using DBConnect;

using System.Data.SqlClient;

using System.Text;

using System.Configuration;

class Program:ConfigurationSection

{

public static void Main(string[] args)

{

StoreManagement sm = new StoreManagement();

sm.Login();

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DBConnect

{

internal class StoreManagement

{

SqlConnection conn;

public void OpenConn()

{

string cnnstr = "data source=LAPTOP-BMALL679\\SQLEXPRESS;Initial catalog=StoreManagement;integrated security=SSPI";

conn = new SqlConnection(cnnstr);

try

{

conn.Open();

Console.WriteLine("Connection Established");

}

catch (SqlException ex)

{

Console.WriteLine("Connection not established");

}

}

public void Login()

{

Console.WriteLine("Enter 1 for admin login or 2 for customer login");

int ch = Convert.ToInt32(Console.ReadLine());

switch (ch)

{

case 1:

string retry = "yes";

while (retry == "yes")

{

Console.WriteLine("Admin Login");

Console.WriteLine("Enter admin userid and password");

string uid = Console.ReadLine();

string pass = Console.ReadLine();

if ((uid == "admin") && (pass == "admin"))

{

OpenConn();

AdminManipluation();

retry= "no";

}

else

{

Console.WriteLine("Incorrect userid/password");

}

}

break;

case 2:

default:

Console.WriteLine("Enter option 1 or 2");

break;

}

}

public void InsertRecord(string product\_id, string product\_name, int quantity\_available, double cost)

{

SqlCommand cmd2 = new SqlCommand("insert into ProductData values (@value1, @value2, @value3, @value4)", conn);

cmd2.Parameters.AddWithValue("@value1", product\_id);

cmd2.Parameters.AddWithValue("@value2", product\_name);

cmd2.Parameters.AddWithValue("@value3", quantity\_available);

cmd2.Parameters.AddWithValue("@value4", cost);

if (conn != null)

{

try

{

int cou = cmd2.ExecuteNonQuery();

Console.WriteLine(cou + " Row inserted");

}

catch (SqlException ex)

{

if (ex.Number == 2627) // unique constraint violation

{

Console.WriteLine($"Error: product\_id:{product\_id} already exists!");

}

else

{

Console.WriteLine("Error: " + ex.Message);

}

}

}

}

public void DeleteRecord(string product\_id)

{

using (SqlCommand command = new SqlCommand("delete from ProductData where product\_id=@product\_id", conn))

{

command.Parameters.AddWithValue("@product\_id", product\_id);

int rowsAffected = command.ExecuteNonQuery();

Console.WriteLine("Rows affected: " + rowsAffected);

}

}

public void CloseConn()

{

if (conn != null)

{

conn.Close();

Console.WriteLine("Connection closed");

}

}

public void AdminManipluation()

{

bool exit = false;

while (exit != true)

{

Console.WriteLine("1. Display products \t2. Insert products \t3. Update products \t4. Delete products \t5.exit");

int ch = Convert.ToInt32(Console.ReadLine());

switch (ch)

{

case 1:

SqlCommand cmd = new SqlCommand("select \* from ProductData", conn);

SqlDataReader sdr;

if (conn != null)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Dataset is empty");

}

while (sdr.Read())

{

Console.WriteLine(sdr["product\_id"].ToString() + "\t" + sdr["product\_name"].ToString() + "\t" + sdr["quantity\_available"] +"\t\t"+ sdr["cost"]);

}

sdr.Close();

}

break;

case 2:

Console.WriteLine("Enter the product id");

string product\_id = Console.ReadLine();

Console.WriteLine("Enter the product name");

string product\_name = Console.ReadLine();

Console.WriteLine("Enter the quantity available");

int quantity\_available = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the cost");

double cost = Convert.ToDouble(Console.ReadLine());

InsertRecord(product\_id, product\_name, quantity\_available, cost);

break;

case 3:

Console.WriteLine("Enter the product id");

string product\_id1 = Console.ReadLine();

Console.WriteLine("Enter the field to be updated");

Console.WriteLine("1. Product name 2. Quantity 3. cost");

int field = Convert.ToInt32(Console.ReadLine());

switch(field)

{

case 1:

Console.WriteLine("Enter the product name");

string prod\_name\_update = Console.ReadLine();

using (SqlCommand command = new SqlCommand("update ProductData set product\_name=@prod\_name\_update where product\_id=@product\_id1", conn))

{

command.Parameters.AddWithValue("@prod\_name\_update", prod\_name\_update);

command.Parameters.AddWithValue("@product\_id1", product\_id1);

int rowsAffected = command.ExecuteNonQuery();

Console.WriteLine("Rows affected: " + rowsAffected);

}

break;

case 2:

Console.WriteLine("Enter the product quantity");

int prod\_quantity\_update = Convert.ToInt32(Console.ReadLine());

using (SqlCommand command = new SqlCommand("update ProductData set quantity\_available=@prod\_quantity\_update where product\_id=@product\_id1", conn))

{

command.Parameters.AddWithValue("@prod\_quantity\_update", prod\_quantity\_update);

command.Parameters.AddWithValue("@product\_id1", product\_id1);

int rowsAffected = command.ExecuteNonQuery();

Console.WriteLine("Rows affected: " + rowsAffected);

}

break;

case 3:

Console.WriteLine("Enter the product cost");

double prod\_cost\_update = Convert.ToDouble(Console.ReadLine());

//UpdateRecord(product\_id1, prod\_cost\_update);

using (SqlCommand command = new SqlCommand("update ProductData set cost=@prod\_cost\_update where product\_id=@product\_id1", conn))

{

command.Parameters.AddWithValue("@prod\_cost\_update", prod\_cost\_update);

command.Parameters.AddWithValue("@product\_id1", product\_id1);

int rowsAffected = command.ExecuteNonQuery();

Console.WriteLine("Rows affected: " + rowsAffected);

}

break;

}

break;

case 4:

Console.WriteLine("Enter the product id to be deleted");

string product\_id\_del = Console.ReadLine();

DeleteRecord(product\_id\_del);

break;

case 5:

exit = true;

CloseConn();

break;

default:

Console.WriteLine("Give proper value");

break;

}

}

}

}

}