using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace AuthenticationSystem

{

public partial class AdminHome : Form

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=|Datadirectory|\authedb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public AdminHome()

{

InitializeComponent();

}

private void AdminHome\_Load(object sender, EventArgs e)

{

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

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.Refresh();

}

private void userDetailsToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form1 ff = new Form1();

ff.Show();

}

private void statementinfoToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

private void logoutToolStripMenuItem\_Click(object sender, EventArgs e)

{

Home hh = new Home();

hh.Show();

this.Close();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace AuthenticationSystem

{

public partial class AdminLogin : Form

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=|Datadirectory|\authedb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public AdminLogin()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox2.Text = "";

int[] array = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };

array = ShuffledArray(array);

foreach (int s in array)

{

Console.WriteLine(s);

}

}

protected int[] ShuffledArray(int[] myArray)

{

int count = myArray.Length - 1;

int[] newArray = new int[count + 1];

Random rnd = new Random();

var randomNumbers = Enumerable.Range(1, count).OrderBy(i => rnd.Next()).ToArray();

int index = 0;

foreach (int i in randomNumbers)

{

newArray[index] = myArray[i];

index++;

}

return newArray;

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "admin" & textBox2.Text == "admin")

{

AdminHome safds = new AdminHome();

safds.Show();

}

else

{

MessageBox.Show("Username or Password Incorrect!");

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using Luxand;

using System.Data.SqlClient;

using System.IO;

namespace AuthenticationSystem

{

public partial class Form1 : Form

{

public string uname;

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=|Datadirectory|\authedb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

// program states: whether we recognize faces, or user has clicked a face

enum ProgramState { psNormal, psRemember, psRecognize }

ProgramState programState = ProgramState.psRecognize;

struct FaceTemplate

{ // single template

public byte[] templateData;

}

List<FaceTemplate> faceTemplates;

string path = Path.GetDirectoryName(Application.ExecutablePath).ToString();

String cameraName;

bool needClose = false;

string userName;

int mouseX = 0;

int mouseY = 0;

// WinAPI procedure to release HBITMAP handles returned by FSDKCam.GrabFrame

[DllImport("gdi32.dll")]

static extern bool DeleteObject(IntPtr hObject);

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

if (FSDK.FSDKE\_OK != FSDK.ActivateLibrary("gyYgVWQTSzjiuGB/hH8dKgg0QrrIuhoHdfUCzD9rY+vru3WRZsaezTX6YWj9osdI/cmxY1NSdLkyWuugMPCxUG7/xNLegHLeaUpzVyKpDkaWL8tJIUsIL7xv9bhmgifPbAyTDuxF3VGxXmHkv/L/MStf9kdXV/A1vVvT93QC4vQ="))

{

MessageBox.Show("Please run the License Key Wizard (Start - Luxand - FaceSDK - License Key Wizard)", "Error activating FaceSDK", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

FSDK.InitializeLibrary();

FSDKCam.InitializeCapturing();

string[] cameraList;

int count;

FSDKCam.GetCameraList(out cameraList, out count);

if (0 == count)

{

MessageBox.Show("Please attach a camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

cameraName = cameraList[0];

FSDKCam.VideoFormatInfo[] formatList;

FSDKCam.GetVideoFormatList(ref cameraName, out formatList, out count);

//int VideoFormat = 0; // choose a video format

//pictureBox1.Width = formatList[VideoFormat].Width;

//pictureBox1.Height = formatList[VideoFormat].Height;

//this.Width = formatList[VideoFormat].Width + 48;

//this.Height = formatList[VideoFormat].Height + 96;

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

// needClose = true;

}

private void button1\_Click(object sender, EventArgs e)

{

this.button1.Enabled = false;

int cameraHandle = 0;

int r = FSDKCam.OpenVideoCamera(ref cameraName, ref cameraHandle);

if (r != FSDK.FSDKE\_OK)

{

MessageBox.Show("Error opening the first camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

btnRemember.Enabled = true;

// set realtime face detection parameters

FSDK.SetFaceDetectionParameters(false, false, 100);

FSDK.SetFaceDetectionThreshold(3);

// list where we store face templates

// faceTemplates = new List();

faceTemplates = new List<FaceTemplate>();

while (!needClose)

{

Int32 imageHandle = 0;

if (FSDK.FSDKE\_OK != FSDKCam.GrabFrame(cameraHandle, ref imageHandle)) // grab the current frame from the camera

{

Application.DoEvents();

continue;

}

FSDK.CImage image = new FSDK.CImage(imageHandle);

Image frameImage = image.ToCLRImage();

Graphics gr = Graphics.FromImage(frameImage);

FSDK.TFacePosition facePosition = image.DetectFace();

// if a face is detected, we can recognize it

if (facePosition.w != 0)

{

gr.DrawRectangle(Pens.LightGreen, facePosition.xc - facePosition.w / 2, facePosition.yc - facePosition.w / 2,

facePosition.w, facePosition.w);

// create a new face template

FaceTemplate template = new FaceTemplate();

template.templateData = new byte[FSDK.TemplateSize];

FaceTemplate template1 = new FaceTemplate();

if (programState == ProgramState.psRemember || programState == ProgramState.psRecognize)

template.templateData = image.GetFaceTemplateInRegion(ref facePosition);

switch (programState)

{

case ProgramState.psNormal: // normal state - do nothing

break;

case ProgramState.psRemember: // Remember Me state - store facial templates

label1.Text = "Templates stored: " + faceTemplates.Count.ToString();

faceTemplates.Add(template);

if (faceTemplates.Count > 9)

{

// get the user name

InputName inputName = new InputName();

inputName.ShowDialog();

userName = inputName.userName;

try

{

cmd = new SqlCommand("insert into facetb values(@Name,@face)", con);

cmd.Parameters.AddWithValue("@Name", userName);

cmd.Parameters.AddWithValue("@face", template.templateData);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Save!");

programState = ProgramState.psRecognize;

}

catch (Exception ex)

{

}

}

break;

case ProgramState.psRecognize: // recognize the user

bool match = false;

int ii = 0;

foreach (FaceTemplate t in faceTemplates)

{

float similarity = 0.0f;

FaceTemplate t1 = t;

FSDK.MatchFaces(ref template.templateData, ref t1.templateData, ref similarity);

float threshold = 0.0f;

FSDK.GetMatchingThresholdAtFAR(0.01f, ref threshold); // set FAR to 1%

if (similarity > threshold)

{

match = true;

break;

}

ii++;

}

con.Close();

if (match)

{

StringFormat format = new StringFormat();

format.Alignment = StringAlignment.Center;

gr.DrawString(userName, new System.Drawing.Font("Arial", 16),

new System.Drawing.SolidBrush(System.Drawing.Color.LightGreen),

facePosition.xc, facePosition.yc + facePosition.w \* 0.55f, format);

label3.Text = userName;

}

else

{

label3.Text = "UnKnow FACE";

}

break;

}

}

// display current frame

pictureBox1.Image = frameImage;

GC.Collect(); // collect the garbage after the deletion

// make UI controls accessible

Application.DoEvents();

}

FSDKCam.CloseVideoCamera(cameraHandle);

FSDKCam.FinalizeCapturing();

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

//programState = ProgramState.psRemember;

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

mouseX = e.X;

mouseY = e.Y;

}

private void pictureBox1\_MouseLeave(object sender, EventArgs e)

{

mouseX = 0;

mouseY = 0;

}

private void button2\_Click(object sender, EventArgs e)

{

needClose = true;

}

private void label2\_Click(object sender, EventArgs e)

{

}

string actype;

private void button3\_Click(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

if (label3.Text == "UnKnow FACE")

{

MessageBox.Show("Please Register Your Name");

}

else

{

needClose = true;

MessageBox.Show("Face Info Saved!");

NewUser nn = new NewUser();

nn.userid = label3.Text;

nn.Show();

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

}

int acc;

private void button6\_Click(object sender, EventArgs e)

{

}

private void btnRemember\_Click(object sender, EventArgs e)

{

faceTemplates.Clear();

programState = ProgramState.psRemember;

label1.Text = "Look at the camera";

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace AuthenticationSystem

{

public partial class Home : Form

{

public Home()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

AdminLogin bb = new AdminLogin();

bb.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

}

public static Random r = new Random();

public static int number;

private void button4\_Click(object sender, EventArgs e)

{

//work

//if (textBox1.Text == "")

//{

// //Response.Write("first enter you count number");

// MessageBox.Show("");

//}

//else

//{

// number = Convert.ToInt32(textBox1.Text);

// List<int> available = new List<int>(number);

// for (int i = 1; i <= number; i++)

// available.Add(i);

// List<int> result = new List<int>(number);

// while (available.Count > 0)

// {

// int index = r.Next(available.Count);

// result.Add(available[index]);

// available.RemoveAt(index);

// }

// listBox1.Items.Clear();

// for (int i = 0; i < result.Count; i++)

// {

// // Response.Write(result[i] + "-");

// listBox1.Items.Add(result[i].ToString());

// }

//}

}

private void button2\_Click(object sender, EventArgs e)

{

Login ll = new Login();

ll.Show();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button3\_Click\_1(object sender, EventArgs e)

{

Random rr = new Random();

int i = rr.Next(0, 9);

// label2.Text = i.ToString();

}

private void label2\_Click(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace AuthenticationSystem

{

public partial class InputName : Form

{

public string userName;

public Bitmap bmp;

public InputName()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

userName = textBox1.Text;

}

private void button2\_Click(object sender, EventArgs e)

{

}

private void comboBox1\_Enter(object sender, EventArgs e)

{

}

private void InputName\_Load(object sender, EventArgs e)

{

}

int acc;

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "")

{

MessageBox.Show("Please Enter Name");

}

else

{

this.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using Luxand;

using System.Data.SqlClient;

using System.IO;

using System.Net;

using System.Net.Mail;

using System.Drawing.Imaging;

namespace AuthenticationSystem

{

public partial class Login : Form

{

public string uname;

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=|Datadirectory|\authedb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

// program states: whether we recognize faces, or user has clicked a face

enum ProgramState { psNormal, psRemember, psRecognize }

ProgramState programState = ProgramState.psRecognize;

struct FaceTemplate

{ // single template

public byte[] templateData;

}

List<FaceTemplate> faceTemplates;

string path = Path.GetDirectoryName(Application.ExecutablePath).ToString();

String cameraName;

bool needClose = false;

string userName;

String TrackerMemoryFile;

int mouseX = 0;

int mouseY = 0;

// WinAPI procedure to release HBITMAP handles returned by FSDKCam.GrabFrame

[DllImport("gdi32.dll")]

static extern bool DeleteObject(IntPtr hObject);

public Login()

{

InitializeComponent();

}

private void Login\_Load(object sender, EventArgs e)

{

if (FSDK.FSDKE\_OK != FSDK.ActivateLibrary("gyYgVWQTSzjiuGB/hH8dKgg0QrrIuhoHdfUCzD9rY+vru3WRZsaezTX6YWj9osdI/cmxY1NSdLkyWuugMPCxUG7/xNLegHLeaUpzVyKpDkaWL8tJIUsIL7xv9bhmgifPbAyTDuxF3VGxXmHkv/L/MStf9kdXV/A1vVvT93QC4vQ="))

{

MessageBox.Show("Please run the License Key Wizard (Start - Luxand - FaceSDK - License Key Wizard)", "Error activating FaceSDK", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

FSDK.InitializeLibrary();

FSDKCam.InitializeCapturing();

string[] cameraList;

int count;

FSDKCam.GetCameraList(out cameraList, out count);

if (0 == count)

{

MessageBox.Show("Please attach a camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

cameraName = cameraList[0];

FSDKCam.VideoFormatInfo[] formatList;

FSDKCam.GetVideoFormatList(ref cameraName, out formatList, out count);

timer1.Start();

}

IList<string> strList = new List<string>();

private void button1\_Click(object sender, EventArgs e)

{

this.button1.Enabled = false;

int cameraHandle = 0;

int r = FSDKCam.OpenVideoCamera(ref cameraName, ref cameraHandle);

if (r != FSDK.FSDKE\_OK)

{

MessageBox.Show("Error opening the first camera", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Application.Exit();

}

// set realtime face detection parameters

FSDK.SetFaceDetectionParameters(false, false, 100);

FSDK.SetFaceDetectionThreshold(3);

// list where we store face templates

// faceTemplates = new List();

faceTemplates = new List<FaceTemplate>();

while (!needClose)

{

// faceTemplates.Clear();

Int32 imageHandle = 0;

if (FSDK.FSDKE\_OK != FSDKCam.GrabFrame(cameraHandle, ref imageHandle)) // grab the current frame from the camera

{

Application.DoEvents();

continue;

}

FSDK.CImage image = new FSDK.CImage(imageHandle);

Image frameImage = image.ToCLRImage();

Graphics gr = Graphics.FromImage(frameImage);

FSDK.TFacePosition facePosition = image.DetectFace();

// if a face is detected, we can recognize it

if (facePosition.w != 0)

{

gr.DrawRectangle(Pens.LightGreen, facePosition.xc - facePosition.w / 2, facePosition.yc - facePosition.w / 2,

facePosition.w, facePosition.w);

// create a new face template

FaceTemplate template = new FaceTemplate();

template.templateData = new byte[FSDK.TemplateSize];

FaceTemplate template1 = new FaceTemplate();

if (programState == ProgramState.psRemember || programState == ProgramState.psRecognize)

template.templateData = image.GetFaceTemplateInRegion(ref facePosition);

switch (programState)

{

case ProgramState.psNormal: // normal state - do nothing

break;

case ProgramState.psRemember: // Remember Me state - store facial templates

label1.Text = "Templates stored: " + faceTemplates.Count.ToString();

faceTemplates.Add(template);

if (faceTemplates.Count > 9)

{

// get the user name

InputName inputName = new InputName();

inputName.ShowDialog();

userName = inputName.userName;

cmd = new SqlCommand("insert into facetb values(@Name,@face)", con);

cmd.Parameters.AddWithValue("@Name", userName);

cmd.Parameters.AddWithValue("@face", template.templateData);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Record Save!");

programState = ProgramState.psRecognize;

}

break;

case ProgramState.psRecognize: // recognize the user

bool match = false;

con.Open();

cmd = new SqlCommand("select \* from facetb ORDER BY id ASC ", con);

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

template1.templateData = (byte[])dr["face"];

faceTemplates.Add(template1);

strList.Add(dr["Name"].ToString());

}

con.Close();

int ii = 0;

foreach (FaceTemplate t in faceTemplates)

{

float similarity = 0.0f;

FaceTemplate t1 = t;

FSDK.MatchFaces(ref template.templateData, ref t1.templateData, ref similarity);

float threshold = 0.0f;

FSDK.GetMatchingThresholdAtFAR(0.01f, ref threshold); // set FAR to 1%

if (similarity > threshold)

{

userName = strList[ii].ToString();

label3.Text = strList[ii].ToString();

match = true;

break;

}

ii++;

}

con.Close();

if (match)

{

StringFormat format = new StringFormat();

format.Alignment = StringAlignment.Center;

gr.DrawString(userName, new System.Drawing.Font("Arial", 16),

new System.Drawing.SolidBrush(System.Drawing.Color.LightGreen),

facePosition.xc, facePosition.yc + facePosition.w \* 0.55f, format);

abc = 0;

send1();

}

else

{

label3.Text = "UnKnow FACE";

send();

abcd = 0;

}

break;

}

}

// display current frame

pictureBox1.Image = frameImage;

GC.Collect(); // collect the garbage after the deletion

// make UI controls accessible

Application.DoEvents();

}

FSDKCam.CloseVideoCamera(cameraHandle);

FSDKCam.FinalizeCapturing();

}

private void button2\_Click(object sender, EventArgs e)

{

needClose = true;

}

int abc,abcd;

private void send()

{

abc++;

//label4.Text = abc.ToString();

if (abc == 100)

{

//Random aa = new Random();

//int aaa = aa.Next(1111, 9999);

sendmessage("7904395266", " UnKnow User");

// sendmessage("7904395266", " UnKnow User");

string to = "mahendiran15798@gmail.com";

string from = "sampletest685@gmail.com";

string password = "mailtest2";

using (MailMessage mm = new MailMessage(from, to))

{

mm.Subject = "Alert";

mm.Body = "UnKnow User Access ";

Image image = pictureBox1.Image;

System.IO.MemoryStream stream = new System.IO.MemoryStream();

image.Save(stream, ImageFormat.Jpeg);

stream.Position = 0;

mm.Attachments.Add(new Attachment(stream, "Screenshot.jpg"));

mm.IsBodyHtml = false;

SmtpClient smtp = new SmtpClient();

smtp.Host = "smtp.gmail.com";

smtp.EnableSsl = true;

NetworkCredential NetworkCred = new NetworkCredential(from, password);

smtp.UseDefaultCredentials = true;

smtp.Credentials = NetworkCred;

smtp.Port = 587;

smtp.Send(mm);

MessageBox.Show("Mail Send!");

}

//needClose = true;

//Otp op = new Otp();

//op.otp = aaa.ToString();

//op.Show();

//this.Close();

abc = 0;

}

}

private void send1()

{

abcd++;

//label4.Text = abc.ToString();

if (abcd == 0)

{

MessageBox.Show("Door Open!");

abcd = 0;

}

}

private void button3\_Click(object sender, EventArgs e)

{

}

public void sendmessage(string targetno, string message)

{

String query = "http://bulksms.mysmsmantra.com:8080/WebSMS/SMSAPI.jsp?username=fantasy5535&password=1163974702&sendername=Sample&mobileno=" + targetno + "&message=" + message;

WebClient client = new WebClient();

Stream sin = client.OpenRead(query);

// Response.Write("<script> alert('Message Send') </script>");

MessageBox.Show("Message Send");

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

int start=0;

private void timer1\_Tick(object sender, EventArgs e)

{

start++;

if (start == 50)

{

button1.PerformClick();

start = 0;

timer1.Stop();

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Data;

namespace AuthenticationSystem

{

public partial class NewUser : Form

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=|Datadirectory|\authedb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public string userid;

public NewUser()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string gender;

if (radioButton1.Checked == true)

{

gender = radioButton1.Text;

}

else

{

gender = radioButton2.Text;

}

cmd = new SqlCommand("select \* from regtb where UserId='" + textBox8.Text + "' ", con);

con.Open();

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

MessageBox.Show("Already Register This userid ");

}

else

{

dr.Close();

cmd = new SqlCommand("insert into regtb values(@FirstName,@LastName,@Gender,@Dob,@Age,@MobileNo,@Email,@Address,@UserId)", con);

cmd.Parameters.AddWithValue("@FirstName", textBox1.Text);

cmd.Parameters.AddWithValue("@LastName", textBox2.Text);

cmd.Parameters.AddWithValue("@Gender", gender);

cmd.Parameters.AddWithValue("@Dob", dateTimePicker1.Text);

cmd.Parameters.AddWithValue("@Age", textBox3.Text);

cmd.Parameters.AddWithValue("@MobileNo", textBox4.Text);

cmd.Parameters.AddWithValue("@Email", textBox5.Text);

cmd.Parameters.AddWithValue("@Address", textBox6.Text);

cmd.Parameters.AddWithValue("@UserId", userid);

cmd.ExecuteNonQuery();

MessageBox.Show("Record Save!");

}

con.Close();

}

private void textBox4\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode < Keys.D0 || e.KeyCode > Keys.D9)

{

if (e.KeyCode < Keys.NumPad0 || e.KeyCode > Keys.NumPad9)

{

if (e.KeyCode != Keys.Back)

{

//nonnumberenter = true;

string abc = "Please enter numbers only.";

textBox5.Text = "";

DialogResult result1 = MessageBox.Show(abc.ToString(), "Validate numbers", MessageBoxButtons.OK);

}

}

}

if (Control.ModifierKeys == Keys.Shift)

{

//nonnumberenter = true;

string abc = "Please enter numbers only.";

DialogResult result1 = MessageBox.Show(abc.ToString(), "Validate numbers", MessageBoxButtons.OK);

}

}

private void textBox6\_Enter(object sender, EventArgs e)

{

string pattern = null;

pattern = "^([0-9a-zA-Z]([-\\.\\w]\*[0-9a-zA-Z])\*@([0-9a-zA-Z][-\\w]\*[0-9a-zA-Z]\\.)+[a-zA-Z]{2,9})$";

if (System.Text.RegularExpressions.Regex.IsMatch(textBox5.Text, pattern))

{

//MessageBox.Show("Valid Email address ");

}

else

{

textBox4.Text = "";

MessageBox.Show("Not a valid Email address ");

}

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

int age = DateTime.Today.Year - dateTimePicker1.Value.Year;

textBox3.Text = age.ToString();

if (age < 18)

{

//MessageBox.Show("Age Limit Low!");

}

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

textBox8.Text = "";

// textBox9.Text = "";

}

private void NewUser\_Load(object sender, EventArgs e)

{

textBox8.Text = userid;

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace AuthenticationSystem

{

public partial class UserDetails : Form

{

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;AttachDbFilename=|Datadirectory|\authedb.mdf;Integrated Security=True;User Instance=True");

SqlCommand cmd;

public UserDetails()

{

InitializeComponent();

}

private void UserDetails\_Load(object sender, EventArgs e)

{

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

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.Refresh();

}

private void label12\_Click(object sender, EventArgs e)

{

}

}

}