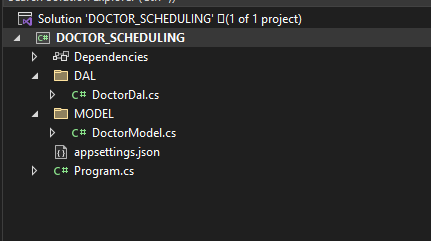
Console Application

**Doctor Scheduling**



**Program.cs code**

using DOCTOR\_SCHEDULING.DAL;

using DOCTOR\_SCHEDULING.MODEL;

using Microsoft.Extensions.Configuration;

using System;

using System.Data;

using System.IO;

namespace DOCTOR\_SCHEDULING

{

class Program

{

private static IConfiguration \_iconfiguration;

static void Main(string[] args)

{

GetAppSettingsFile();

string r;

try {

do

{

Console.Clear();

Console.WriteLine("\n\n\n\n\t\t\t\t\t\t\t\t +++++ SPEKTRA-HEALTHCARE +++++");

Console.WriteLine("\t\t\t\t\t\t\t\t1. Book appointment ");

Console.WriteLine("\t\t\t\t\t\t\t\t2. Cancel appointment ");

Console.WriteLine("Enter The Choice : ");

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

switch (input)

{

case 1:

PrintDoctor();

break;

case 2:

Cancelslot();

break;

default:

Console.WriteLine("No such option...");

break;

}

Console.WriteLine("Do you Want to Continue Y/N : ");

r=Console.ReadLine().ToLower();

}

while (r=="y");

}

catch(Exception e)

{

Console.WriteLine(e.Message);

}

}

static void GetAppSettingsFile()

{

var builder = new ConfigurationBuilder()

.SetBasePath(Directory.GetCurrentDirectory())

.AddJsonFile("appsettings.json", optional: false, reloadOnChange: true);

\_iconfiguration = builder.Build();

}

static void PrintDoctor()

{

Console.Clear();

var doctorDAL = new DoctorDal(\_iconfiguration);

var listDoctorModel = doctorDAL.GetList();

Console.WriteLine("\n\n\n\n\n\t\t\t\t\t\t\t\t DOCTER\_ID || DOCTOR\_NAME ");

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

listDoctorModel.ForEach(item =>

{

Console.WriteLine("\t\t\t\t\t\t\t\t{0}\t\t{1}", item.DR\_ID, item.DR\_NAME);

});

try

{

Console.Write("\nEnter The Doctor\_ID of the Doctor you Want To Visit Either 1 or 2 : ");

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

Printfulllist(a);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

static void Printfulllist(int doctorid)

{

Console.Clear();

char re='y';

int b = 0;

string name="";

var flist = new DoctorDal(\_iconfiguration);

var flistModel = flist.GetLists(doctorid);

if (doctorid==1)

{

Console.WriteLine("\n\t xxxxxxxx Dr. Vikas Sharma xxxxxxxx");

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

}

else

{

Console.WriteLine("\n\t xxxxxxxx Dr. Manish Joshi xxxxxxxx");

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

}

string av = "";

int i = 1,total=0;

foreach (var f in flistModel)

{

total+=f.AVAILABLE;

}

if (total==6)

{

Console.WriteLine("All the slots are filled for this doctor...........");

return;

}

Console.WriteLine("\nSLOT\_ID\t|| AVAILABILITY\t|| DOCTER\_ID \t|| SLOTTIME\t \n");

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

flistModel.ForEach(item =>

{

if (item.AVAILABLE==0)

{

av="AVAILABLE";

}

else

{

av="---------";

}

Console.WriteLine("{0}\t|\t{1}\t|\t{2}\t|\t{3}\t ",i,av, item.DR\_ID, item.SLOTIME);

i++;

});

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

do

{

try

{

Console.Write("Enter The Slot Id to Book Your Appointment : ");

b = Convert.ToInt32(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

if (flistModel[b-1].AVAILABLE==1)

{

Console.WriteLine("Slot is already booked......");

continue;

}

else

{

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

name=Console.ReadLine();

Console.Clear();

Console.WriteLine("\n\n\n\n\n\t\t\t\t\t\t\t\tYour Appointment Has Been Booked........\n\n ");

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

Console.WriteLine("\t\t\t\t\t\t\t\tName : {0}", name);

Console.WriteLine("\t\t\t\t\t\t\t\tSlot Id : {0}",b);

Console.WriteLine("\t\t\t\t\t\t\t\tSlotime : {0}", flistModel[b-1].SLOTIME);

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

re='n';

}

}

while(re=='y' || re=='Y') ;

if (doctorid==2)

{

b=b+6;

}

Updateslot(b);

flist.Getpat(name, b, doctorid);

}

static public void Updateslot(int slotid)

{

var ulist = new DoctorDal(\_iconfiguration);

ulist.Getval(slotid);

}

static public void Cancelslot()

{

Console.Clear();

var ulist = new DoctorDal(\_iconfiguration);

int slotid=1, drid=1;

try

{

Console.WriteLine("\n\t\t\txxxxx SLOT CANCELLATION xxxxx");

Console.Write("Enter SlotID : ");

slotid = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter DoctorID : ");

drid = Convert.ToInt32(Console.ReadLine());

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

var ulistModel = ulist.GetLists(drid);

if (ulistModel[slotid-1].AVAILABLE==0)

{

Console.WriteLine("Slot is Already Empty.....");

return;

}

if (drid==2)

{

slotid=slotid+6;

}

ulist.Getcan(slotid);

Console.WriteLine("Slot Has Been canceled .......");

}

}

}

**DOCTORDAL.cs CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.Extensions.Configuration;

using DOCTOR\_SCHEDULING.MODEL;

using System.Data.SqlClient;

using System.Data;

namespace DOCTOR\_SCHEDULING.DAL

{

public class DoctorDal

{

private string \_connectionString;

public DoctorDal(IConfiguration iconfiguration)

{

\_connectionString = iconfiguration.GetConnectionString("Default");

}

public List<DoctorModel> GetList()

{

var listCountryModel = new List<DoctorModel>();

try

{

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand cmd = new SqlCommand("SP\_DOCTOR\_GET\_LIST", con);

cmd.CommandType = CommandType.StoredProcedure;

con.Open();

SqlDataReader rdr = cmd.ExecuteReader();

while (rdr.Read())

{

listCountryModel.Add(new DoctorModel

{

DR\_ID = Convert.ToInt32(rdr[0]),

DR\_NAME = Convert.ToString(rdr[1]),

});

}

}

}

catch (Exception ex)

{

throw ex;

}

return listCountryModel;

}

public List<fulllistModel> GetLists(int drid)

{

var fullModel = new List<fulllistModel>();

try

{

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand cmd = new SqlCommand("SP\_DOCTOR\_GET\_FULLLIST ", con);

cmd.CommandType = CommandType.StoredProcedure;

SqlParameter param1 = new SqlParameter

{

ParameterName="@drid",

SqlDbType = SqlDbType.Int,

Value = drid,

Direction = ParameterDirection.Input,

};

cmd.Parameters.Add(param1);

con.Open();

SqlDataReader rdr = cmd.ExecuteReader();

while (rdr.Read())

{

fullModel.Add(new fulllistModel

{

SLOTIME = Convert.ToString(rdr[3]),

SLOT\_ID = Convert.ToInt32(rdr[0]),

AVAILABLE = Convert.ToInt32(rdr[1]),

DR\_ID = Convert.ToInt32(rdr[2]),

DR\_NAME = Convert.ToString(rdr[4]),

});

}

}

}

catch (Exception ex)

{

throw ex;

}

return fullModel;

}

public void Getval(int show)

{

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand cmd = new SqlCommand("book\_appointment", con);

cmd.CommandType = CommandType.StoredProcedure;

SqlParameter param2 = new SqlParameter

{

ParameterName="@slotid",

SqlDbType = SqlDbType.Int,

Value =show,

Direction = ParameterDirection.Input,

};

cmd.Parameters.Add(param2);

con.Open();

cmd.ExecuteReader();

}

}

public void Getpat(string name,int slotid,int drid)

{

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand cmd = new SqlCommand("insert\_patient", con);

cmd.CommandType = CommandType.StoredProcedure;

SqlParameter param1 = new SqlParameter

{

ParameterName="@name",

SqlDbType = SqlDbType.VarChar,

Value=name,

Direction = ParameterDirection.Input,

};

SqlParameter param2 = new SqlParameter

{

ParameterName="@slot\_id",

SqlDbType = SqlDbType.Int,

Value =slotid,

Direction = ParameterDirection.Input,

};

SqlParameter param3 = new SqlParameter

{

ParameterName="@dr\_id",

SqlDbType = SqlDbType.Int,

Value =drid,

Direction = ParameterDirection.Input,

};

cmd.Parameters.Add(param1);

cmd.Parameters.Add(param2);

cmd.Parameters.Add(param3);

con.Open();

cmd.ExecuteReader();

}

}

public void Getcan(int show)

{

using (SqlConnection con = new SqlConnection(\_connectionString))

{

SqlCommand cmd = new SqlCommand("cancel\_appointment", con);

cmd.CommandType = CommandType.StoredProcedure;

SqlParameter param2 = new SqlParameter

{

ParameterName="@cslotid",

SqlDbType = SqlDbType.Int,

Value =show,

Direction = ParameterDirection.Input,

};

cmd.Parameters.Add(param2);

con.Open();

cmd.ExecuteReader();

}

}

}

}

**DoctorModel.cs CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DOCTOR\_SCHEDULING.MODEL

{

public class DoctorModel

{

public int DR\_ID { get; set; }

public string DR\_NAME { get; set; }

}

public class fulllistModel

{

public int SLOT\_ID { get; set; }

public int AVAILABLE { get; set; }

public int DR\_ID { get; set; }

public string SLOTIME { get; set; }

public string DR\_NAME { get; set; }

}

}

**Appsettings.json**

{

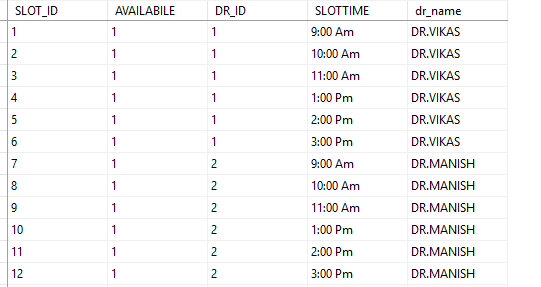
"ConnectionStrings": {

"Default": "Server=DESKTOP-3QD8K18\\SQLEXPRESS;Database=mydatabase;Persist Security Info = True; Integrated Security=SSPI"

}

}

**Tables Used**

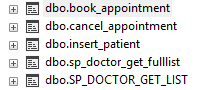


**Patient Table**

**HERE ALL PATIENT DATA WILL GET STORED**



**STORED PROCEDURED USE**



1. ALTER procedure [dbo].[book\_appointment](@slotid int) As begin

update DOCTOR set AVAILABILE =1 where SLOT\_ID=@slotid end

1. ALTER procedure [dbo].[cancel\_appointment](@cslotid int) As begin

update DOCTOR set AVAILABILE = 0 where SLOT\_ID=@cslotid end

1. ALTER procedure [dbo].[cancel\_appointment](@cslotid int) As begin

update DOCTOR set AVAILABILE = 0 where SLOT\_ID=@cslotid end

1. ALTER procedure [dbo].[sp\_doctor\_get\_fulllist]

(@drid varchar(1)) AS

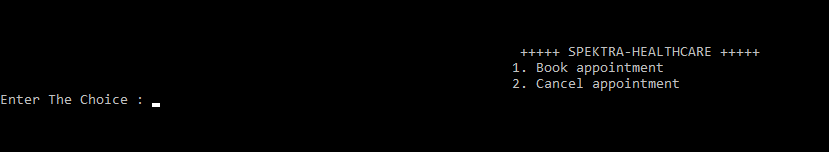
begin

select \* from DOCTOR where dr\_id=@drid

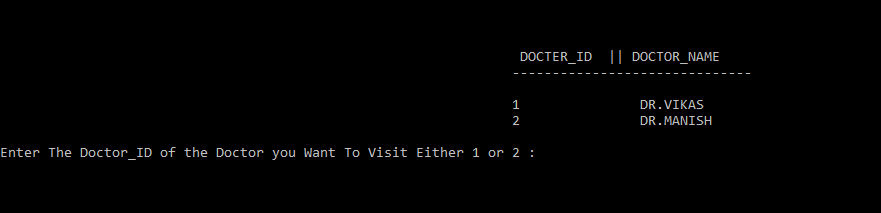
end

5. ALTER procedure [dbo].[SP\_DOCTOR\_GET\_LIST] As begin select dr\_id, dr\_name from doctor group by dr\_name , dr\_id order by dr\_id end

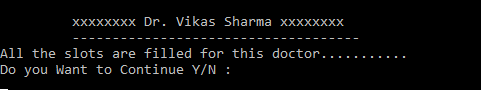
**OUTPUT**



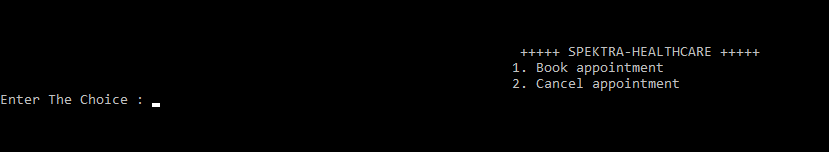
**AFTER ENTERING 1**

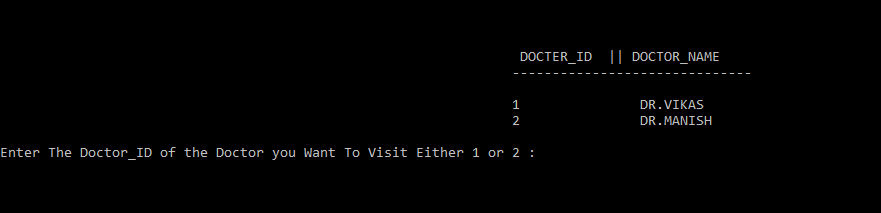


**AFTER ENTERING 1**

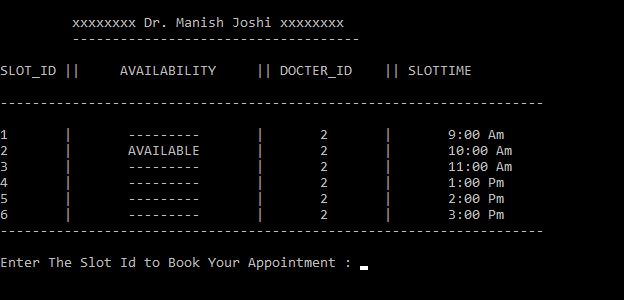


**As all slots has been filled of Doctor Vikas it shows this message, lets check for second doctor….**

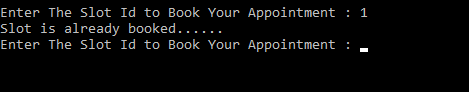




**After entering 2**



**If you enter 1 it will show**



**If you enter 2**

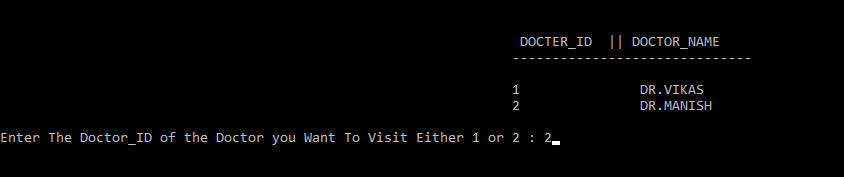


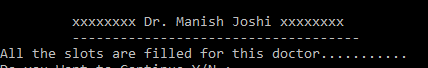


**Patient Table Updated**



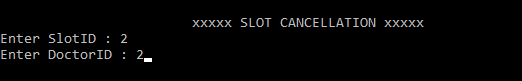
**Lets check the slot of Doctor Manish**

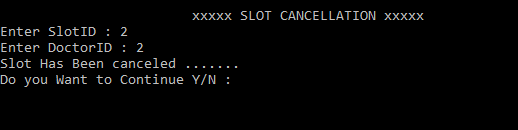




**As it was the last slot that we have booked so no slots are available**

**Lets cancel the slots that we have booked..**





**So the slot has been canceled lets check the slot again..**



**Its available again…**

**Thank you…**