using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Net.NetworkInformation;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Xml.Linq;

using System.Data;

namespace Login

{

class ConnectedArchitecture

{

static SqlConnection con = new SqlConnection(System.Configuration.ConfigurationSettings.AppSettings["MyDBConnect"]);

public void Menu()

{

string Achoice;

do

{

Console.WriteLine("Enter Your Choice from the list\n1. Insert\n2. Delete\n3. Update\n4. Select");

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

try

{

switch (ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

update();

break;

case 4:

print();

break;

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

System.Console.Write("Do you want to perform more operation from the lis? Y/N: ");

Achoice = Console.ReadLine();

} while (Achoice == "y" || Achoice == "Y");

}

public static void insert()

{

Console.Write("Enter the Employee Id : ");

string userIdInput = Console.ReadLine();

Console.Write("Enter Employee Name : ");

string userNameInput = Console.ReadLine();

Console.Write("Enter Employee age in years : ");

string userAgeInput = Console.ReadLine();

SqlConnection con = new SqlConnection(System.Configuration.ConfigurationSettings.AppSettings["MyDBConnect"]);

con.Open();

string cmdStr = "insert into employee values(@userIdInput,@userNameInput,@userAgeInput)";

SqlCommand com = new SqlCommand(cmdStr, con);

com.Parameters.AddWithValue("@userIdInput", userIdInput);

com.Parameters.AddWithValue("@userNameInput", userNameInput);

com.Parameters.AddWithValue("@userAgeInput", userAgeInput);

com.ExecuteNonQuery();

Console.WriteLine("Data Inserted..");

}

public static void print()

{

con.Open();

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

SqlDataReader reader = cmd.ExecuteReader();

Console.WriteLine("");

Console.WriteLine("The Deatails in DB are");

while (reader.Read())

{

Console.WriteLine(reader[0] + " " + reader[1] + " "+ reader[2]);

}

Console.ReadLine();

}

public static void delete()

{

Console.WriteLine("Enter your choice from the list\n1. delete by Id\n2. delete by name");

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

switch (choice)

{

case 1: deleteById();

break;

case 2: deleteByName();

break;

default: Console.WriteLine("Invalid Choice");

break;

}

}

static public void deleteById()

{

Console.Write("Enter Employee Id to Delete : ");

string e\_id = Console.ReadLine();

con.Open();

string cmdStr = "delete from employee where employeeId=@e\_id";

SqlCommand com = new SqlCommand(cmdStr, con);

com.Parameters.AddWithValue("@e\_id", e\_id);

com.ExecuteNonQuery();

Console.WriteLine("Data Deleted" +

"..");

con.Close();

}

static public void deleteByName()

{

Console.Write("Enter Employee Name to Delete : ");

string e\_name = Console.ReadLine();

con.Open();

string cmdStr = "delete from employee where emp\_name=@e\_name";

SqlCommand com = new SqlCommand(cmdStr, con);

com.Parameters.AddWithValue("@e\_name", e\_name);

com.ExecuteNonQuery();

Console.WriteLine("Data Deleted" +

"..");

}

public static void update()

{

Console.WriteLine("Enter your choice from the list\n1. Update by Id\n2. Update by name");

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

switch (choice)

{

case 1:

UpdateById();

break;

case 2:

UpdateByName();

break;

default:

Console.WriteLine("Invalid Choice");

break;

}

}

public static void UpdateById()

{

Console.Write("\tEnter the Id from where you want to change the Name : ");

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

Console.WriteLine("Enter name");

string name = Console.ReadLine();

con.Open();

//here name is getting changed using id

string cmdStr = "update employee set emp\_name=@e\_name where employeeId=@c\_id";

SqlCommand com = new SqlCommand(cmdStr, con);

com.Parameters.AddWithValue("@e\_name", name);

com.Parameters.AddWithValue("@c\_id", id);

com.ExecuteNonQuery();

Console.WriteLine("Data Updated..");

}

public static void UpdateByName()

{

Console.Write("Enter the Name from where you want to change the ID : ");

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

Console.WriteLine("Enter name");

string name = Console.ReadLine();

con.Open();

//here id is getting changed using name

string cmdStr = "update employee set employeeId=@c\_id where emp\_name=@e\_name";

SqlCommand com = new SqlCommand(cmdStr, con);

com.Parameters.AddWithValue("@e\_name", name);

com.Parameters.AddWithValue("@c\_id", id);

com.ExecuteNonQuery();

Console.WriteLine("Data Updated..");

}

}

class DisconnectedArchitecture

{

static string connectionString = System.Configuration.ConfigurationSettings.AppSettings["MyDBConnect"];

static DataSet dataSet = new DataSet();//local Db ready

public void Menu()

{

string AChoice;

Console.WriteLine("Enter Your Choice from the list\n1.Press 1 for Insert\n2.Press 2 for Delete\n3.Press 3 for Update\n4.Press 4 for Select");

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

switch (ch)

{

case 1:

Insert();

break;

case 2:

Delete();

break;

case 3:

Update();

break;

case 4:Print();

break;

}

}

void Print()

{

try {

using (SqlConnection connection = new SqlConnection(connectionString))

{

string selectStatement = "SELECT \* FROM employee";

SqlDataAdapter adapter = new SqlDataAdapter(selectStatement, connection);

adapter.Fill(dataSet, "employee");//MyTable---> tblCustomerDetails

DataTable usersTableRead = dataSet.Tables["employee"];

/\*usersTableRead.PrimaryKey = new DataColumn[] { usersTableRead.Columns["Id"] };\*/

Console.WriteLine("Records in the Users table:");

foreach (DataRow row in usersTableRead.Rows)

{

int id = (int)row["employeeId"];

string name = (string)row["emp\_name"];

string age = (string)row["age"];

Console.WriteLine($"Id: {id}, Name: {name},Age: {age}");

}

}

}catch (Exception ex)

{

Console.WriteLine(ex);

}

}

void Update()

{

string selectStatement = "SELECT \* FROM employee";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlDataAdapter adapter = new SqlDataAdapter(selectStatement, connection);

adapter.Fill(dataSet, "employee");//MyTable---> tblCustomerDetails

DataTable usersTableRead = dataSet.Tables["employee"];

usersTableRead.PrimaryKey = new DataColumn[] { usersTableRead.Columns["employeeId"] };

Console.Write("Enter ID : ");

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

Console.Write("Enter Name to update : ");

string name = Console.ReadLine();

//Update

DataRow dr1 = usersTableRead.Rows.Find(id);

dr1["emp\_name"] = name;

SqlCommandBuilder cb = new SqlCommandBuilder(adapter);

adapter.Update(usersTableRead);

}

}

void Delete()

{

string selectStatement = "SELECT \* FROM employee";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlDataAdapter adapter = new SqlDataAdapter(selectStatement, connection);

adapter.Fill(dataSet, "employee");//MyTable---> tblCustomerDetails

DataTable usersTableRead = dataSet.Tables["employee"];

usersTableRead.PrimaryKey = new DataColumn[] { usersTableRead.Columns["employeeId"] };

//Delete

Console.Write("Enter ID : ");

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

DataRow dr2 = usersTableRead.Rows.Find(id);

dr2.Delete();

SqlCommandBuilder cb = new SqlCommandBuilder(adapter);

adapter.Update(usersTableRead);

}

}

void Insert()

{

string selectStatement = "SELECT \* FROM employee";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlDataAdapter adapter = new SqlDataAdapter(selectStatement, connection);

DataSet dataSet = new DataSet();//local Db ready

adapter.Fill(dataSet, "employee");//MyTable---> tblCustomerDetails

DataTable usersTableRead = dataSet.Tables["employee"];

DataRow dr = usersTableRead.NewRow();

Console.Write("Enter ID of the new Employee : ");

dr["employeeId"] = Convert.ToInt32(Console.ReadLine());//assigning Id Column MyTable

Console.Write("Enter name of the new Employee : ");

dr[1] = Console.ReadLine();//assigning Name column MyTable

Console.Write("Enter age of the new Employee : ");

dr[2] = Console.ReadLine();//assigning age column MyTable

usersTableRead.Rows.Add(dr);// temporary insert

SqlCommandBuilder cb = new SqlCommandBuilder(adapter);//

adapter.Update(usersTableRead);//real insert

}

}

}

internal class Program

{

static SqlConnection con = new SqlConnection(System.Configuration.ConfigurationSettings.AppSettings["MyDBConnect"]);

static void Main(string[] args)

{

Console.WriteLine("\t\t Welcome To Admin System\t");

Console.WriteLine("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

Console.WriteLine("\t1. Here you can Login using Admin Creadentials or Can also register a new admin\n\t2. You can add New Employee\n\t3. You can Update Employee Details\n\t4. You can also remove Employee from Database ");

Console.Write("\nAre you a registered Admin? Y/N : ");

string registered = Console.ReadLine();

ConnectedArchitecture ca = new ConnectedArchitecture();

DisconnectedArchitecture dca = new DisconnectedArchitecture();

if (registered == "Y" || registered == "y")

{

Console.WriteLine("\n\t\tIf you are Registered then \n\t");

Console.Write("Enter User Id : ");

string userInputId = Console.ReadLine();

Console.Write("Enter your Password : ");

string userInputPass = Console.ReadLine();

con.Open();

string query = "SELECT COUNT(\*) FROM registration WHERE userId = @username AND pass = @password";

SqlCommand cmd = new SqlCommand(query, con);

cmd.Parameters.AddWithValue("@username", userInputId);

cmd.Parameters.AddWithValue("@password", userInputPass);

int count = (int)cmd.ExecuteScalar();

if (count > 0)

{

Console.WriteLine("\n\t\tLogin successful!\n");

Console.WriteLine("Enter your choice\nPress 1. for CRUD operation using Connected Architecture \nPress 2. for CRUD operation using Disconnected Architecture");

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

switch(ch)

{

case 1:ca.Menu();

break;

case 2:dca.Menu();

break;

default: Console.WriteLine("Invalid Input");

break;

}

}

else

{

Console.WriteLine("Invalid username or password.");

}

}

else

{

string checkPassword = @"^[a-zA-Z0-9#$%^&\*()\_+=]{8,15}$";

string checkEmail = @"^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$";

Console.Write("\n\tEnter the Details to Register a New Admin\n\n");

Console.Write("Enter Admin Mail Id : ");

string userInputId = Console.ReadLine();

Console.Write("Enter your Password : ");

string userInputPass = Console.ReadLine();

Console.Write("Enter your Job Profile : ");

string userInputRole = Console.ReadLine();

Regex r = new Regex(checkPassword);

Regex r2 = new Regex(checkEmail);

if (r.IsMatch(userInputPass) && r2.IsMatch(userInputId))

{

//Console.WriteLine("Valid Input");

try

{

con.Open();

string cmdStr = "insert into registration values(@userInputId,@userInputPass,@userInputRole)";

SqlCommand com = new SqlCommand(cmdStr, con);

com.Parameters.AddWithValue("@userInputId", userInputId);

com.Parameters.AddWithValue("@userInputPass", userInputPass);

com.Parameters.AddWithValue("@userInputRole", userInputRole);

com.ExecuteNonQuery();

Console.WriteLine("Data Inserted..");

con.Close();

}

catch(Exception ex)

{

Console.WriteLine(ex);

}

}

else

{

Console.WriteLine("\n\tNeed to Enter your Mail/Password Again \n\tBecause password length is too short or did not used '@' in the mail id");

}

}

}

}

}