**CONSOLE APPLICATION-EMPLOYEE MANAGEMENT SYSTEM**

***RITHISH GURU G***

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Configuration;

using System.Net.Mail;

using System.Text.RegularExpressions;

using System.Threading;

using System.Collections;

public class AccountNumber

{

public virtual void Name()

{

Console.WriteLine("Your Name is : Rithish");

}

}

public class EmployeeName : AccountNumber

{

public override void Name()

{

int Flag = 0;

string Name = Console.ReadLine();

Regex regex = new Regex("[A-Z][a-z]{5,10}");

MatchCollection employeeName =

regex.Matches(Name);

foreach (char letter in Name)

{

if (!Char.IsLetter(letter) && Name.Contains(""))

{

Flag = 1;

}

}

if (Flag == 1)

{

Console.WriteLine("Name should not contain any special characters or numbers or spaces");

Console.WriteLine("Please Re-Enter proper Name");

this.Name();

}

else if (employeeName.Count != 0)

{

Console.WriteLine("Your Name is : " + Name);

}

else if (!Char.IsUpper(Name[0]))

{

Console.WriteLine("First Letter of Your Name must be in Capital Letter");

Console.WriteLine("Please Enter Proper Name");

this.Name();

}

else

{

Console.WriteLine("Your Name must contain atleast 5 letters");

Console.WriteLine("Please Enter Proper Name");

this.Name();

}

}

}

public class EmployeeDetails:EmployeeName

{

public string AccountNumber\_Result { get; private set; }

public string EmployeeName\_Result { get; private set; }

public string DateOfBirth\_Result { get; private set; }

public string DateOfJoining\_Result { get; private set; }

public string MailID\_Result { get; private set; }

public string MobileNumber\_Result { get; private set; }

public string Experience\_Result { get; private set; }

public string Place\_Result { get; private set; }

static void Main(String[] args)

{

string connectionString\_1 = @"Data Source=DESKTOP-7P47GVS\SQLEXPRESS;Initial Catalog=Employee;Integrated Security=True";

// string connectionString\_1 = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

try

{

using (SqlConnection connection = new SqlConnection(connectionString\_1))

{

connection.Open();

string AddRecord = "INSERT INTO EmployeeT2 values(@AccountNumber,@EmployeeName,@DateOfBirth,@DateOfJoining,@MailId,@MobileNumber,@Experience,@Role,@Place)";

SqlCommand command = new SqlCommand(AddRecord, connection);

EmployeeDetails employeeDetails = new EmployeeDetails();

EmployeeDetails employeeDetail = new EmployeeDetails();

employeeDetail.AccountNumberOfEmployee();

command.Parameters.AddWithValue("@AccountNumber", employeeDetails.EmployeeAccountNumber());

employeeDetail.Employee();

employeeDetail.NameOfEmployee();

command.Parameters.AddWithValue("@EmployeeName", employeeDetails.Name1());

employeeDetail.Employee();

employeeDetail.DateOfBirthOfEmployee();

command.Parameters.AddWithValue("@DateOfBirth", employeeDetails.DateofBirth());

employeeDetail.Employee();

employeeDetail.DateOfJoinOfEmployee();

command.Parameters.AddWithValue("@DateOfJoining", employeeDetails.DateOfJoining());

employeeDetail.Employee();

employeeDetail.MailOfEmployee();

command.Parameters.AddWithValue("@MailId", employeeDetails.Mail());

employeeDetail.Employee();

employeeDetail.MobileNumberOfEmployee();

command.Parameters.AddWithValue("@MobileNumber", employeeDetails.GetMobileNumber());

employeeDetail.Employee();

employeeDetail.ExperienceOfEmployee();

command.Parameters.AddWithValue("@Experience", employeeDetails.Experience());

employeeDetail.Employee();

employeeDetail.JobRoleOfEmployee();

command.Parameters.AddWithValue("@Role", employeeDetails.Role());

employeeDetail.Employee();

employeeDetail.PlaceOfEmployee();

command.Parameters.AddWithValue("@Place", employeeDetails.Place());

employeeDetail.Employee();

employeeDetail.ResponseAdded();

SqlDataReader dataReader = command.ExecuteReader();

if (dataReader.HasRows)

{

while (dataReader.Read())

{

//display retrieved record (first column only/string value)

Console.WriteLine(dataReader.GetString(100));

}

}

dataReader.Close();

}

}

catch (Exception )

{

//display error message

Console.WriteLine("SORRY!!! YOUR RECORD HAS NOT BEEN ADDED");

}

}

public string EmployeeAccountNumber()

{

EmployeeDetails ed = new EmployeeDetails();

string AccountsNumber = Console.ReadLine();

Regex regex = new Regex("[a][c][e][1-9][0-9][0-9][0-9]");

MatchCollection givenNumber =

regex.Matches(AccountsNumber);

if (givenNumber.Count != 0 && AccountsNumber.Length <= 7)

{

AccountNumber\_Result = AccountsNumber;

Console.WriteLine("Your Account Number is : " + AccountNumber\_Result);

}

else

{

Console.WriteLine("Your Account Number should be in the form 'ace' followed by four numbers");

Console.WriteLine("Please Enter Proper Number");

EmployeeAccountNumber();

}

return AccountNumber\_Result;

}

public string Name1()

{

int Flag = 0;

string Name = Console.ReadLine();

Regex regex = new Regex("[A-Z][a-z]{5,10}");

MatchCollection employeeName =

regex.Matches(Name);

foreach (char letter in Name)

{

if (!Char.IsLetter(letter) && Name.Contains(""))

{

Flag = 1;

}

}

if (Flag == 1)

{

Console.WriteLine("Name should not contain any special characters or numbers or spaces");

Console.WriteLine("Please Re-Enter proper Name");

Name1();

}

else if (employeeName.Count != 0)

{

EmployeeName\_Result = Name;

Console.WriteLine("Your Name is : " + Name);

}

else if (!Char.IsUpper(Name[0]))

{

Console.WriteLine("First Letter of Your Name must be in Capital Letter");

Console.WriteLine("Please Enter Proper Name");

Name1();

}

else

{

Console.WriteLine("Your Name must contain atleast 5 letters");

Console.WriteLine("Please Enter Proper Name");

Name1();

}

return EmployeeName\_Result;

}

public string DateofBirth()

{

string DateOfBirth = Console.ReadLine();

try

{

Regex regex = new Regex("[1-2]{1}[0-9]{1}[0-9]{1}[0-9]{1}[-]{1}[1-12]{1}[1-12]{1}[-]{1}[1-31]{1}[1-31]{1}");

MatchCollection datebirth =

regex.Matches(DateOfBirth);

DateTime bday = Convert.ToDateTime(DateOfBirth);

DateTime today = DateTime.Today;

int Age = today.Year - bday.Year;

if (bday > today.AddYears(Age))

{

Console.Write("You have entered future date!!!\n");

Console.WriteLine("Your Age must be between 18 and 60\n");

Console.WriteLine("Please Re-Enter your correct date of birth");

DateofBirth();

}

else if (bday == today)

{

Console.WriteLine("You cannot enter today's date!!!\n");

Console.WriteLine("Your Age must be between 18 and 60\n");

Console.WriteLine("Please Re-Enter your correct date of birth");

DateofBirth();

}

else if (Age < 18)

{

Console.Write("You are too young\n");

Console.WriteLine("Please Enter a date before 01/01/2003");

Console.WriteLine("Please Re-Enter your correct date of birth");

DateofBirth();

}

else if (datebirth.Count != 0 && DateOfBirth.Length == 10 || Age >= 18 || Age <= 60 )

{

DateOfBirth\_Result = DateOfBirth;

Console.WriteLine("Your Date of Birth is : " + DateOfBirth);

}

else

{

Console.WriteLine("You have entered wrong format of date!!!\n");

Console.WriteLine("Please Re-Enter your correct date of birth");

DateofBirth();

}

}

catch (FormatException )

{

Console.WriteLine("Your Date should be in the format of 'YYYY/MM/DD'");

Console.WriteLine("Date should not contain alphabets or Special Characters!!!!!");

Console.WriteLine("Please Re-Enter Correct Date Format");

DateofBirth();

}

return DateOfBirth\_Result;

}

public string DateOfJoining()

{

string DateofJoining = Console.ReadLine();

try

{

Regex regex = new Regex("[1-2]{1}[0-9]{1}[0-9]{1}[0-9]{1}[-]{1}[1-12]{1}[1-12]{1}[-]{1}[1-31]{1}[1-31]{1}");

MatchCollection dateJoin =

regex.Matches(DateofJoining);

DateTime dateOfJoin = Convert.ToDateTime(DateofJoining);

DateTime today1 = DateTime.Today;

int Ages = today1.Year - dateOfJoin.Year;

if (dateOfJoin > today1.AddYears(Ages))

{

Console.Write("You have entered future date!!!\n");

Console.WriteLine("Your Date Of Joining must be till today's date\n");

Console.WriteLine("Please Re-Enter your correct date of joining");

this.DateOfJoining();

}

else if (Ages > 18)

{

Console.WriteLine("Your Date Of Joining is inappropriate to your Date Of Birth");

Console.WriteLine("Please Re-Enter your correct date of joining");

this.DateOfJoining();

}

else if ((dateOfJoin.Year - Ages) < 18)

{

Console.WriteLine("Your Date Of Joining is inappropriate to your Date Of Birth");

Console.WriteLine("Please Re-Enter your correct date of joining");

this.DateOfJoining();

}

else if (dateJoin.Count != 0 || DateofJoining.Length==10)

{

DateOfJoining\_Result = DateofJoining;

Console.WriteLine("Your Date Of Joining is : " + DateofJoining);

}

else { }

}

catch (FormatException )

{

Console.WriteLine("Your Date should be in the format of 'YYYY/MM/DD'");

Console.WriteLine("Date should not contain alphabets or Special Characters!!!!!");

Console.WriteLine("Please Re-Enter Correct Date Format");

DateOfJoining();

}

return DateOfJoining\_Result;

}

public string Mail()

{

string EmailId = Console.ReadLine();

Regex regex = new Regex("[a-zA-Z0-9-\_.][@][a-z]{5,10}[.][a-z]{3}");

MatchCollection employeeName =

regex.Matches(EmailId);

if (employeeName.Count != 0)

{

MailID\_Result = EmailId;

Console.WriteLine("Your Mail PassCode is : " + EmailId);

}

else

{

Console.WriteLine("You have entered wrong format of MailId");

Console.WriteLine("Please Enter Proper Mail ID");

Mail();

}

return MailID\_Result;

}

public string GetMobileNumber()

{

string mobileInput = Console.ReadLine();

foreach (char character in mobileInput)

{

if (Char.IsLetter(character))

{

Console.WriteLine("Mobile Number should not contain Alphabets");

Console.WriteLine("Please Re-Enter proper Mobile Number");

GetMobileNumber();

}

}

Regex regex = new Regex("[^A-Za-z0-9]");

// Find match between given

// string & regular expression

MatchCollection mobile =

regex.Matches(mobileInput);

// Print Yes If the string matches

// with the Regex

if (mobile.Count != 0)

{

Console.WriteLine("Mobile Number should not contain any special characters");

Console.WriteLine("Please Re-Enter proper Mobile Number");

GetMobileNumber();

}

else if (mobileInput.Length > 10)

{

Console.WriteLine("Mobile Number should contain only 10 digits");

Console.WriteLine("Please Re-Enter proper mobile number");

GetMobileNumber();

}

else if (mobileInput.Length < 10)

{

Console.WriteLine("Mobile Number should contain 10 digits");

Console.WriteLine("Please Re-Enter proper mobile number");

GetMobileNumber();

}

else if (mobileInput[0] != '9' && mobileInput[0] != '8' && mobileInput[0] != '7' && mobileInput[0] != '6')

{

Console.WriteLine("You have entered wrong format of mobile number");

Console.WriteLine("Your mobile number should start with '9','8','7' or '6'");

Console.WriteLine("Please Re-Enter proper mobile number");

GetMobileNumber();

}

else

{

MobileNumber\_Result = mobileInput;

Console.WriteLine("Your Mobile Number is : " + mobileInput);

}

return MobileNumber\_Result;

}

string Role()

{

string EmployeeRole = Console.ReadLine();

bool Result = EmployeeRole.All(Char.IsLetter);

if (Result == true && EmployeeRole == "Analyst" || EmployeeRole == "Tester" || EmployeeRole == "Developer")

{

lock (this)

{

Console.Write("Your Job ");

Thread.Sleep(100);

Console.Write("Role ");

Console.Write(" is : ");

Console.WriteLine(EmployeeRole);

}

}

else

{

Console.WriteLine("Please Enter Roles Only among ('Tester'/'Analyst'/'Developer')");

Role();

}

return EmployeeRole;

}

public string Experience()

{

string inp= Console.ReadLine();

if(inp=="Yes")

{

Experience\_Result = inp;

Console.WriteLine("Your Experience is:" + Experience\_Result);

}

else if(inp=="No")

{

Experience\_Result = inp;

Console.WriteLine("Your Experience is:" + Experience\_Result);

}

else

{

Console.WriteLine("Please Enter options only between 'Yes'/'No'");

Experience();

}

return Experience\_Result;

}

public string Place()

{

try

{

EmployeePlace customer1 = new EmployeePlace()

{

PassCode = "Emp\_Place",

Country = "India",

};

Dictionary<string, EmployeePlace> dictionaryCustomer = new Dictionary<string, EmployeePlace>();

dictionaryCustomer.Add(customer1.PassCode, customer1);

string IdInput = Console.ReadLine();

EmployeePlace empPlace = dictionaryCustomer[IdInput];

Place\_Result = empPlace.Country;

Console.WriteLine("Your Country : " + empPlace.Country);

}

catch (KeyNotFoundException)

{

Console.WriteLine("You have entered wrong Passcode");

Console.WriteLine("Please Enter proper Passcode");

Place();

}

return Place\_Result;

}

class EmployeePlace

{

public string PassCode { get; set; }

public string Country { get; set; }

public string State { get; set; }

}

public void AccountNumberOfEmployee()

{

Console.WriteLine("Welcome!!!");

Console.WriteLine("Please Enter Your Account Number (Format:'ace1111')");

}

public void NameOfEmployee()

{

Console.WriteLine("Please Enter Your Name with first letter as Capital (Format:'Rithish')");

}

public void DateOfBirthOfEmployee()

{

Console.WriteLine("Please Enter your Date Of Birth (Format:'2001/01/16')");

}

public void DateOfJoinOfEmployee()

{

Console.WriteLine("Please Enter your Date of Joining(Format:'2019/03/22')");

}

public void MailOfEmployee()

{

Console.WriteLine("Please Enter Your Mail-ID (Format:'rithishguru78@gmail.com')");

}

public void MobileNumberOfEmployee()

{

Console.WriteLine("Please Enter Your Mobile Number (Format:'8220112659')");

}

public void ExperienceOfEmployee()

{

Console.WriteLine("Are you a fresher ?(Format:'Yes'/'No')");

}

public void JobRoleOfEmployee()

{

Console.WriteLine("Please Enter Your Job Role (Format:'Analyst'/'Tester'/'Developer')");

}

public void PlaceOfEmployee()

{

Console.WriteLine("Please Enter Your Passcode to display your Place(Format:'Emp\_Place')");

}

public void ResponseAdded()

{

Console.WriteLine("YOUR DETAILS HAVE BEEN ADDED TO THE DATABASE");

}

public void EmployeeManagementSystem()

{

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

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

Console.WriteLine("\t\t\t\t\t\tSYSTEM\t\t\t\t\t\t\t");

}

public void Employee()

{

Console.WriteLine("----------------------------------------------------------------------------------------------------------------------");

}

}

**App.config**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />

</startup>

<connectionStrings>

<add name="DefaultConnection" connectionString="Data Source = DESKTOP-7P47GVS\SQLEXPRESS; Initial Catalog = Employee; Integrated Security = True"

providerName="System.Data.SqlClient"/>

</connectionStrings>

</configuration>



