ReadAllText(Server.MapPath("~/Upload/Split/") + fnamwoex + ".2" + ext);

TextBox6.Text = hashvalue(TextBox3.Text);

TextBox7.Text = hashvalue(TextBox4.Text);

TextBox8.Text = hashvalue(TextBox5.Text);

}

}

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);

string splitfname = Server.MapPath("~/Upload/Split/") + Label18.Text;

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

i.ToString().PadLeft(1, Convert.ToChar("0")) + Extension, 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);

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

Packets.Add(packet);

}

outputFile.Flush();

outputFile.Close();

outputFile.Close();

}

fs.Flush();

fs.Dispose();

fs.Close();

}

catch (Exception Ex)

{

throw new ArgumentException(Ex.Message);

}

return Split;

}

public string hashvalue(string textval)

{

byte[] srcdata = ASCIIEncoding.ASCII.GetBytes(textval);

byte[] hashdata = new MD5CryptoServiceProvider().ComputeHash(srcdata);

return ByteArrayToString(hashdata);

}

string dname;

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

{

Random rm = new Random();

int rno = rm.Next(1, 4);

if (rno == 1)

{

dname = "Disk0";

}

else if (rno == 2)

{

dname = "Disk1";

}

else if (rno >= 3)

{

dname = "Disk2";

}

string fname = Server.MapPath("~/Upload/") + TextBox2.Text;

FileStream fls = new FileStream(fname, FileMode.Open, FileAccess.ReadWrite);

byte[] buffer = new byte[fls.Length];

fls.Read(buffer, 0, (int)fls.Length);

long fsize = fls.Length;

fls.Flush();

fls.Dispose();

fls.Close();

con.Open();

SqlCommand cmd0 = new SqlCommand("select Filename from HDD\_tbl where Filename = '"+TextBox2.Text+"' AND Hash1 = '"+TextBox6.Text+"' AND Hash2 = '"+TextBox7.Text+"' AND Hash3 = '"+TextBox8.Text+"'",con);

string filnam = (string)cmd0.ExecuteScalar();

con.Close();

if (filnam != null)

{

Response.Write("<script>alert('Checking Duplication')</script>");

Thread.Sleep(4000);

Response.Write("<script>alert('Data Duplication Found')</script>");

Label37.Visible = true;

Label36.Visible = true;

Label36.Text = "Select-Dedupe deduplicate the redundant data Blocks";

con.Open();

SqlCommand cmdup1 = new SqlCommand("update HDD\_tbl set Chunk1 = '"+TextBox3.Text+"',Hash1 = '"+TextBox6.Text+"',Chunk2 = '"+TextBox4.Text+"',Hash2 = '"+TextBox7.Text+"',Chunk3 = '"+TextBox5.Text+"',Hash3 = '"+TextBox8.Text+"' where Filename = '"+TextBox2.Text+"'",con);

cmdup1.ExecuteNonQuery();

con.Close();

}

else

{

con.Open();

SqlCommand cmd = new SqlCommand("insert into Upload values('" + Label17.Text + "','" + TextBox1.Text + "','" + TextBox2.Text + "',@Filedata)", con);

cmd.Parameters.AddWithValue("@Filedata", buffer);

cmd.ExecuteNonQuery();

SqlCommand cmd1 = new SqlCommand("insert into HDD\_tbl values('" + Label17.Text + "','" + Label35.Text

+ "','" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox6.Text

+ "','" + TextBox4.Text + "','" + TextBox7.Text + "','" + TextBox5.Text + "','" + TextBox8.Text

+ "','" + dname + "','NO','" + fsize + "')", con);

cmd1.ExecuteNonQuery();

con.Close();

Response.Write("<script>alert('File Uploaded to Cloud Server!')</script>");

}

}

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

{

Response.Redirect("UserUpload.aspx");

}

static string ByteArrayToString(byte[] arrInput)

{

int i;

StringBuilder sOutput = new StringBuilder(arrInput.Length);

for (i = 0; i < arrInput.Length; i++)

{

sOutput.Append(arrInput[i].ToString("X2"));

}

return sOutput.ToString();

}

}