**Hospital Management System**



Session: 2021 – 2024

**Submitted by:**

Muhammad Nazir (2021-CS-08)

**Supervised by:**

Sir Awais Hassan & Mam Maida

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**Table of Content**

**Topics**  **Page No**

1. Class Diagram………………………………………………….......03
2. Sequence Diagram...………….……………………………………04
3. Wire Frames ...………….…………………………………….........04

3.1 Main Menu ……………………………………………………................ 04

3.2 SignUP page …………………………………………………………….. 05

3.3 Login page ………………………………………………………………. 05

3.4 Admin Menu…………………………………………………………......... 06

3.5 Add Patient……………….. ……………………………………………… 06

3.6 View Patient in Sorted Order …………………………………………….. 07

3.7 Delete Patient Data……….………………………………………………... 07

3.8 Edit Patient Data…………………………………………………….…….. 08

3.9 Add Birth Data…………………………………………………………..... 08

3.10 View Birth Data …………………………………………………………. 09

3.11 Delete Birth Data ………………………………………..………………. 09

3.12 Add Death Data ………………………………….…………………….. 10

3.13 View Death Data ………………………………….…………………… 10

3.14 Delete Death Data ……………………………………………………… 11

3.15 Room Information ……………………………………………..………. 11

3.16 Patient Menu ………………………………………………..………….. 12

3.17 Add Personal Data ………………………………………………………. 12

3.18 View Individual personal Data ………………………………...………... 13

3.19 Add Medical Data ……………………………………………………… 13

3.20 View Individual Medical Data ……………………………………………14

3.21 View Doctor Data …………………………………………………………14

3.22 Recommendation …..………………………………………..………….. 15

3.23 Sent Appointment request ..………………………………………………. 15

3.24 Give Feedback…………………………………………………...………... 15

3.25 Receptionist Menu ……………………………………………………… 16

3.26 View all Patients Personal Data ……………………………………………16

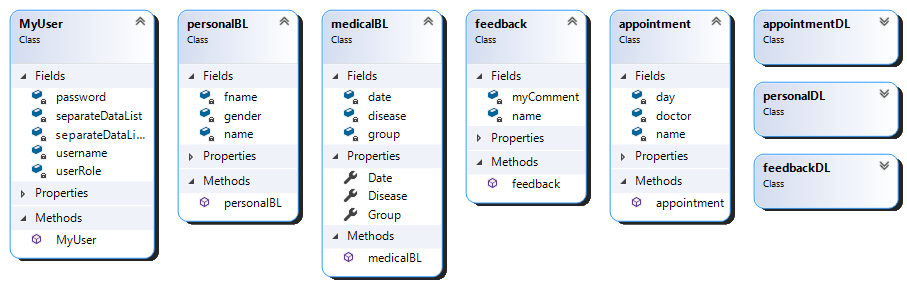
3.27 View all Patients Medical Data ……………………………………………17

3.28 View Feedback …………………………………………………….………17

3.29 View Appointment………………………………………………….………17

1. Complete Code ……………………..……………………………. 18

**Class Diagram**



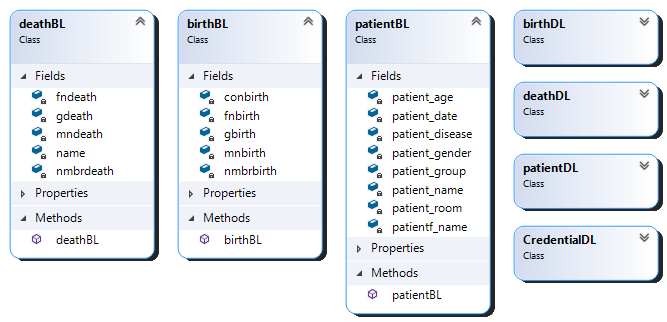
**∞**

**∞**

**∞**

**∞**

1

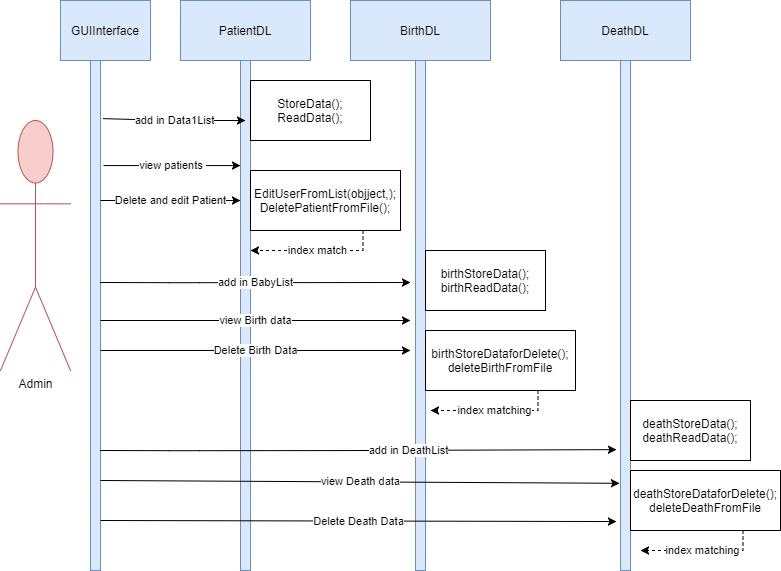


**∞**

**∞**

**∞**

**Sequence Diagram**

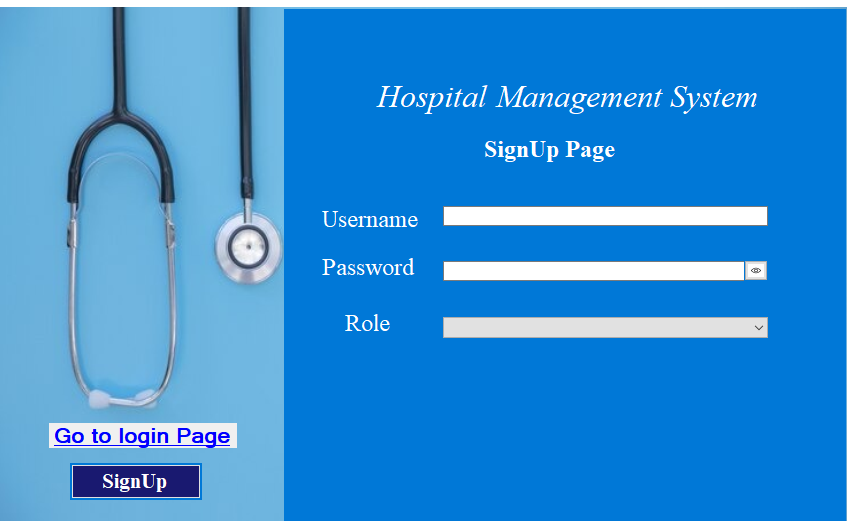
****

**Wireframe**

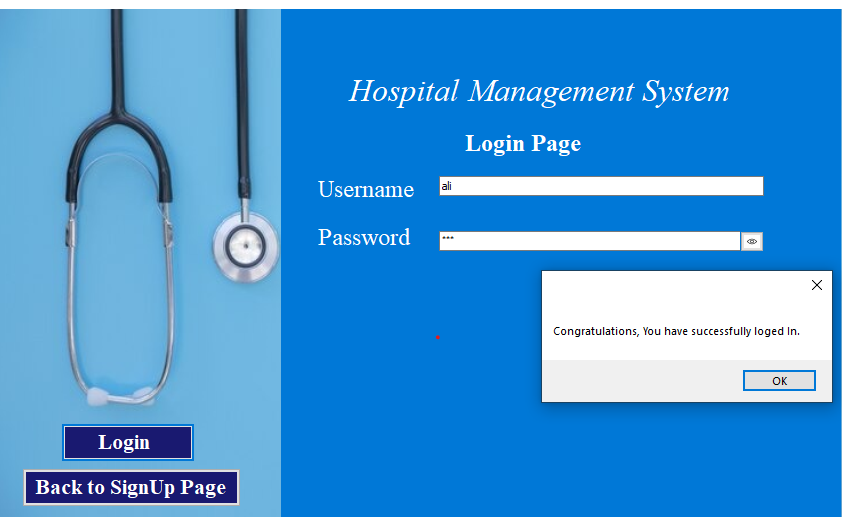
**Main Page**

****

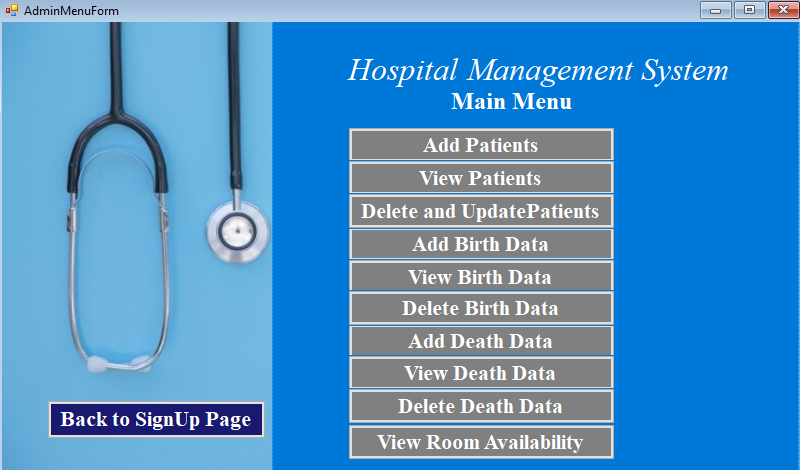
**SignUP Page**



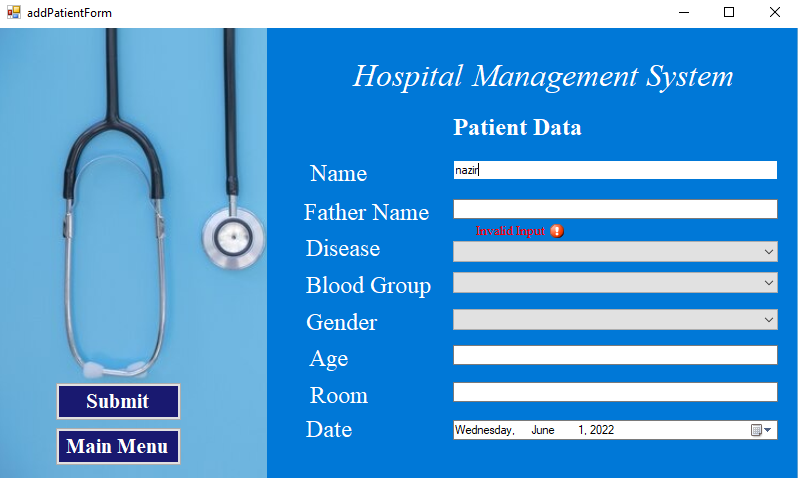
**Login Page**



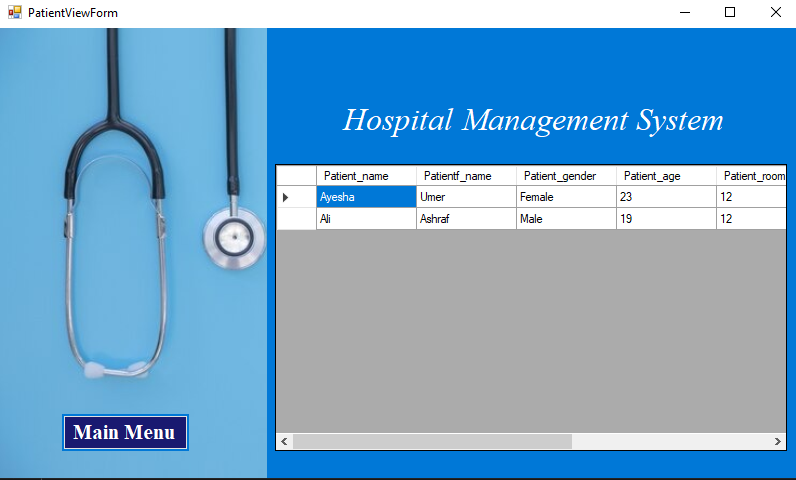
**Admin Menu**



**Add Patient**



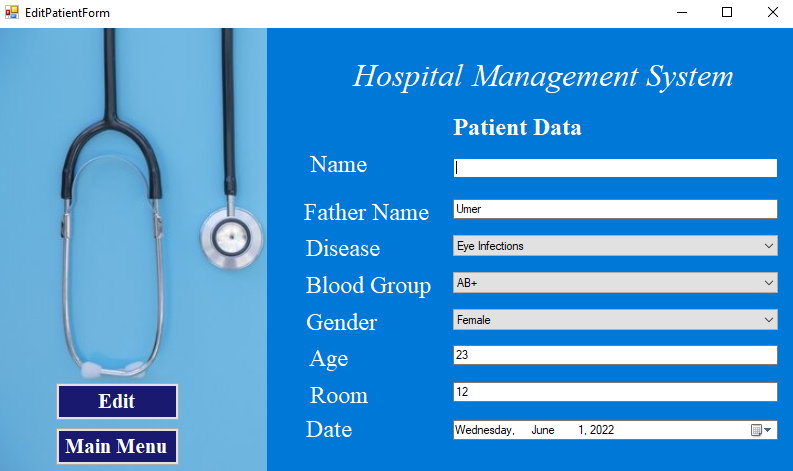
**View Patient in Sorted Order**



**Delete Patient Data**



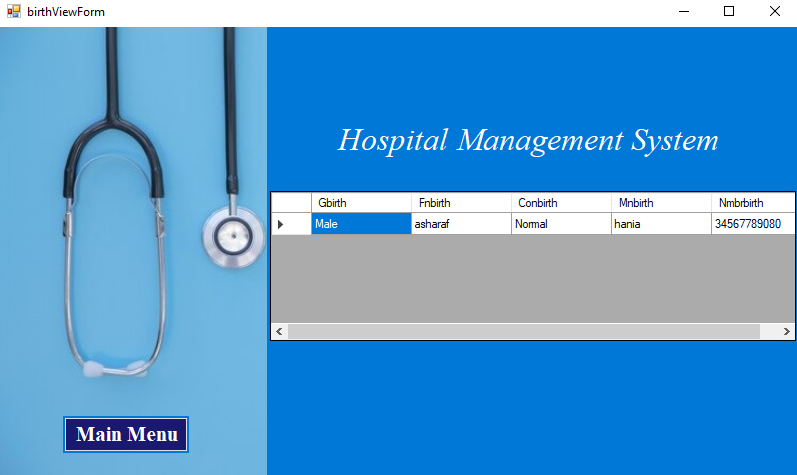
**Edit patient Data**



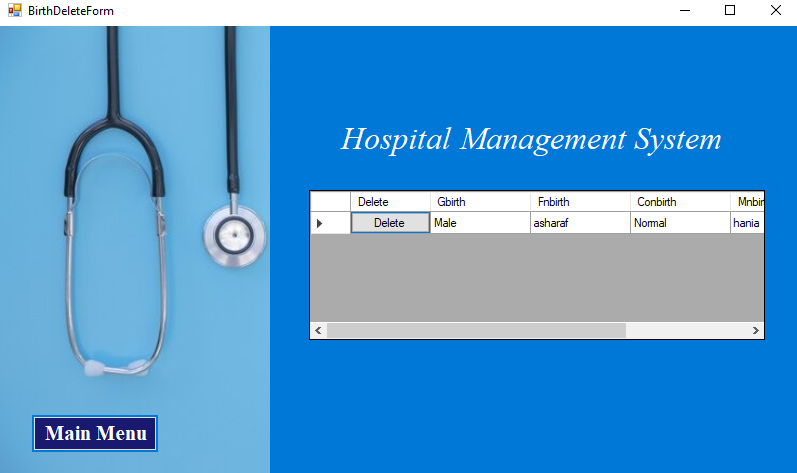
**Add Birth Data**



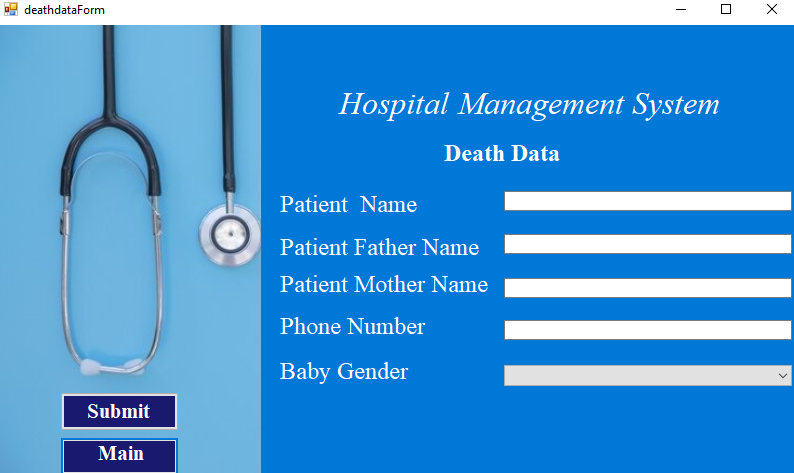
**View Birth Data**



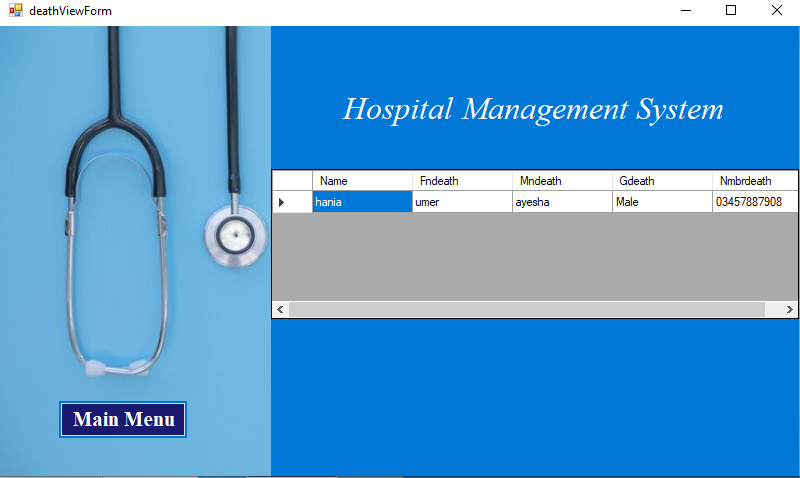
**Delete Birth Data**

****

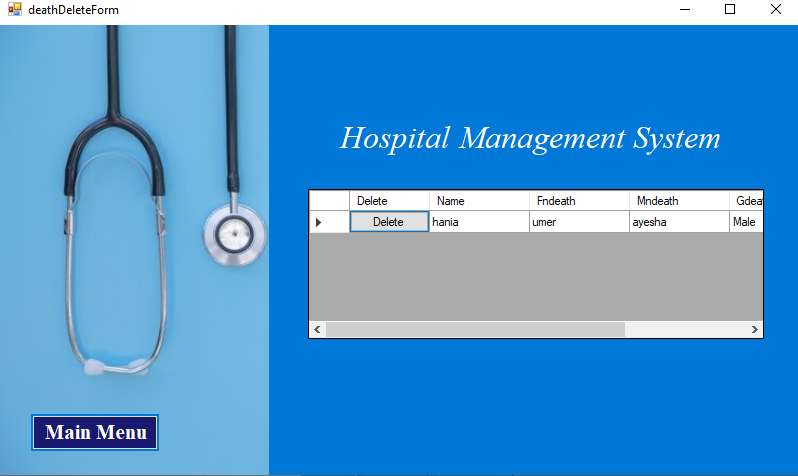
**Death Data**

****

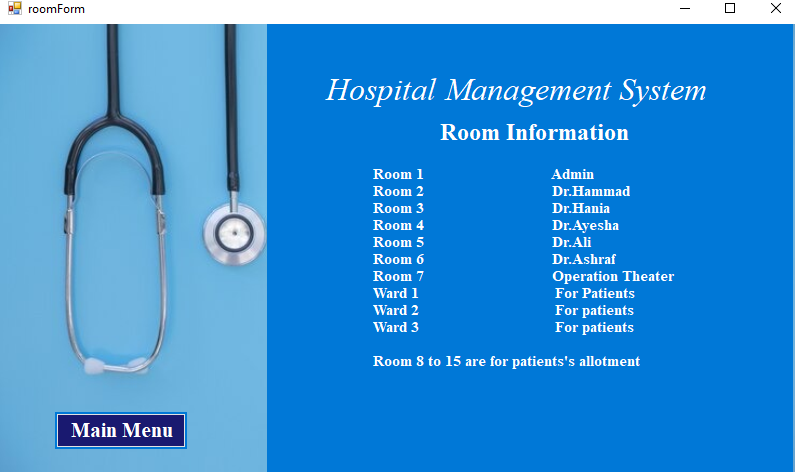
**Death View Data**

****

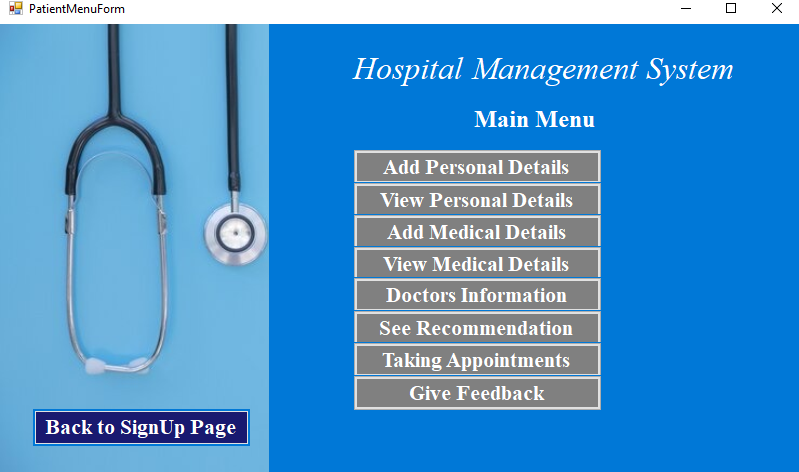
**Delete Death Data**

****

**Room Information**

****

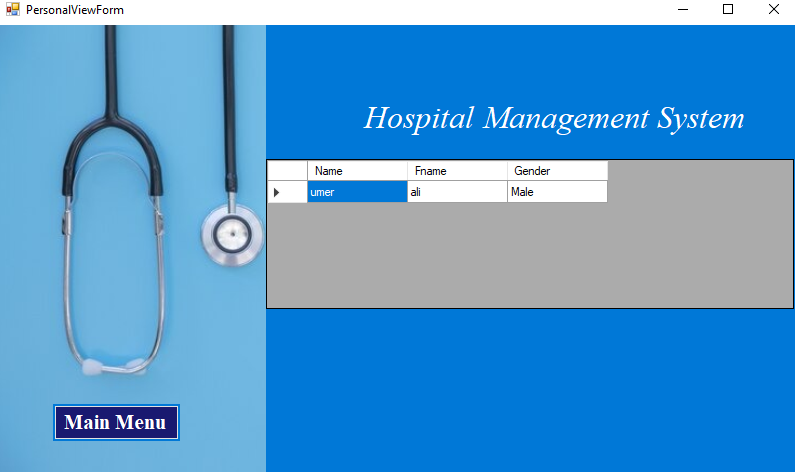
**Patient Menu**

****

**Add Personal Data**

****

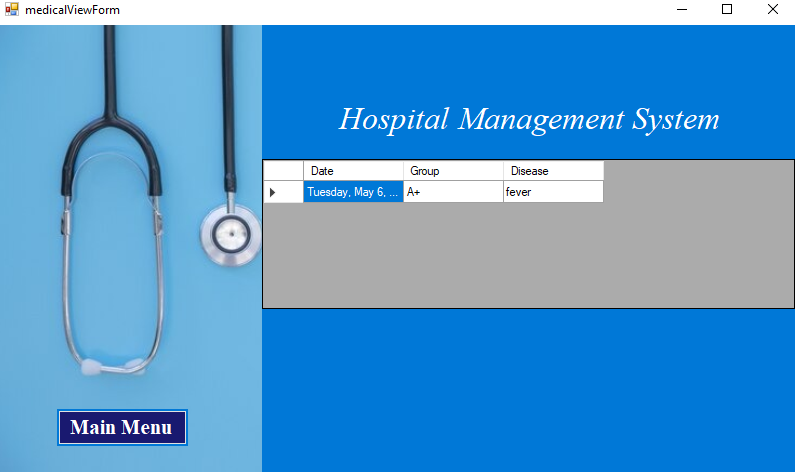
**View individual Personal Data**

****

**Add Medical Data**

****

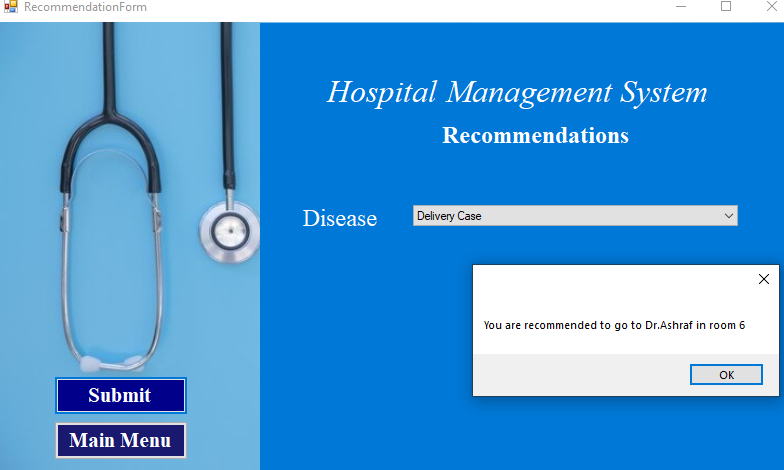
**View Medical Data**

****

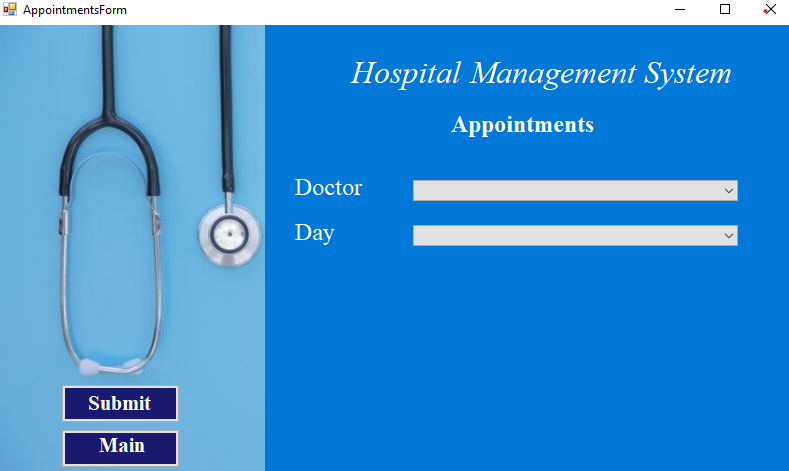
**View Doctors Data**

****

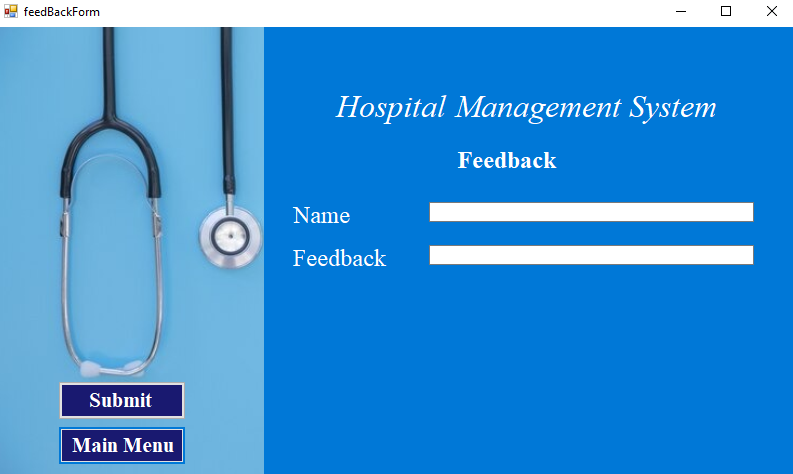
**Recommendations**

****

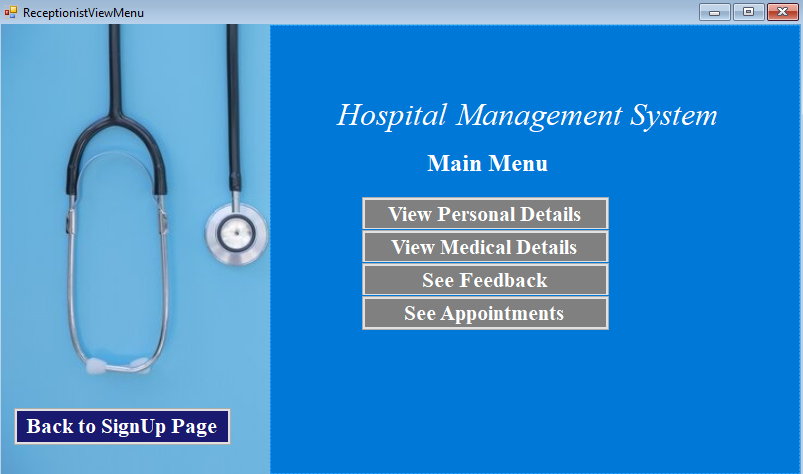
**Sent Appointments Request**

****

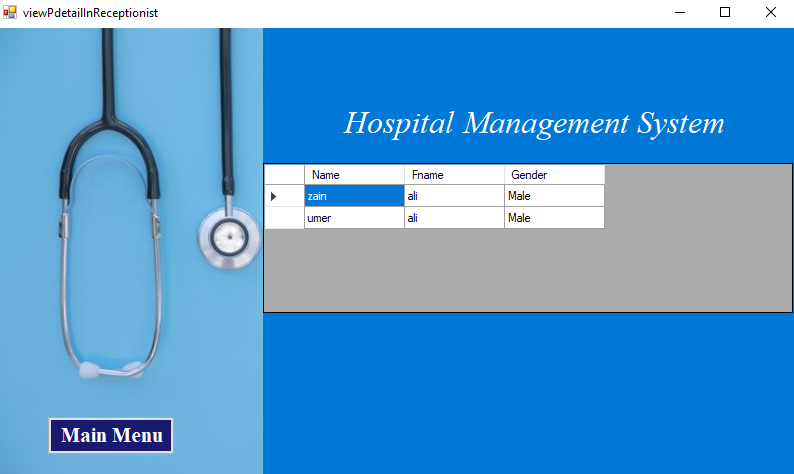
**Give Feedback**

****

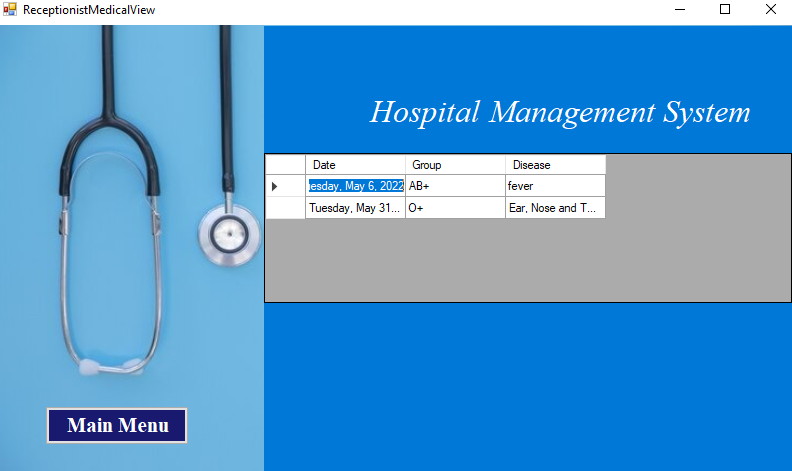
**Receptionist Menu**

****

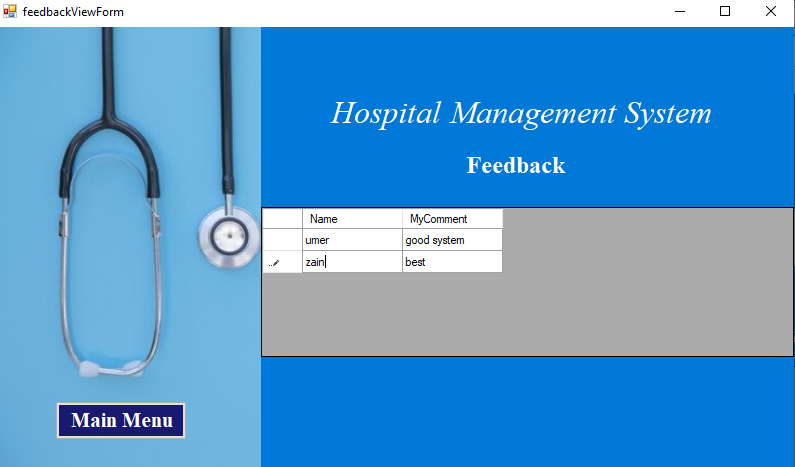
**View All Patients Personal Data**



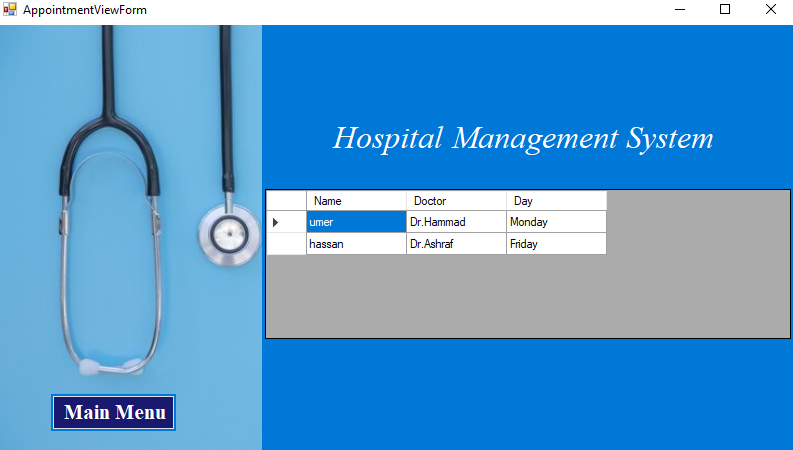
**View All Patients Medical Data**

****

**View Feedbacks**

****

**View Appointments Request**



**Complete Code of Project**

**CredentialBL:**

class MyUser

{

private string username;

private string password;

private string userRole;

private List<personalBL> separateDataList = new List<personalBL>();

private List<medicalBL> separateDataListForMedical = new List<medicalBL>();

public MyUser(string username, string password, string userRole)

{

this.Username = username;

this.Password = password;

this.UserRole = userRole;

}

public string Username { get => username; set => username = value; }

public string Password { get => password; set => password = value; }

public string UserRole { get => userRole; set => userRole = value; }

internal List<medicalBL> SeparateDataListForMedical { get => separateDataListForMedical; set => separateDataListForMedical = value; }

internal List<personalBL> SeparateDataList { get => separateDataList; set => separateDataList = value; }

}

**PatientBL:**

**public class patientBL**

**{**

**//data members......................**

**private string patient\_name;**

**private string patientf\_name;**

**private string patient\_gender;**

**private float patient\_age;**

**private int patient\_room;**

**private string patient\_date;**

**private string patient\_group;**

**private string patient\_disease;**

**public patientBL(string patient\_name, string patientf\_name, string patient\_gender, float patient\_age,**

**int patient\_room, string patient\_date, string patient\_group, string patient\_disease)**

**{**

**this.patient\_name = patient\_name;**

**this.patientf\_name = patientf\_name;**

**this.patient\_gender = patient\_gender;**

**this.patient\_age = patient\_age;**

**this.patient\_room = patient\_room;**

**this.patient\_date = patient\_date;**

**this.patient\_group = patient\_group;**

**this.patient\_disease = patient\_disease;**

**}**

**public string Patient\_name { get => patient\_name; set => patient\_name = value; }**

**public string Patientf\_name { get => patientf\_name; set => patientf\_name = value; }**

**public string Patient\_gender { get => patient\_gender; set => patient\_gender = value; }**

**public float Patient\_age { get => patient\_age; set => patient\_age = value; }**

**public int Patient\_room { get => patient\_room; set => patient\_room = value; }**

**public string Patient\_date { get => patient\_date; set => patient\_date = value; }**

**public string Patient\_group { get => patient\_group; set => patient\_group = value; }**

**public string Patient\_disease { get => patient\_disease; set => patient\_disease = value; }**

**}**

**BirthBL:**

class birthBL

{

private string gbirth;

private string fnbirth;

private string conbirth;

private string mnbirth;

private string nmbrbirth;

public birthBL(string gbirth, string fnbirth, string conbirth, string mnbirth, string nmbrbirth)

{

this.gbirth = gbirth;

this.fnbirth = fnbirth;

this.conbirth = conbirth;

this.mnbirth = mnbirth;

this.nmbrbirth = nmbrbirth;

}

public string Gbirth { get => gbirth; set => gbirth = value; }

public string Fnbirth { get => fnbirth; set => fnbirth = value; }

public string Conbirth { get => conbirth; set => conbirth = value; }

public string Mnbirth { get => mnbirth; set => mnbirth = value; }

public string Nmbrbirth { get => nmbrbirth; set => nmbrbirth = value; }

}

**DeathBL:**

class deathBL

{

private string name;

private string fndeath;

private string mndeath;

private string gdeath;

private string nmbrdeath;

public deathBL(string name, string fndeath, string mndeath, string gdeath, string nmbrdeath)

{

this.name = name;

this.fndeath = fndeath;

this.mndeath = mndeath;

this.gdeath = gdeath;

this.nmbrdeath = nmbrdeath;

}

public string Name { get => name; set => name = value; }

public string Fndeath { get => fndeath; set => fndeath = value; }

public string Mndeath { get => mndeath; set => mndeath = value; }

public string Gdeath { get => gdeath; set => gdeath = value; }

public string Nmbrdeath { get => nmbrdeath; set => nmbrdeath = value; }

}

**PersonalBL:**

class personalBL

{

private string name;

private string fname;

private string gender;

public personalBL(string name, string fname, string gender)

{

this.name = name;

this.fname = fname;

this.gender = gender;

}

public string Name { get => name; set => name = value; }

public string Fname { get => fname; set => fname = value; }

public string Gender { get => gender; set => gender = value; }

}

**MedicalBL:**

class medicalBL

{

private string date;

private string group;

private string disease;

public medicalBL(string date, string group, string disease)

{

this.date = date;

this.group = group;

this.disease = disease;

}

public string Date { get => date; set => date = value; }

public string Group { get => group; set => group = value; }

public string Disease { get => disease; set => disease = value; }

}

**AppointmentBL:**

class appointment

{

private string name;

private string doctor;

private string day;

public appointment(string name, string doctor, string day)

{

this.name = name;

this.doctor = doctor;

this.day = day;

}

public string Name { get => name; set => name = value; }

public string Doctor { get => doctor; set => doctor = value; }

public string Day { get => day; set => day = value; }

}

**FeedbackBL:**

class feedback

{

private string name;

private string myComment;

public feedback(string name, string comment)

{

this.name = name;

MyComment = comment;

}

public string Name { get => name; set => name = value; }

public string MyComment { get => myComment; set => myComment = value; }

}

**CredentialDL:**

class CredentialDL

{

public static int idx = 0;

private static List<MyUser> signUpdata = new List<MyUser>();

internal static List<MyUser> SignUpdata { get => signUpdata; set => signUpdata = value; }

public static void addSignUpIntoList(MyUser credential)

{

SignUpdata.Add(credential);

}

public static string SignIn(string username, string password)

{

for (int i= 0 ; i<SignUpdata.Count; i++)

{

if (username == SignUpdata[i].Username && password == SignUpdata[i].Password)

{

idx = i;

return SignUpdata[i].UserRole;

}

}

return "NA";

}

public static void storeData(MyUser data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\LoginData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine(data.Username + "," + data.Password + "," + data.UserRole);

file.Flush();

file.Close();

}

public static bool readData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\LoginData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string username = splittedRecord[0];

string password = splittedRecord[1];

string userRole = splittedRecord[2];

MyUser data = new MyUser(username, password, userRole);

SignUpdata.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**PatientDL:**

class patientDL

{

private static List<patientBL> data1 = new List<patientBL>();

internal static List<patientBL> Data1 { get => data1; set => data1 = value; }

public static void storeData(patientBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\PatientData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine(data.Patient\_name + "?" + data.Patientf\_name + "?" + data.Patient\_gender + "?" +

data.Patient\_age + "?" + data.Patient\_room + "?" + data.Patient\_date + "?" + data.Patient\_group

+ "?" + data.Patient\_disease);

file.Flush();

file.Close();

}

public static void storeDataForDelete()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\PatientData.txt";

StreamWriter file = new StreamWriter(path1);

foreach(patientBL data in Data1)

{

file.WriteLine(data.Patient\_name + "?" + data.Patientf\_name + "?" + data.Patient\_gender + "?" +

data.Patient\_age + "?" + data.Patient\_room + "?" + data.Patient\_date + "?" + data.Patient\_group

+ "?" + data.Patient\_disease);

}

file.Flush();

file.Close();

}

public static void EditUserFromList(patientBL previous, patientBL updated)

{

foreach (patientBL patient in Data1)

{

if (patient.Patient\_name == previous.Patient\_name && patient.Patientf\_name == previous.Patientf\_name

&& patient.Patient\_gender == previous.Patient\_gender && patient.Patient\_age == previous.Patient\_age

&& patient.Patient\_room == previous.Patient\_room && patient.Patient\_date == previous.Patient\_date

&& patient.Patient\_group == previous.Patient\_group && patient.Patient\_disease == previous.Patient\_disease)

{

patient.Patient\_name = updated.Patient\_name;

patient.Patientf\_name = updated.Patientf\_name;

patient.Patient\_gender = updated.Patient\_gender;

patient.Patient\_age = updated.Patient\_age;

patient.Patient\_room = updated.Patient\_room;

patient.Patient\_date = updated.Patient\_date;

patient.Patient\_group = updated.Patient\_group;

patient.Patient\_disease = updated.Patient\_disease;

}

}

}

public static void deletePatientsFromFile( patientBL data)

{

for(int i=0; i< Data1.Count; i++)

{

if(Data1[i].Patient\_name == data.Patient\_name && Data1[i].Patientf\_name == data.Patientf\_name &&

Data1[i].Patient\_gender == data.Patient\_gender && Data1[i].Patient\_age == data.Patient\_age &&

Data1[i].Patient\_room == data.Patient\_room && Data1[i].Patient\_date == data.Patient\_date &&

Data1[i].Patient\_group == data.Patient\_group && Data1[i].Patient\_disease == data.Patient\_disease)

{

Data1.RemoveAt(i);

}

}

}

public static void sorting() {

Data1 = Data1.OrderByDescending(o => o.Patient\_date).ToList();

}

public static bool readData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\PatientData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split('?');

string name = splittedRecord[0];

string fname = splittedRecord[1];

string gender = splittedRecord[2];

float age = float.Parse(splittedRecord[3]);

int room = int.Parse(splittedRecord[4]);

string date = splittedRecord[5];

string group = splittedRecord[6];

string disease = splittedRecord[7];

patientBL data = new patientBL(name , fname, gender , age , room, date , group , disease);

Data1.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**BirthDL:**

class birthDL

{

private static List<birthBL> baby = new List<birthBL>();

internal static List<birthBL> Baby { get => baby; set => baby = value; }

public static void birthStoreData(birthBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\BirthData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine(data.Gbirth + "," + data.Fnbirth + "," + data.Conbirth + "," + data.Mnbirth + "," +data.Nmbrbirth );

file.Flush();

file.Close();

}

public static void birthStoreDataForDelete()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\BirthData.txt";

StreamWriter file = new StreamWriter(path1);

foreach (birthBL data in Baby)

{

file.WriteLine(data.Gbirth + "," + data.Fnbirth + "," + data.Conbirth + "," + data.Mnbirth + "," + data.Nmbrbirth);

}

file.Flush();

file.Close();

}

public static void deleteBirthFromFile(birthBL data)

{

for (int i = 0; i < Baby.Count; i++)

{

if (Baby[i].Fnbirth == data.Fnbirth && Baby[i].Gbirth == data.Gbirth && Baby[i].Mnbirth == data.Mnbirth &&

Baby[i].Conbirth == data.Conbirth && Baby[i].Nmbrbirth == data.Nmbrbirth )

{

Baby.RemoveAt(i);

}

}

}

public static bool birthReadData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\BirthData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string gender = splittedRecord[0];

string fname = splittedRecord[1];

string condition = splittedRecord[2];

string mname = splittedRecord[3];

string number = splittedRecord[4];

birthBL data = new birthBL( gender, fname, condition, mname ,number);

Baby.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**DeathDL:**

class deathDL

{

private static List<deathBL> death = new List<deathBL>();

internal static List<deathBL> Death { get => death; set => death = value; }

public static void deathStoreData(deathBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\DeathData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine(data.Name + "," + data.Fndeath + "," + data.Mndeath + "," + data.Gdeath + "," + data.Nmbrdeath);

file.Flush();

file.Close();

}

public static void deathStoreDataFromFile()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\DeathData.txt";

StreamWriter file = new StreamWriter(path1, true);

foreach (deathBL data in Death)

{

file.WriteLine(data.Name + "," + data.Fndeath + "," + data.Mndeath + "," + data.Gdeath + "," + data.Nmbrdeath);

}

file.Flush();

file.Close();

}

public static void deleteDeathFromFile(deathBL data)

{

for (int i = 0; i < Death.Count; i++)

{

if (Death[i].Fndeath == data.Fndeath && Death[i].Gdeath == data.Gdeath && Death[i].Mndeath == data.Mndeath &&

Death[i].Name == data.Name && Death[i].Nmbrdeath == data.Nmbrdeath)

{

Death.RemoveAt(i);

}

}

}

public static bool deathReadData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\DeathData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string name = splittedRecord[0];

string fname = splittedRecord[1];

string mname = splittedRecord[2];

string condition = splittedRecord[3];

string number = splittedRecord[4];

deathBL data = new deathBL(name, fname, mname, condition, number);

Death.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**PersonalDL:**

class personalDL

{

private static List<personalBL> p = new List<personalBL>();

internal static List<personalBL> P { get => p; set => p = value; }

public static void addIntoList(personalBL data , int idx)

{

CredentialDL.SignUpdata[idx].SeparateDataList.Add(data);

}

public static void PersonalStoreData(personalBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\PersonalData.txt";

StreamWriter file = new StreamWriter(path1,true);

file.WriteLine(CredentialDL.idx + "," + data.Name + "," + data.Fname + "," + data.Gender );

file.Flush();

file.Close();

}

public static bool PersonalReadData(int idx)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\PersonalData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

int TempIndex = 0;

if (splittedRecord[0] != "")

{

TempIndex = int.Parse(splittedRecord[0]);

}

if (TempIndex == CredentialDL.idx)

{

string name = splittedRecord[1];

string fname = splittedRecord[2];

string gender = splittedRecord[3];

personalBL data = new personalBL(name, fname, gender);

CredentialDL.SignUpdata[idx].SeparateDataList.Add(data);

}

}

file.Close();

return true;

}

else

{

return false;

}

}

public static void AllPersonalStoreData(personalBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\AllPersonalData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine( data.Name + "," + data.Fname + "," + data.Gender);

file.Flush();

file.Close();

}

public static bool AllPersonalReadData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\AllPersonalData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string name = splittedRecord[0];

string fname = splittedRecord[1];

string gender = splittedRecord[2];

personalBL data = new personalBL(name, fname, gender);

personalDL.P.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**MedicalDL:**

class medicalDL

{

private static List<medicalBL> m = new List<medicalBL>();

internal static List<medicalBL> M { get => m; set => m = value; }

public static void addIntoMedicalList(medicalBL data, int idx)

{

CredentialDL.SignUpdata[idx].SeparateDataListForMedical.Add(data);

}

public static void medicalStoreData(medicalBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\MedicalData.txt";

StreamWriter file = new StreamWriter(path1 , true);

file.WriteLine( CredentialDL.idx +"?" +data.Date + "?" + data.Group + "?" + data.Disease );

file.Flush();

file.Close();

}

public static void AllmedicalStoreData(medicalBL data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\AllMedicalData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine( data.Date + "?" + data.Group + "?" + data.Disease);

file.Flush();

file.Close();

}

public static bool AllmedicalReadData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\AllMedicalData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split('?');

string date = splittedRecord[0];

string group = splittedRecord[1];

string disease = splittedRecord[2];

medicalBL data = new medicalBL(date, group, disease);

medicalDL.M.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

public static bool medicalReadData(int idx)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\MedicalData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split('?');

int TempIndex = int.Parse(splittedRecord[0]);

if (TempIndex == CredentialDL.idx)

{

string date = splittedRecord[1];

string group = splittedRecord[2];

string disease = splittedRecord[3];

medicalBL data = new medicalBL(date, group, disease);

CredentialDL.SignUpdata[idx].SeparateDataListForMedical.Add(data);

}

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**AppointmentDL:**

class appointmentDL

{

private static List<appointment> appointmentList = new List<appointment>();

internal static List<appointment> AppointmentList { get => appointmentList; set => appointmentList = value; }

public static void storeData(string name,appointment data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\AppointmentData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine(name+","+data.Doctor + "," + data.Day);

file.Flush();

file.Close();

}

public static bool readData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\AppointmentData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string name = splittedRecord[0];

string doctor = splittedRecord[1];

string day = splittedRecord[2];

appointment data = new appointment(name,doctor, day);

appointmentList.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**FeedbackDL:**

class feedbackDL

{

private static List<feedback> commentList = new List<feedback>();

internal static List<feedback> CommentList { get => commentList; set => commentList = value; }

public static void FeedbackStoreData(feedback data)

{

string path1 = "D:\\GUIHMS\\signUp\\files\\FeedbackData.txt";

StreamWriter file = new StreamWriter(path1, true);

file.WriteLine(data.Name + "," + data.MyComment );

file.Flush();

file.Close();

}

public static bool FeedbackReadData()

{

string path1 = "D:\\GUIHMS\\signUp\\files\\FeedbackData.txt";

StreamReader file = new StreamReader(path1);

string record;

if (File.Exists(path1))

{

while ((record = file.ReadLine()) != null)

{

string[] splittedRecord = record.Split(',');

string name = splittedRecord[0];

string comment = splittedRecord[1];

feedback data = new feedback(name, comment);

CommentList.Add(data);

}

file.Close();

return true;

}

else

{

return false;

}

}

}

**AdminMenuForm:**

public partial class AdminMenuForm : Form

{

public AdminMenuForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

Form1 obj = new Form1();

obj.Show();

}

private void Abtn7\_Click(object sender, EventArgs e)

{

this.Hide();

roomForm obj = new roomForm();

obj.Show();

}

private void Abtn1\_Click\_1(object sender, EventArgs e)

{

this.Hide();

addPatientForm obj = new addPatientForm();

obj.Show();

}

private void Abtn2\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PatientViewForm obj = new PatientViewForm();

obj.Show();

}

private void Abtn3\_Click\_1(object sender, EventArgs e)

{

this.Hide();

birthDataForm obj = new birthDataForm();

obj.Show();

}

private void Abtn4\_Click\_1(object sender, EventArgs e)

{

this.Hide();

birthViewForm obj = new birthViewForm();

obj.Show();

}

private void Abtn5\_Click\_1(object sender, EventArgs e)

{

this.Hide();

deathdataForm obj = new deathdataForm();

obj.Show();

}

private void Abtn6\_Click\_1(object sender, EventArgs e)

{

this.Hide();

deathViewForm obj = new deathViewForm();

obj.Show();

}

private void Abtn7\_Click\_1(object sender, EventArgs e)

{

this.Hide();

roomForm obj = new roomForm();

obj.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

DeletePatientForm obj = new DeletePatientForm();

obj.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Hide();

BirthDeleteForm obj = new BirthDeleteForm();

obj.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Hide();

deathDeleteForm obj = new deathDeleteForm();

obj.Show();

}

}

**AddPatient Form:**

public partial class addPatientForm : Form

{

public addPatientForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void submitBtn\_Click(object sender, EventArgs e)

{

string patient\_name, patientf\_name, patient\_gender, patient\_date, patient\_group, patient\_disease;

float patient\_age;

int patient\_room;

if (nameTextBox.Text == "" || fNameTextBox.Text == "" || comboBox1.Text == "" || ageTextBox.Text == "" ||

dateTimePicker1.Text == "" || comboBox2.Text == "" || comboBox3.Text == "" || roomTextBox.Text == "")

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else if ( ageTextBox.ForeColor == Color.Black && roomTextBox.ForeColor == Color.Black)

{

patient\_name = nameTextBox.Text;

patientf\_name = fNameTextBox.Text;

patient\_group = comboBox1.Text;

patient\_gender = comboBox2.Text;

patient\_date = dateTimePicker1.Text;

patient\_disease = comboBox3.Text;

patient\_age = float.Parse(ageTextBox.Text);

patient\_room = int.Parse(roomTextBox.Text);

patientBL data = new patientBL(patient\_name, patientf\_name, patient\_gender, patient\_age, patient\_room,

patient\_date, patient\_group, patient\_disease);

patientDL.Data1.Add(data);

patientDL.storeData(data);

MessageBox.Show("All credentials are Submitted.");

nameTextBox.Clear();

fNameTextBox.Clear();

comboBox1.Text = "";

comboBox2.Text = "";

dateTimePicker1.Text = "";

comboBox3.Text = "";

ageTextBox.Clear();

roomTextBox.Clear();

}

else

{

MessageBox.Show("Not Submitted.Please fill correct input.");

}

}

private void ageTextBox\_TextChanged(object sender, EventArgs e)

{

Decimal a;

bool isAValid = Decimal.TryParse(ageTextBox.Text, out a);

if (isAValid)

{

ageTextBox.ForeColor = Color.Black;

}

else

ageTextBox.ForeColor = Color.Red;

}

private void ageTextBox\_MaskInputRejected\_1(object sender, MaskInputRejectedEventArgs e)

{

}

private void roomTextBox\_TextChanged(object sender, EventArgs e)

{

Decimal a;

bool isAValid = Decimal.TryParse(roomTextBox.Text, out a);

if (isAValid)

{

roomTextBox.ForeColor = Color.Black;

}

else

roomTextBox.ForeColor = Color.Red;

}

private void nameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label1.Visible = true;

errorProvider1.SetError(label1, "Invalid Input");

label1.Text = "Invalid Input";

}

else

{

label1.Visible = false;

errorProvider1.SetError(label1, "");

label1.Text = "";

}

}

private void fNameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label2.Visible = true;

errorProvider1.SetError(label2, "Invalid Input");

label2.Text = "Invalid Input";

}

else

{

label2.Visible = false;

errorProvider1.SetError(label2, "");

label2.Text = "";

}

}

}

**ViewPatient Form:**

public partial class PatientViewForm : Form

{

public PatientViewForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void PatientViewForm\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = patientDL.Data1;

patientDL.sorting();

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

dataGridView1.DataSource = patientDL.Data1;

patientDL.sorting();

}

}

}

**DeleteEditPatientForm:**

public partial class DeletePatientForm : Form

{

public DeletePatientForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

dataGridView1.DataSource = patientDL.Data1;

}

private void Databind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = patientDL.Data1;

dataGridView1.Refresh();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

patientBL data = (patientBL)dataGridView1.CurrentRow.DataBoundItem;

if(dataGridView1.Columns["Delete"].Index == e.ColumnIndex)

{

patientDL.deletePatientsFromFile(data);

patientDL.storeDataForDelete();

Databind();

}

else if (dataGridView1.Columns["Edit"].Index == e.ColumnIndex)

{

EditPatientForm myform = new EditPatientForm(data);

myform.ShowDialog();

patientDL.storeDataForDelete();

Databind();

}

}

}

**BirthDataForm:**

public partial class birthDataForm : Form

{

public birthDataForm()

{

InitializeComponent();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void submitBtn\_Click(object sender, EventArgs e)

{

string gbirth, fnbirth, conbirth, mnbirth, nmbrbirth;

if (nameTextBox.Text == "" || mNameTextBox.Text == "" || comboBox1.Text == "" ||

phoneTextBox.Text == "" || comboBox2.Text == "")

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else

{

gbirth = comboBox2.Text;

fnbirth = nameTextBox.Text;

mnbirth = mNameTextBox.Text;

conbirth = comboBox1.Text;

nmbrbirth = phoneTextBox.Text;

birthBL birthData = new birthBL(gbirth, fnbirth, conbirth, mnbirth, nmbrbirth);

MessageBox.Show("All credentials are Submitted.");

birthDL.Baby.Add(birthData);

birthDL.birthStoreData(birthData);

comboBox2.Text = "";

nameTextBox.Clear();

mNameTextBox.Clear();

comboBox1.Text = "";

phoneTextBox.Clear();

}

}

private void phoneTextBox\_TextChanged(object sender, EventArgs e)

{

phoneTextBox.Mask = "00000000000";

}

private void nameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label1.Visible = true;

errorProvider1.SetError(label1, "Invalid Input");

label1.Text = "Invalid Input";

}

else

{

label1.Visible = false;

errorProvider1.SetError(label1, "");

label1.Text = "";

}

}

private void mNameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label2.Visible = true;

errorProvider1.SetError(label2, "Invalid Input");

label2.Text = "Invalid Input";

}

else

{

label2.Visible = false;

errorProvider1.SetError(label2, "");

label2.Text = "";

}

}

}

**DeathDataForm:**

public partial class deathdataForm : Form

{

public deathdataForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void submitBtn\_Click(object sender, EventArgs e)

{

string name ,gdeath, fndeath, mndeath, nmbrdeath;

if (nameTextBox.Text == "" || mNameTextBox.Text == "" || fNameTextBox.Text == "" ||

phoneTextBox.Text == "" || comboBox2.Text == "")

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else

{

gdeath = comboBox2.Text;

name = nameTextBox.Text;

mndeath = mNameTextBox.Text;

fndeath = fNameTextBox.Text;

nmbrdeath = phoneTextBox.Text;

deathBL deathData = new deathBL(name, fndeath, mndeath, gdeath, nmbrdeath);

MessageBox.Show("All credentials are Submitted.");

deathDL.Death.Add(deathData);

deathDL.deathStoreData(deathData);

comboBox2.Text = "";

nameTextBox.Clear();

mNameTextBox.Clear();

phoneTextBox.Clear();

fNameTextBox.Clear();

}

}

private void phoneTextBox\_TextChanged(object sender, EventArgs e)

{

phoneTextBox.Mask = "00000000000";

}

private void nameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label1.Visible = true;

errorProvider1.SetError(label1, "Invalid Input");

label1.Text = "Invalid Input";

}

else

{

label1.Visible = false;

errorProvider1.SetError(label1, "");

label1.Text = "";

}

}

private void fNameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label2.Visible = true;

errorProvider1.SetError(label2, "Invalid Input");

label2.Text = "Invalid Input";

}

else

{

label2.Visible = false;

errorProvider1.SetError(label2, "");

label2.Text = "";

}

}

private void mNameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label3.Visible = true;

errorProvider1.SetError(label3, "Invalid Input");

label3.Text = "Invalid Input";

}

else

{

label3.Visible = false;

errorProvider1.SetError(label3, "");

label3.Text = "";

}

}

}

**BirthViewData:**

public partial class birthViewForm : Form

{

public birthViewForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void birthViewForm\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = birthDL.Baby;

}

}

**DeathViewData:**

public partial class deathViewForm : Form

{

public deathViewForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

dataGridView1.DataSource = deathDL.Death;

}

}

**BirthDeleteData:**

public partial class BirthDeleteForm : Form

{

public BirthDeleteForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

dataGridView1.DataSource = birthDL.Baby;

}

private void Databind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = birthDL.Baby;

dataGridView1.Refresh();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

birthBL data = (birthBL)dataGridView1.CurrentRow.DataBoundItem;

if (dataGridView1.Columns["Delete"].Index == e.ColumnIndex)

{

birthDL.deleteBirthFromFile(data);

birthDL.birthStoreDataForDelete();

Databind();

}

}

}

**DeathDeleteData:**

public partial class deathDeleteForm : Form

{

public deathDeleteForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

dataGridView1.DataSource = deathDL.Death;

}

private void Databind()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = birthDL.Baby;

dataGridView1.Refresh();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

deathBL data = (deathBL)dataGridView1.CurrentRow.DataBoundItem;

if (dataGridView1.Columns["Delete"].Index == e.ColumnIndex)

{

deathDL.deleteDeathFromFile(data);

deathDL.deathStoreDataFromFile();

Databind();

}

}

}

**RoomForm:**

public partial class roomForm : Form

{

public roomForm()

{

InitializeComponent();

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

AdminMenuForm obj = new AdminMenuForm();

obj.Show();

}

}

**PatientMenuForm:**

public partial class PatientMenuForm : Form

{

public PatientMenuForm()

{

InitializeComponent();

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Form1 obj = new Form1();

obj.Show();

}

private void Pbtn1\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PersonalDetailForm obj = new PersonalDetailForm();

obj.Show();

}

private void Pbtn2\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PersonalViewForm obj = new PersonalViewForm();

obj.Show();

}

private void Pbtn3\_Click\_1(object sender, EventArgs e)

{

this.Hide();

medicalDetailForm obj = new medicalDetailForm();

obj.Show();

}

private void Pbtn4\_Click\_1(object sender, EventArgs e)

{

this.Hide();

medicalViewForm obj = new medicalViewForm();

obj.Show();

}

private void Pbtn5\_Click\_1(object sender, EventArgs e)

{

this.Hide();

DoctorForm obj = new DoctorForm();

obj.Show();

}

private void Pbtn6\_Click\_1(object sender, EventArgs e)

{

this.Hide();

RecommendationForm obj = new RecommendationForm();

obj.Show();

}

private void Pbtn7\_Click(object sender, EventArgs e)

{

this.Hide();

AppointmentsForm obj = new AppointmentsForm();

obj.Show();

}

private void Pbtn8\_Click\_1(object sender, EventArgs e)

{

this.Hide();

feedBackForm obj = new feedBackForm();

obj.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

}

**PersonalDetailForm:**

public partial class PersonalDetailForm : Form

{

private void submitBtn\_Click\_1(object sender, EventArgs e)

{

string patient\_name, patientf\_name, patient\_gender;

if (nameTextBox.Text == "" || fNameTextBox.Text == "" || comboBox2.Text == "")

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else

{

patient\_name = nameTextBox.Text;

patientf\_name = fNameTextBox.Text;

patient\_gender = comboBox2.Text;

personalBL data = new personalBL(patient\_name, patientf\_name, patient\_gender);

MessageBox.Show(" All Credentials are Submitted.");

//for specific person specific data

personalDL.addIntoList(data,CredentialDL.idx);

personalDL.PersonalStoreData(data);

//For receptionst view

personalDL.P.Add(data);

personalDL.AllPersonalStoreData(data);

nameTextBox.Clear();

fNameTextBox.Clear();

comboBox2.Text = "";

}

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

private void nameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label1.Visible = true;

errorProvider1.SetError(label1, "Invalid Input");

label1.Text = "Invalid Input";

}

else

{

label1.Visible = false;

errorProvider1.SetError(label1, "");

label1.Text = "";

}

}

private void fNameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label2.Visible = true;

errorProvider1.SetError(label2, "Invalid Input");

label2.Text = "Invalid Input";

}

else

{

label2.Visible = false;

errorProvider1.SetError(label2, "");

label2.Text = "";

}

}

}

**MedicalDetailForm:**

public partial class medicalDetailForm : Form

{

public medicalDetailForm()

{

InitializeComponent();

}

private void submitBtn\_Click\_1(object sender, EventArgs e)

{

string patient\_date, patient\_group, patient\_disease;

if (comboBox1.Text == "" || dateTimePicker1.Text == "" || comboBox2.Text == "" )

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else

{

patient\_group = comboBox1.Text;

patient\_date = dateTimePicker1.Text;

patient\_disease = comboBox2.Text;

medicalBL data = new medicalBL(patient\_date, patient\_group,patient\_disease );

MessageBox.Show(" All Credentials are Submitted.");

medicalDL.addIntoMedicalList(data,CredentialDL.idx);

medicalDL.M.Add(data);

medicalDL.medicalStoreData(data);

medicalDL.AllmedicalStoreData(data);

comboBox1.Text = "";

dateTimePicker1.Text = "";

comboBox2.Text = "";

}

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

private void comboBox1\_TextChanged(object sender, EventArgs e)

{

if (comboBox1.Text == "A+" || comboBox1.Text == "AB+" || comboBox1.Text == "B+" || comboBox1.Text == "O+" ||

comboBox1.Text == "A-" || comboBox1.Text == "AB-" || comboBox1.Text == "B-" || comboBox1.Text == "O-")

{

comboBox1.ForeColor = Color.Black;

}

else

{

comboBox1.ForeColor = Color.Red;

}

}

}

}

**PersonalViewForm:**

public partial class PersonalViewForm : Form

{

public PersonalViewForm()

{

InitializeComponent();

}

private void PersonalViewForm\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = CredentialDL.SignUpdata[CredentialDL.idx].SeparateDataList;

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

}

**MedicalViewForm:**

public partial class medicalViewForm : Form

{

public medicalViewForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

}

private void medicalViewForm\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = CredentialDL.SignUpdata[CredentialDL.idx].SeparateDataListForMedical;

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

}

**AppointmentsForm:**

public partial class AppointmentsForm : Form

{

public AppointmentsForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

private void submitBtn\_Click(object sender, EventArgs e)

{

string name = Form2.username;

string doctor, day;

if (comboBox1.Text == "" || comboBox2.Text == "")

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else

{

doctor = comboBox1.Text;

day = comboBox2.Text;

appointment data = new appointment(name,doctor, day);

MessageBox.Show("Request has been sent");

appointmentDL.AppointmentList.Add(data);

appointmentDL.storeData(name,data);

comboBox1.Text = "";

comboBox2.Text = "";

}

}

}

**DoctorsForm:**

public partial class DoctorForm : Form

{

public DoctorForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

}

**RecommendationForm:**

public partial class RecommendationForm : Form

{

public RecommendationForm()

{

InitializeComponent();

}

private void submitBtn\_Click\_1(object sender, EventArgs e)

{

if (comboBox1.Text == "Fever,Flue or Headache")

{

MessageBox.Show( "You are recommended to go to Dr.Ali in room 5");

}

else if (comboBox1.Text == "Viral Infections")

{

MessageBox.Show("You are recommended to go to Dr.Ali in room 5");

}

else if (comboBox1.Text == "Others")

{

MessageBox.Show("You are recommended to go to Dr.Hammad in room 5");

}

else if (comboBox1.Text == "Delivery Case")

{

MessageBox.Show("You are recommended to go to Dr.Ashraf in room 6 ");

}

else if (comboBox1.Text == "Eye Infections")

{

MessageBox.Show("You are recommended to go to Dr.Ayesha in room 4 ");

}

else if (comboBox1.Text == "Ear, Nose and Throat infections")

{

MessageBox.Show("You are recommended to go to Dr.Hania in room 3 ");

}

else

{

MessageBox.Show("Please enter correct input. ");

}

}

private void BackButton\_Click\_2(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

}

**FeedbackForm:**

public partial class feedBackForm : Form

{

public feedBackForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

PatientMenuForm obj = new PatientMenuForm();

obj.Show();

}

private void submitBtn\_Click(object sender, EventArgs e)

{

if(feedbackTextBox.Text == "")

{

MessageBox.Show("Not Submitted.Please fill all Credentials.");

}

else

{

string Name, Comment;

Name = nameTextBox.Text;

Comment = feedbackTextBox.Text;

feedback data = new feedback(Name, Comment);

MessageBox.Show("Submitted.");

feedbackDL.CommentList.Add(data);

feedbackDL.FeedbackStoreData(data);

nameTextBox.Clear();

feedbackTextBox.Clear();

}

}

private void nameTextBox\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && !char.IsWhiteSpace(e.KeyChar))

{

e.Handled = true;

label1.Visible = true;

errorProvider1.SetError(label1, "Invalid Input");

label1.Text = "Invalid Input";

}

else

{

label1.Visible = false;

errorProvider1.SetError(label1, "");

label1.Text = "";

}

}

}

**ReceptionistViewMenu:**

public partial class ReceptionistViewMenu : Form

{

public ReceptionistViewMenu()

{

InitializeComponent();

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

Form1 obj = new Form1();

obj.Show();

}

private void Pbtn2\_Click\_1(object sender, EventArgs e)

{

this.Hide();

viewPdetailInReceptionist obj = new viewPdetailInReceptionist();

obj.Show();

}

private void Pbtn4\_Click\_1(object sender, EventArgs e)

{

this.Hide();

ReceptionistMedicalView obj = new ReceptionistMedicalView();

obj.Show();

}

private void Pbtn8\_Click\_1(object sender, EventArgs e)

{

this.Hide();

feedbackViewForm obj = new feedbackViewForm();

obj.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

AppointmentViewForm obj = new AppointmentViewForm();

obj.Show();

}

}

**ReceptionistMedicalView:**

public partial class ReceptionistMedicalView : Form

{

public ReceptionistMedicalView()

{

InitializeComponent();

}

private void ReceptionistMedicalView\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = medicalDL.M;

}

private void ApplicationLabel\_Click(object sender, EventArgs e)

{

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

ReceptionistViewMenu obj = new ReceptionistViewMenu();

obj.Show();

}

}

**ViewPdetailinReceptionist:**

public partial class viewPdetailInReceptionist : Form

{

public viewPdetailInReceptionist()

{

InitializeComponent();

}

private void viewPdetailInReceptionist\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = personalDL.P;

}

private void BackButton\_Click\_1(object sender, EventArgs e)

{

this.Hide();

ReceptionistViewMenu obj = new ReceptionistViewMenu();

obj.Show();

}

}

**AppointmentViewForm:**

public partial class AppointmentViewForm : Form

{

public AppointmentViewForm()

{

InitializeComponent();

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

ReceptionistViewMenu obj = new ReceptionistViewMenu();

obj.Show();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

dataGridView1.DataSource = appointmentDL.AppointmentList;

}

}

**FeedbackViewForm:**

public partial class feedbackViewForm : Form

{

public feedbackViewForm()

{

InitializeComponent();

}

private void feedbackViewForm\_Load(object sender, EventArgs e)

{

dataGridView1.DataSource = feedbackDL.CommentList;

}

private void BackButton\_Click(object sender, EventArgs e)

{

this.Hide();

ReceptionistViewMenu obj = new ReceptionistViewMenu();

obj.Show();

}

}