using System;

using System.Collections.Generic;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

List<User> resultantList = new List<User>();

UserRepository userRepository = new UserRepository();

//user list

resultantList = userRepository.Users();

foreach (var item in resultantList)

{

Console.WriteLine(item.Id + "\t" + item.Name + "\t" + item.Country + "\t" + item.Isactive);

}

//check active users

resultantList = userRepository.ActiveUsers();

foreach (var item in resultantList)

{

Console.WriteLine(item.Id + "\t" + item.Name + "\t" + item.Country + "\t" + item.Isactive);

}

//add user

int forNewUserInMain = userRepository.forNewUser;

forNewUserInMain++;

resultantList = userRepository.AddUser(new User() { Id = forNewUserInMain, Name = "Ram" + forNewUserInMain, Country = "TN", Isactive = true });

foreach (var item in resultantList)

{

Console.WriteLine(item.Id + "\t" + item.Name + "\t" + item.Country + "\t" + item.Isactive);

}

//delete users

Console.WriteLine("Enter the user id to delete it's details:");

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

resultantList = userRepository.DeleteUser(idMainDelete);

foreach (var item in resultantList)

{

Console.WriteLine(item.Id + "\t" + item.Name + "\t" + item.Country + "\t" + item.Isactive);

}

//view user

Console.WriteLine("Enter the user id to view it's details:");

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

var getUser = userRepository.GetUser(idMainView);

if (getUser == null)

{

Console.WriteLine("Please,enter correct userId next time!!!");

}

else

{

Console.WriteLine(getUser.Id + "\t" + getUser.Name + "\t" + getUser.Country + "\t" + getUser.Isactive);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Net.Http.Headers;

using System.Text;

namespace ConsoleApp1

{

public class User

{

public string Name { get; set; }

public int Id { get; set; }

public string Country { get; set; }

public bool Isactive { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

namespace ConsoleApp1

{

public class UserRepository : IUserRepostiroy

{

private List<User> users = new List<User>();

private List<User> ActiveUser = new List<User>();

public int forNewUser;

public UserRepository()

{

string connectionString = "Data Source=INDU-PC\\SQLEXPRESS;database=persons;Trusted\_Connection=True";

SqlConnection con = new SqlConnection();

con.ConnectionString = connectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "select \* from users select \* from duplicateuser";

cmd.Connection = con;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds);

int count = ds.Tables[0].Rows.Count;

if (count > 0)

{

foreach (DataRow row in ds.Tables[0].Rows)

{

users.Add(new User()

{

Id = Convert.ToInt32(row["Id"]),

Name = Convert.ToString(row["Name"]),

Country = Convert.ToString(row["Country"]),

Isactive = Convert.ToBoolean(row["isactive"])

});

}

}

}

public List<User> ActiveUsers()

{

ActiveUser = users.Where(a => a.Isactive).ToList();

return ActiveUser;

}

public List<User> AddUser(User user)

{

forNewUser++;

users.Add(user);

return users;

//throw new NotImplementedException();

}

public List<User> DeleteUser(int id)

{

for (int i = 1; i <= users.Count; i++)

{

try

{

if (users[i].Id == id)

{

users.RemoveAt(i);

break;

}

}

catch

{

Console.WriteLine("User not available");

}

}

return users;

//throw new NotImplementedException();

}

public User GetUser(int id)

{

var specificUserLinq = users.Where(a => a.Id == id).FirstOrDefault();

return specificUserLinq;

// throw new NotImplementedException();

}

public List<User> Users()

{

return users;

// throw new NotImplementedException();

}

}

}

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp1

{

public interface IUserRepostiroy

{

List<User> Users();

User GetUser(int id);

List<User> DeleteUser(int id);

List<User> ActiveUsers();

List<User> AddUser(User user);

}

}

------------------------------------------------------------------------------------------------------------------------------------------