**Products Project**

**Description Page**

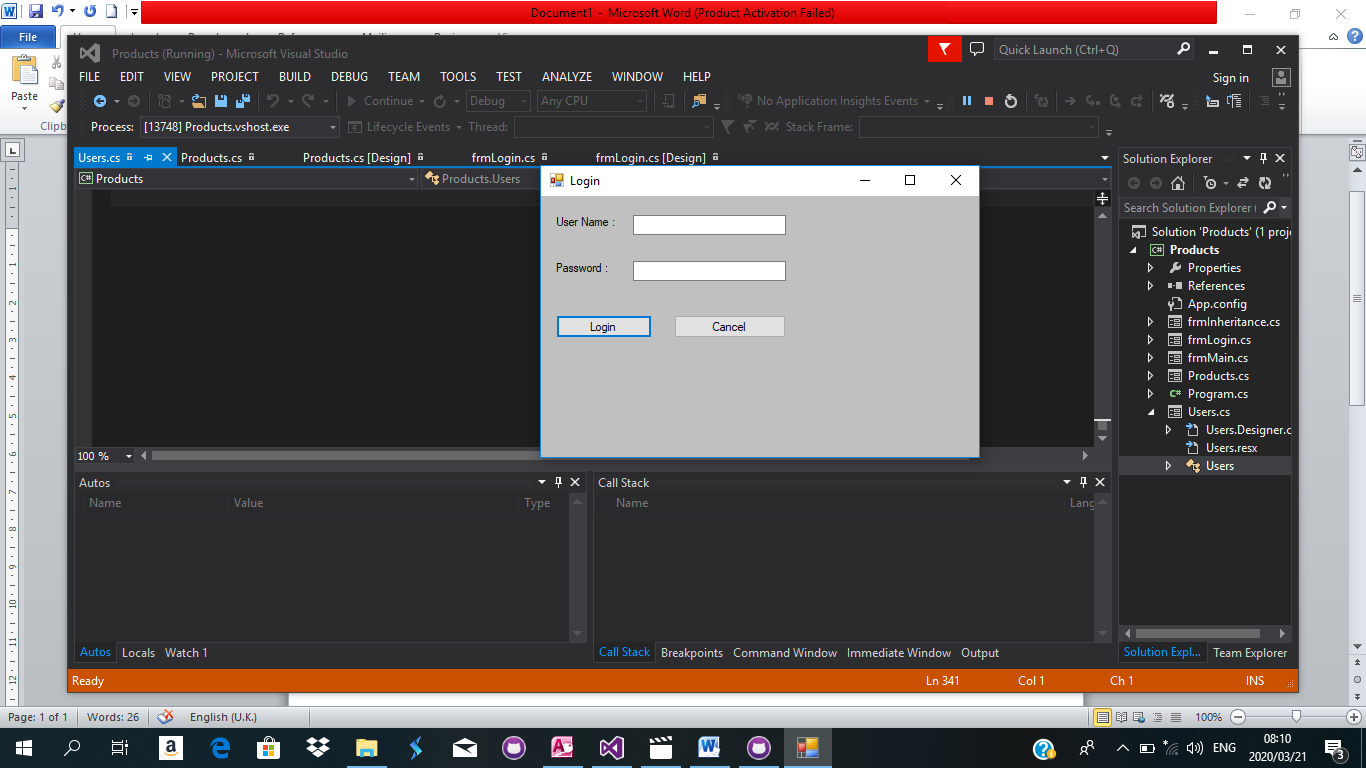
Project name: Products

Built by: Kagiso Mailula

Link to Github: thulikagiso030@gmail.com

Password: 8712thulisile

Login Form Section



using System;

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.Windows.Forms;

using System.Data.Sql;

using System.Data.SqlClient;

using System.Data.Odbc;

namespace Products

{

public partial class frmLogin : frmInheritance

{

bool boolUserExists = false;

string strAccessConnectionString = "Driver= {Microsoft Access Driver (\*.mdb)}; Dbq=Products.mdb;Uid=Admin;Pwd=;";

public frmLogin()

{

InitializeComponent();

}

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

{

if (btnLogin.Text == "Login")

{

if (txtUsername.Text == "")

{

MessageBox.Show("User Name field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtPassword.Text == "")

{

MessageBox.Show("Password field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

checkIfUserExists();

if(boolUserExists == false)

{

//createUser();

//controlsLoad();

ClearTextBoxes();

//loadUser();

}

else if (boolUserExists == true)

{

string UserName = txtUsername.Text;

string Password = txtPassword.Text;

string query = "SELECT UserName, Password FROM Users WHERE UserName=@UserName and Password=@Password";

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = query;

cmd.Parameters.AddWithValue("@UserName", UserName);

cmd.Parameters.AddWithValue("@Password", Password);

if (UserName == "Admin" && Password == "54321")

{

frmMain frmMain = new frmMain();

frmMain.Show();

this.Hide();

}

else

{

MessageBox.Show("Username Or Password are Incorrect", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//SqlConnection conn = new SqlConnection(strAccessConnectionString);

//conn.Open();

//string UserName = txtUsername.Text;

//string Password = txtPassword.Text;

//string query = "SELECT UserName, Password FROM Users WHERE UserName=@UserName and Password=@Password";

//SqlCommand cmd = new SqlCommand();

//cmd.CommandType = CommandType.Text;

//cmd.CommandText = query;

//cmd.Parameters.AddWithValue("@UserName", UserName);

//cmd.Parameters.AddWithValue("@Password", Password);

}

}

//frmMain frmMain = new frmMain();

//frmMain.Show();

//this.Hide();

}

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

{

Application.Exit();

}

private void ClearTextBoxes()

{

txtUsername.Text = "";

txtPassword.Text = "";

}

private void checkIfUserExists()

{

string query = "SELECT \* FROM Users WHERE UserName='" + txtUsername.Text + "'";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcCommand cmd;

OdbcDataReader dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

//OdbcDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

boolUserExists = true;

}

dr.Close();

OdbcConnection.Close();

dr.Dispose();

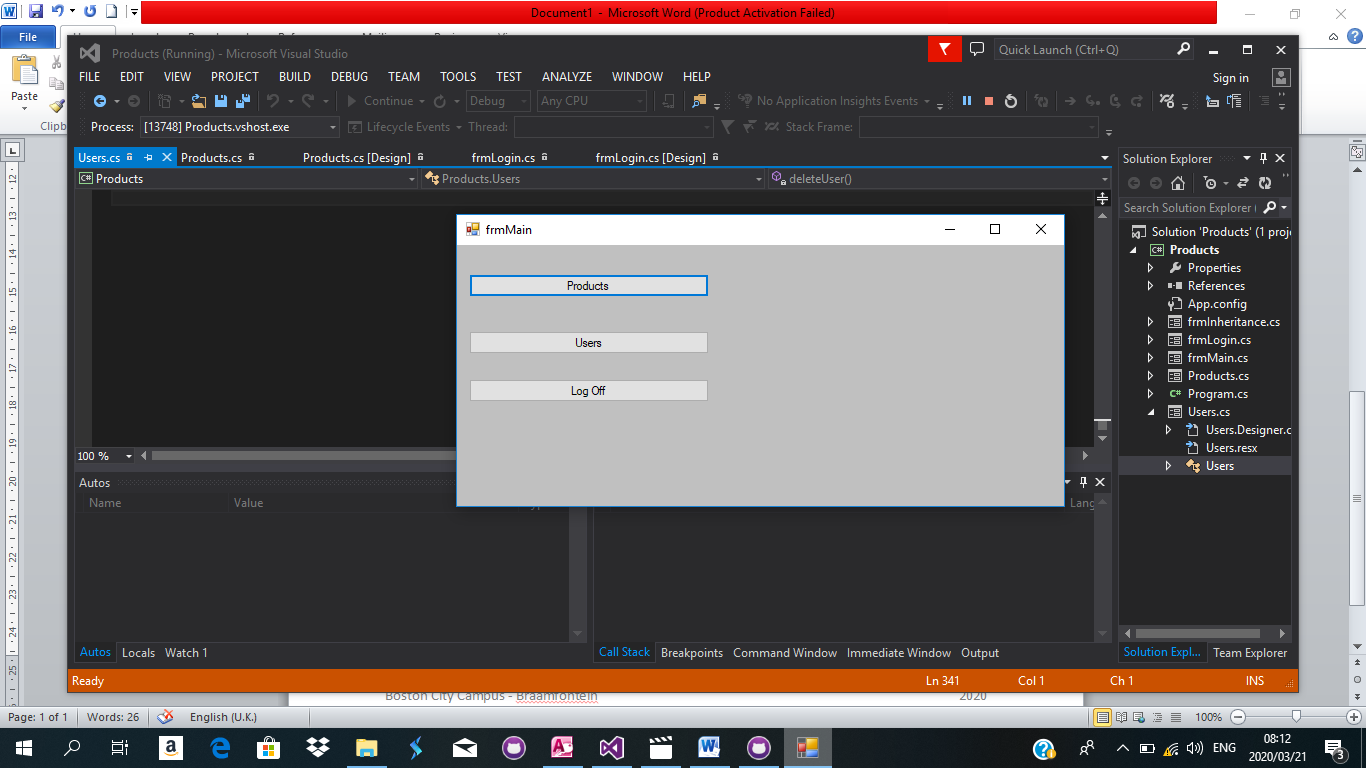
OdbcConnection.Dispose();

}

}

}

Main Form Section



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.Windows.Forms;

namespace Products

{

public partial class frmMain : frmInheritance

{

public frmMain()

{

InitializeComponent();

}

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

{

}

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

{

frmLogin frmLogOff = new frmLogin();

frmLogOff.Show();

this.Hide();

}

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

{

Products frmProducts = new Products();

frmProducts.Show();

this.Hide();

}

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

{

Users frmUser = new Users();

frmUser.Show();

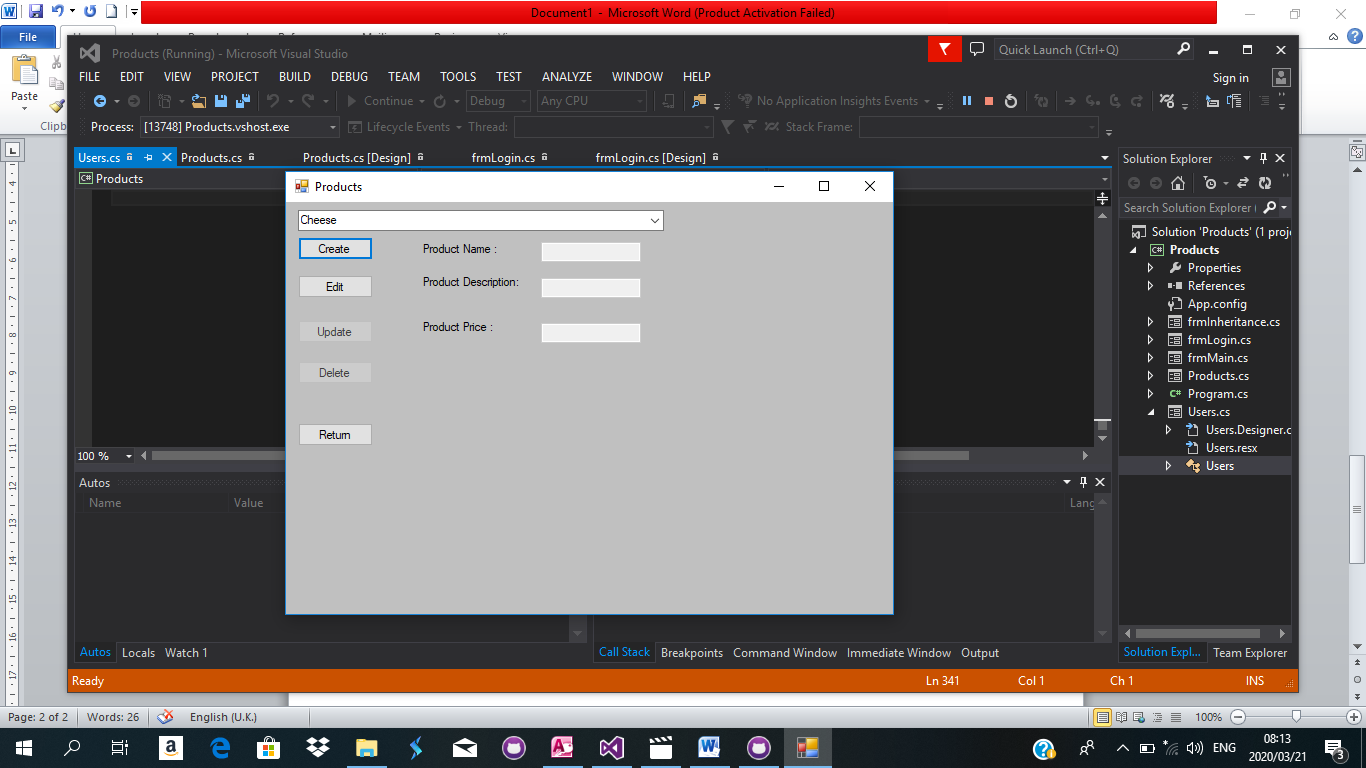
this.Hide();

}

}

}

Product Form Section



using System;

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.Windows.Forms;

using System.Data.Sql;

using System.Data.SqlClient;

using System.Data.Odbc;

namespace Products

{

public partial class Products : frmInheritance

{

string strProductName;

string strProductDescription;

double dblProductPrice;

bool boolProductExists = false;

int intProductID = 0;

string strAccessConnectionString = "Driver= {Microsoft Access Driver (\*.mdb)}; Dbq=Products.mdb;Uid=Admin;Pwd=;";

public Products()

{

InitializeComponent();

}

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

{

controlsLoad();

loadProducts();

}

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

{

frmMain frmMain = new frmMain();

frmMain.Show();

this.Hide();

}

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

{

if (btnCreate.Text == "Save")

{

if(txtProductName.Text == "")

{

MessageBox.Show("Product Name field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtProductDescription.Text == "")

{

MessageBox.Show("Product Description field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtProductPrice.Text == "")

{

MessageBox.Show("Product Price field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

checkIfProductsExists();

if(boolProductExists == false)

{

createProduct();

controlsLoad();

ClearTextBoxes();

loadProducts();

}

else if (boolProductExists == true)

{

MessageBox.Show("Product already exists", "Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

}

else if (btnCreate.Text == "Create")

{

controlsCreate();

}

}

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

{

editProduct();

controlsEdit();

}

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

{

updateProduct();

controlsLoad();

ClearTextBoxes();

loadProducts();

}

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

{

deleteProduct();

controlsLoad();

ClearTextBoxes();

loadProducts();

}

private void controlsLoad()

{

txtProductDescription.Enabled = false;

txtProductName.Enabled = false;

txtProductPrice.Enabled = false;

cboProducts.Enabled = true;

btnCreate.Enabled = true;

btnDelete.Enabled = false;

btnEdit.Enabled = true;

btnReturn.Enabled = true;

btnUpdate.Enabled = false;

btnCreate.Text = "Create";

}

private void controlsCreate()

{

txtProductDescription.Enabled = true;

txtProductName.Enabled = true;

txtProductPrice.Enabled = true;

cboProducts.Enabled = false;

btnCreate.Enabled = true;

btnDelete.Enabled = false;

btnEdit.Enabled = false;

btnReturn.Enabled = false;

btnUpdate.Enabled = false;

btnCreate.Text = "Save";

}

private void controlsEdit()

{

txtProductDescription.Enabled = true;

txtProductName.Enabled = true;

txtProductPrice.Enabled = true;

cboProducts.Enabled = false;

btnCreate.Enabled = false;

btnDelete.Enabled = true;

btnEdit.Enabled = false;

btnReturn.Enabled = false;

btnUpdate.Enabled = true;

// btnCreate.Text = "Save";

}

private void ClearTextBoxes()

{

txtProductDescription.Text = "";

txtProductName.Text = "";

txtProductPrice. Text= "";

}

private void loadProducts()

{

cboProducts.DataSource = null;

cboProducts.Items.Clear();

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcConnection.ConnectionString = strAccessConnectionString;

string query = "SELECT ProductName FROM Products";

OdbcCommand cmd = new OdbcCommand(query, OdbcConnection);

OdbcConnection.Open();

OdbcDataReader dr = cmd.ExecuteReader();

AutoCompleteStringCollection ProductCollection = new AutoCompleteStringCollection();

while(dr.Read())

{

ProductCollection.Add(dr.GetString(0));

}

cboProducts.DataSource = ProductCollection;

OdbcConnection.Close();

}

private void createProduct()

{

string query = "SELECT \* FROM Products WHERE ID=0";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcDataAdapter da = new OdbcDataAdapter(query, OdbcConnection);

DataSet ds = new DataSet();

DataTable dt = new DataTable();

DataRow dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

da.Fill(ds,"Products");

dt = ds.Tables["Products"];

try

{

dr = dt.NewRow();

dr["ProductName"] = txtProductName.Text;

dr["ProductDescription"] = txtProductDescription.Text;

dr["Price"] = txtProductPrice.Text;

dt.Rows.Add(dr);

OdbcCommandBuilder cmd = new OdbcCommandBuilder(da);

da.Update(ds, "Products");

}

catch (Exception EX)

{

MessageBox.Show(EX.Message.ToString());

}

finally

{

OdbcConnection.Close();

OdbcConnection.Dispose();

}

}

private void checkIfProductsExists()

{

string query = "SELECT \* FROM Products WHERE ProductName='" + txtProductName.Text + "'";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcCommand cmd;

OdbcDataReader dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

//OdbcDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

boolProductExists = true;

}

dr.Close();

OdbcConnection.Close();

dr.Dispose();

OdbcConnection.Dispose();

}

private void editProduct()

{

string query = "SELECT \* FROM Products WHERE ProductName='" + cboProducts.Text + "'";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcCommand cmd;

OdbcDataReader dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

if (dr.Read())

{

intProductID = dr.GetInt32(0);

txtProductName.Text = dr.GetString(1);

txtProductDescription.Text = dr.GetString(2);

txtProductPrice.Text = dr.GetString(3);

}

dr.Close();

OdbcConnection.Close();

dr.Dispose();

OdbcConnection.Dispose();

}

private void updateProduct()

{

string query = "SELECT \* FROM Products WHERE ID=" + intProductID;

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcDataAdapter da = new OdbcDataAdapter(query, OdbcConnection);

DataSet ds = new DataSet("Products");

da.FillSchema(ds, SchemaType.Source, "Products");

da.Fill(ds, "Products");

DataTable dt;

dt = ds.Tables["Products"];

DataRow dr;

dr = dt.NewRow();

try

{

dr = dt.Rows.Find(intProductID);

dr.BeginEdit();

dr["ProductName"] = txtProductName.Text;

dr["ProductDescription"] = txtProductDescription.Text;

dr["Price"] = txtProductPrice.Text;

dr.EndEdit();

OdbcCommandBuilder cmd = new OdbcCommandBuilder(da);

da.Update(ds, "Products");

//dt.Rows.Add(dr);

//OdbcCommandBuilder cmd = new OdbcCommandBuilder(da);

}

catch (Exception EX)

{

MessageBox.Show(EX.Message.ToString());

}

finally

{

OdbcConnection.Close();

OdbcConnection.Dispose();

}

}

private void deleteProduct()

{

string query = "DELETE FROM Products WHERE ID=" + intProductID;

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcCommand cmd;

OdbcDataReader dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

//OdbcDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

}

dr.Close();

OdbcConnection.Close();

dr.Dispose();

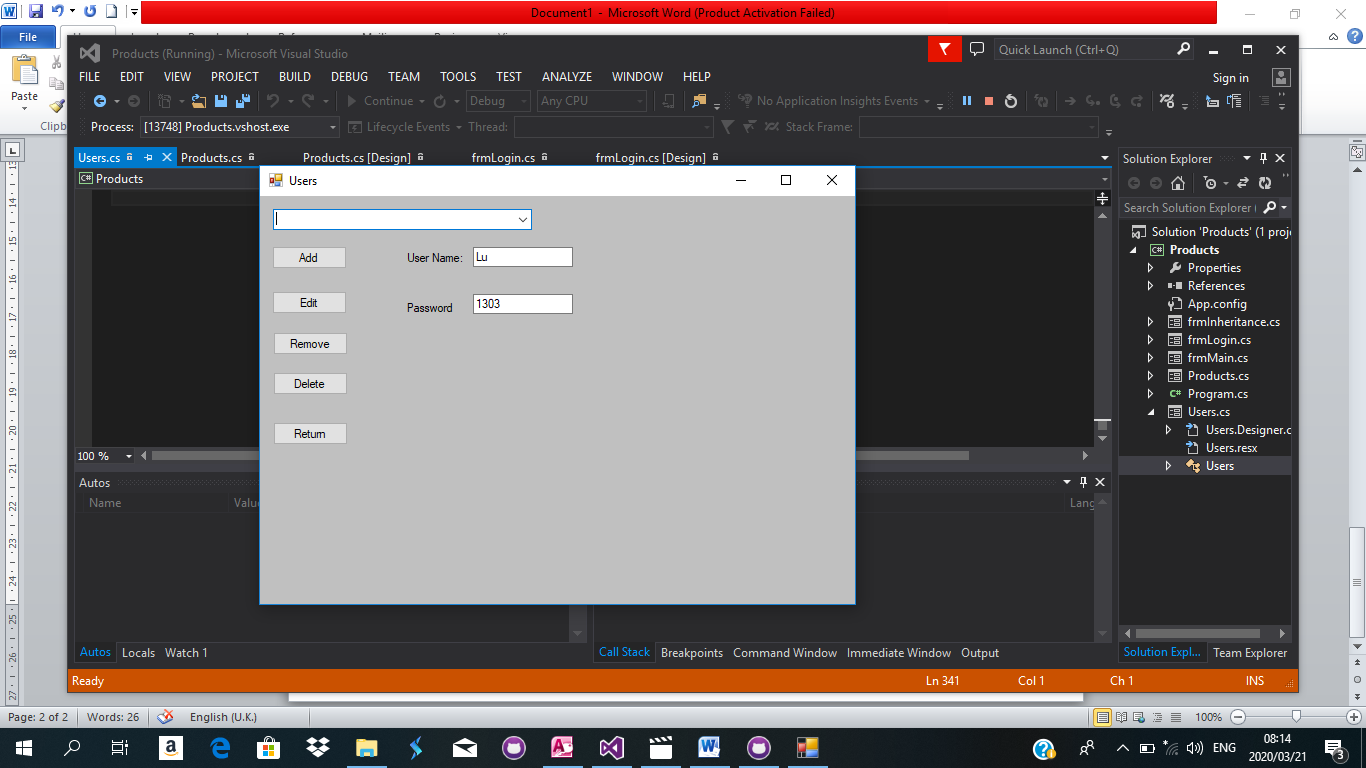
OdbcConnection.Dispose();

}

}

}

User form Section



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.Windows.Forms;

using System.Data.Sql;

using System.Data.SqlClient;

using System.Data.Odbc;

namespace Products

{

public partial class Users : frmInheritance

{

bool boolUserExists = false;

int intUserID = 0;

string strAccessConnectionString = "Driver= {Microsoft Access Driver (\*.mdb)}; Dbq=Products.mdb;Uid=Admin;Pwd=;";

public Users()

{

InitializeComponent();

}

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

{

}

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

{

frmMain frmMain = new frmMain();

frmMain.Show();

this.Hide();

}

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

{

if (btnAdd.Text == "ADD")

{

if(txtUsername.Text == "")

{

MessageBox.Show("User Name field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (txtPassword.Text == "")

{

MessageBox.Show("Password field cannot be left empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

checkIfUserExists();

if(boolUserExists == false)

{

createUser();

controlsLoad();

ClearTextBoxes();

loadUser();

}

else if (boolUserExists == true)

{

MessageBox.Show("User already exists", "Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

}

else if (btnAdd.Text == "Add")

{

controlsAdd();

}

}

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

{

editUser();

controlsEdit();

}

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

{

ClearTextBoxes();

}

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

{

deleteUser();

controlsLoad();

ClearTextBoxes();

loadUser();

}

private void controlsLoad()

{

txtUsername.Enabled = false;

txtPassword.Enabled = false;

cboUsers.Enabled = true;

btnAdd.Enabled = true;

btnDelete.Enabled = false;

btnEdit.Enabled = true;

btnReturn.Enabled = true;

btnRemove.Enabled = false;

btnAdd.Text = "ADD";

}

private void controlsAdd()

{

txtUsername.Enabled = true;

txtPassword.Enabled = true;

cboUsers.Enabled = false;

btnAdd.Enabled = true;

btnDelete.Enabled = false;

btnEdit.Enabled = false;

btnReturn.Enabled = false;

btnRemove.Enabled = false;

btnAdd.Text = "ADD";

}

private void controlsEdit()

{

txtUsername.Enabled = true;

txtPassword.Enabled = true;

cboUsers.Enabled = false;

btnAdd.Enabled = false;

btnDelete.Enabled = true;

btnEdit.Enabled = false;

btnReturn.Enabled = false;

btnRemove.Enabled = true;

// btnCreate.Text = "Save";

}

private void ClearTextBoxes()

{

txtUsername.Text = "";

txtPassword.Text = "";

}

private void loadUser()

{

cboUsers.DataSource = null;

cboUsers.Items.Clear();

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcConnection.ConnectionString = strAccessConnectionString;

string query = "SELECT UserName FROM Users";

OdbcCommand cmd = new OdbcCommand(query, OdbcConnection);

OdbcConnection.Open();

OdbcDataReader dr = cmd.ExecuteReader();

AutoCompleteStringCollection ProductCollection = new AutoCompleteStringCollection();

while(dr.Read())

{

ProductCollection.Add(dr.GetString(0));

}

cboUsers.DataSource = ProductCollection;

OdbcConnection.Close();

}

private void createUser()

{

string query = "SELECT \* FROM Users WHERE ID=0";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcDataAdapter da = new OdbcDataAdapter(query, OdbcConnection);

DataSet ds = new DataSet();

DataTable dt = new DataTable();

DataRow dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

da.Fill(ds,"Users");

dt = ds.Tables["Users"];

try

{

dr = dt.NewRow();

dr["UserName"] = txtUsername.Text;

dr["Password"] = txtPassword.Text;

dt.Rows.Add(dr);

OdbcCommandBuilder cmd = new OdbcCommandBuilder(da);

da.Update(ds, "Users");

}

catch (Exception EX)

{

MessageBox.Show(EX.Message.ToString());

}

finally

{

OdbcConnection.Close();

OdbcConnection.Dispose();

}

}

private void checkIfUserExists()

{

string query = "SELECT \* FROM Users WHERE UserName='" + txtUsername.Text + "'";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcCommand cmd;

OdbcDataReader dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

//OdbcDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

boolUserExists = true;

}

dr.Close();

OdbcConnection.Close();

dr.Dispose();

OdbcConnection.Dispose();

}

private void editUser()

{

string query = "SELECT \* FROM Users WHERE UserName='" + cboUsers.Text + "'";

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcDataAdapter da = new OdbcDataAdapter(query, OdbcConnection);

OdbcCommand cmd;

OdbcDataReader dr;

//OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

if (dr.Read())

{

intUserID = dr.GetInt32(0);

txtUsername.Text = dr.GetString(1);

txtPassword.Text = dr.GetString(2);

}

//dr.Close();

//OdbcConnection.Close();

//dr.Dispose();

//OdbcConnection.Dispose();

DataSet ds = new DataSet("Users");

da.FillSchema(ds, SchemaType.Source, "Users");

da.Fill(ds, "Users");

DataTable dt;

dt = ds.Tables["Users"];

DataRow dr1;

dr1 = dt.NewRow();

try

{

dr1 = dt.Rows.Find(intUserID);

dr1.BeginEdit();

dr1["UserName"] = txtUsername.Text;

dr1["Password"] = txtPassword.Text;

dr1.EndEdit();

OdbcCommandBuilder command = new OdbcCommandBuilder(da);

da.Update(ds, "Users");

}

catch (Exception EX)

{

MessageBox.Show(EX.Message.ToString());

}

finally

{

OdbcConnection.Close();

OdbcConnection.Dispose();

}

}

private void deleteUser()

{

string query = "DELETE FROM Users WHERE ID=" + intUserID;

OdbcConnection OdbcConnection = new OdbcConnection();

OdbcCommand cmd;

OdbcDataReader dr;

OdbcConnection.ConnectionString = strAccessConnectionString;

OdbcConnection.Open();

cmd = new OdbcCommand(query, OdbcConnection);

dr = cmd.ExecuteReader();

//OdbcDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

}

dr.Close();

OdbcConnection.Close();

dr.Dispose();

OdbcConnection.Dispose();

}

}

}

Database

Microsoft Access

