using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

using System.Windows.Forms;

using FireSharp.Config;

using FireSharp.Response;

using FireSharp.Interfaces;

namespace GatewayExpress

{

public partial class Admins : Form

{

Thread th; //declaration of thread as th

int panelWidth, taken=0;

bool Hidden;

int Counter = 0;

DateTime time = DateTime.Now;

IFirebaseConfig ifc = new FirebaseConfig()

{

AuthSecret = "P2k1udVSBnRvxFKNk1Ve5OVZo8A4x90jWAEFXqYH",

BasePath = "https://lancaster-new-city-default-rtdb.firebaseio.com/"

};

IFirebaseClient client;

public Admins()

{

InitializeComponent();

panelWidth = PanelSlide.Width;

Hidden = false;

}

void reload()

{

btnAdd.Enabled = true;

btnUpdate.Enabled = false;

btnDelete.Enabled = false;

try

{

client = new FireSharp.FirebaseClient(ifc);

DataTable dt = new DataTable();

dt.Columns.Add("ID");

dt.Columns.Add("USERNAME");

dt.Columns.Add("PASSWORD");

dataGridView1.Rows.Clear();

FirebaseResponse res = client.Get(@"ADMIN\_USER/ADMIN\_COUNTER");

Counter = int.Parse(res.ResultAs<String>());

for (int i = Counter; i > 0; i--)

{

var res3 = client.Get(@"ADMIN\_ACCOUNT/" + i);

MyUser std = res3.ResultAs<MyUser>();

if (std.USERNAME != null)//(std!=null)

{

if (std.ACCOUNT\_STATUS == "ACTIVE")

{

dt.Rows.Add(i, std.USERNAME, std.PASSWORD);

}

}

else { }

}

foreach (DataRow item in dt.Rows)

{

dataGridView1.Rows.Add(item.ItemArray);

}

}

catch

{

//MessageBox.Show("No Internet or Connection Problem");

}

dataGridView1.BorderStyle = BorderStyle.None;

dataGridView1.AlternatingRowsDefaultCellStyle.BackColor = Color.FromArgb(238, 239, 249);

dataGridView1.CellBorderStyle = DataGridViewCellBorderStyle.SingleHorizontal;

dataGridView1.DefaultCellStyle.SelectionBackColor = Color.DarkTurquoise;

dataGridView1.DefaultCellStyle.SelectionForeColor = Color.WhiteSmoke;

dataGridView1.BackgroundColor = Color.White;

dataGridView1.EnableHeadersVisualStyles = false;

dataGridView1.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.None;

dataGridView1.ColumnHeadersDefaultCellStyle.BackColor = Color.FromArgb(36, 37, 45);

dataGridView1.ColumnHeadersDefaultCellStyle.ForeColor = Color.White;

}

void clear()

{

txtUsername.Text = "";

txtPassword.Text ="";

MyUser erasure = new MyUser()

{

ID = "",

USERNAME = "",

PASSWORD = "",

};

taken = 0;

}

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

{

this.Close(); //command for closing the application

th = new Thread(openSplashScreen); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Close(); //command for closing the application

th = new Thread(openNewEToll); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Close(); //command for closing the application

th = new Thread(openNewLoad); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Close(); //command for closing the application

th = new Thread(openNewDrivers); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

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

{

this.Refresh();

}

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

{

this.Close(); //command for closing the application

th = new Thread(openNewForm1); //creating thread

th.SetApartmentState(ApartmentState.STA); //setting thread as STA

th.Start(); //start the thread

}

private void openNewDashBoad(object obj)

{

Application.Run(new Dashboard()); //Run Dashboard

}

private void openNewEToll(object obj)

{

Application.Run(new EToll()); //Run EToll Tab

}

private void openNewLoad(object obj)

{

Application.Run(new Load()); //Run Load Tab

}

private void openNewDrivers(object obj)

{

Application.Run(new Drivers()); //Run Drivers or User's Info Tab

}

private void openSplashScreen(object obj)

{

Application.Run(new SplashScreen()); //Run SplashScreen

}

private void openNewForm1(object obj)

{

Application.Run(new Form1()); //Run Form1 or Login Tab

}

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

{

reload();

}

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

{

try

{

FirebaseResponse res1 = client.Get(@"ADMIN\_USER/ADMIN\_COUNTER");

Counter = int.Parse(res1.ResultAs<String>());

for (int i = Counter; i > 0; i--)

{

var res3 = client.Get(@"ADMIN\_ACCOUNT/" + i);

MyUser std = res3.ResultAs<MyUser>();

if (std.USERNAME == txtUsername.Text)

{

taken++;

}

}

if (taken == 0)

{

if (txtUsername.Text != "" || txtPassword.Text != "")

{

Counter++;

MyUser std1 = new MyUser()

{

ADMIN\_COUNTER = Counter.ToString()

};

var setter1 = client.Set("ADMIN\_USER", std1);

MyUser std3 = new MyUser()

{

USERNAME = txtUsername.Text,

PASSWORD = txtPassword.Text,

ACCOUNT\_STATUS = "ACTIVE"

};

var setter2 = client.Set("ADMIN\_ACCOUNT/" + Counter.ToString(), std3);

MessageBox.Show("Data Inserted Successfully!");

clear();

reload();

}

else

{

MessageBox.Show("Username or Password can't be empty!");

}

}

else

{

MessageBox.Show("Username already taken!");

clear();

}

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

clear();

}

}

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

{

try

{

if (txtUsername.Text != "" || txtPassword.Text != "")

{

MyUser std3 = new MyUser()

{

USERNAME = txtUsername.Text,

PASSWORD = txtPassword.Text,

ACCOUNT\_STATUS = "ACTIVE"

};

var setter2 = client.Update("ADMIN\_ACCOUNT/" + Counter.ToString(), std3);

MessageBox.Show("Data Updated Successfully!");

clear();

reload();

}

else

{

MessageBox.Show("Username or Password can't be empty!");

}

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

clear();

}

}

private void dataGridView1\_MouseClick(object sender, MouseEventArgs e)

{

//Get the data from data grid view and load it to the textboxes

Counter = int.Parse(dataGridView1.SelectedRows[0].Cells[0].Value.ToString().Trim());

txtUsername.Text = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();

txtPassword.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();

btnAdd.Enabled = false;

btnUpdate.Enabled = true;

btnDelete.Enabled = true;

}

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

{

try

{

MyUser std3 = new MyUser()

{

USERNAME = txtUsername.Text,

PASSWORD = txtPassword.Text,

ACCOUNT\_STATUS = "DEACTIVATED"

};

var setter2 = client.Update("ADMIN\_ACCOUNT/" + Counter.ToString(), std3);

MessageBox.Show("Data Deleted Successfully!");

clear();

reload();

}

catch (Exception ex1)

{

MessageBox.Show(ex1.Message);

clear();

}

}

}

}