**Hospital Management System**



Session: 2022 – 2025

**Submitted by:**

Mudasir Ahmad 2022-CS-77

**Supervised by:**

Prof.Muhammad Awais Hassan

Prof. Laeeq Khan Niazi & Ma’am Maida Shahid

Department of Computer Science

**University of Engineering and Technology**

**Lahore, Pakistan**

Contents

[Project Details: 3](#_Toc141316365)

[Users: 3](#_Toc141316366)

[Functional Requirements: 3](#_Toc141316367)

[Project fulfilling OOP Concepts till Week 8: 4](#_Toc141316368)

[CRC Diagram: 5](#_Toc141316369)

[Code: 7](#_Toc141316370)

[Class Details: 75](#_Toc141316371)

[WireFrames: 75](#_Toc141316372)

[Conclusion: 87](#_Toc141316373)

**YouTube Link:**

<https://youtu.be/r2KUGqOuCe8>

# Project Details:

Hospital Management System is used for managing hospital work because manual management is very difficult. It saves the time of patients and staff because the patient cannot need to stand in line. Hospital Management System is basically about adding Patients which is used by Management staff to perform different operations on their data such as generating reports, suggesting medicines, and calculating their bills. Management Staff performs the Patient **CRUD** operations. Management staff also manages the doctor’s schedule and appointments. A patient can check a doctor prescription, view his medical report, and bill. If the patient has a common disease, this software asks for the symptoms and suggests the medicine according to the symptoms. This system also manages the Pharmacy store of the Hospital in which pharmacists can add their stock to the system, check the expired medicines, and view the medicines in demand. This system helps the hospital to run more efficiently by automating patient information and medical records, reducing errors, and improving financial management.

# Users:

* The **management staff/Admin** of the hospital can use this application. This application can help them in reducing their work and provide more accuracy.
* The **pharmacist** can use it to manage stocks of medicine.
* **Patients** can use our application to see their medical report and expenses.

# Functional Requirements:

|  |  |  |
| --- | --- | --- |
| ***As a*** | ***I want to perform*** | ***So that I can*** |
| Management Staff | *Add new patient*  *Manage the doctor's schedule*  *Manage the appointment with a doctor*  *Add the record of expenses* | Maintain his record  Inform patients  Save the time of patients  Calculate the bill according to the services provided to a patient |
| *Patient* | *View the medical report*  *Check the bill* | *remain up-to-date about my health*  *pay for it* |
| Pharmacist | Check the medicines in demand  Check the Expired medicines | Send a request to the supplier for those medicines  Delete it from my record |

# Project fulfilling OOP Concepts till Week 8:

I have used the Concepts of OOP in my project. Inheritance is a powerful feature that enables us to create hierarchical relationships between classes. I have used inheritance based on the different roles of users and used static polymorphism on validations. I have used the association such as aggregation and composition in different stages of this project. When the object of the patient is created then it automatically creates the object of its report and bill because there is no existence of a patient report without the patient this is composition. A patient can book many appointments with different doctors. A customer can purchase many medicines from a pharmacy store because these things are independent this is aggregation. OOP promotes modularity by breaking down complex problems into smaller, manageable components called objects. These objects encapsulate data and behavior, making it easier to understand, maintain, and modify code. Objects can also be reused in different parts of a program, leading to code reusability, which saves development time and effort.

# CRC Diagram:

has

∞

∞

∞

∞

∞

∞

∞

∞

1

1

1

1

1

1

1

1

∞

1

contains

contains

has

buy

book

contains

contains

contains

contains

contains

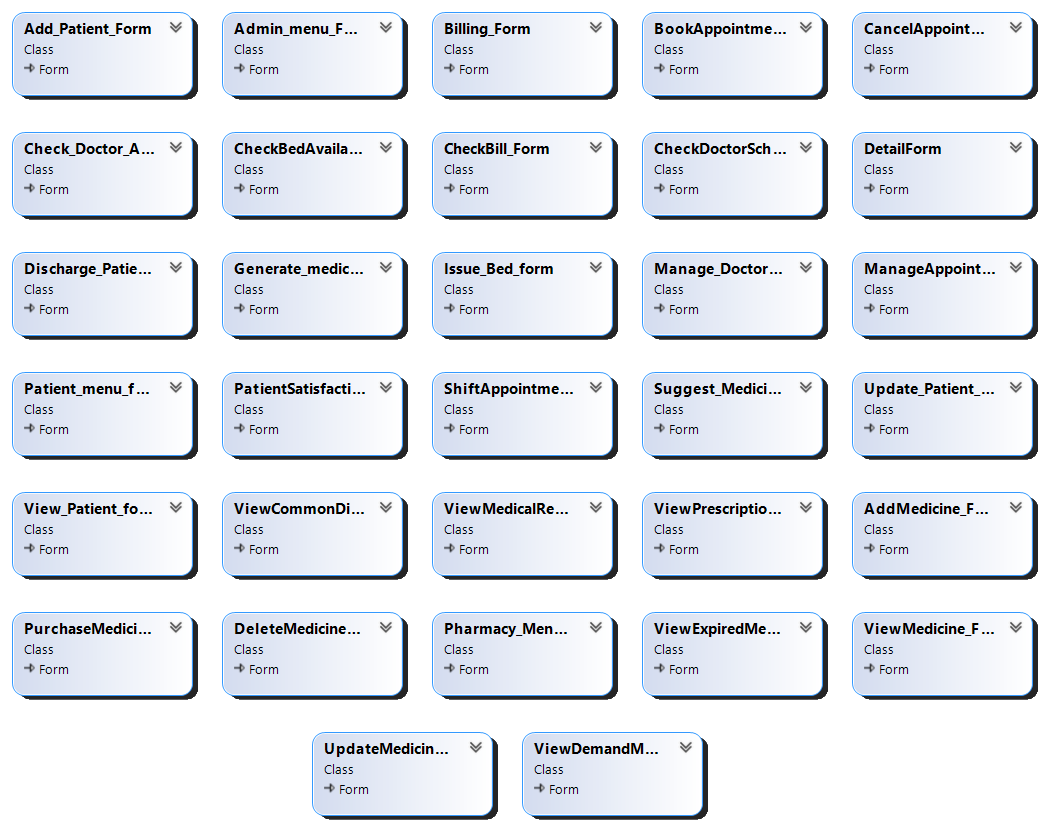
contains

is

is

is

is





# Code:

**BL Code:** MUser.cs

class MUser

{

protected string name;

protected string password;

protected string role;

public MUser()

{

name = "";

password = "";

role = "";

}

public MUser(string name, string password)

{

this.name = name;

this.password = password;

}

public MUser(string name, string password, string role)

{

this.name = name;

this.password = password;

this.role = role;

}

public string getUsername()

{

return name;

}

public string getPassword()

{

return password;

}

public string getRole()

{

return role;

}

public void setUsername(string name)

{

this.name = name;

}

public void setPassword(string password)

{

this.password = password;

}

public void setRole(string role)

{

this.role = role;

}

public bool isAdmin()

{

if (role == "Admin" || role == "admin")

{

return true;

}

return false;

}

public bool isPatient()

{

if (role == "Patient" || role == "patient")

{

return true;

}

return false;

}

public bool isPharmacist()

{

if (role == "Pharmacist" || role == "pharmacist")

{

return true;

}

return false;

}

}

**BL Code:** Patient.cs

class Patient : MUser

{

public Patient(string username, string password, string role, string name, int age, string contactNo, string cnic, string address, string visit\_date, string medicalHistory) : base(username, password, role)

{

this.name = name;

this.age = age;

this.contactNo = contactNo;

this.cnic = cnic;

this.address = address;

this.visit\_date = visit\_date;

this.medicalHistory = medicalHistory;

}

public Patient(string username, string password, string role) : base(username, password, role)

{

}

public Patient()

{

}

private string name;

private int age;

private string contactNo;

private string cnic;

private string address;

private string visit\_date;

private string medicalHistory;

public Report PatientReport = new Report();

public PatientBill bill = new PatientBill();

public string getName() { return name; }

public int getAge() { return age; }

public string getContactno() { return contactNo; }

public string getCnic() { return cnic; }

public string getAddress() { return address; }

public string getVisitdate() { return visit\_date; }

public string getMedicalhistory() { return medicalHistory; }

public void setName(string name) { this.name = name; }

public void setAge(int age) { this.age = age; }

public void setContactno(string contactNo) { this.contactNo = contactNo; }

public void setCnic(string cnic) { this.cnic = cnic; }

public void setAddress(string address) { this.address = address; }

public void setVisitdate(string visit\_date) { this.visit\_date = visit\_date; }

public void setMedicalhistory(string medicalHistory) { this.medicalHistory = medicalHistory; }

private string medicine;

public string getMedicine() { return medicine; }

public void SymptomsCheck(string[] symptoms,int i,ref char check1,ref char check2,ref char check3)

{

if (symptoms[i] == "fever" || symptoms[i] == "pain" || symptoms[i] == "headache")

{

check1 = 'p';

}

if (symptoms[i] == "cold" || symptoms[i] == "flu")

{

check2 = 'c';

}

if (symptoms[i] == "cough")

{

check3 = 'h';

}

}

public string GiveMedicine(char check1,char check2,char check3)

{

string medicine1 = "",medicine2="",medicine3="";

if (check1 == 'p')

{

medicine1 = " Panadol";

}

else

{

medicine1 = "";

}

if (check2 == 'c')

{

medicine2 = " Crocin C&F max";

}

else

{

medicine2 = "";

}

if (check3 == 'h')

{

medicine3 = " Hydryllin";

}

else

{

medicine3 = "";

}

return medicine = medicine1 + medicine2 + medicine3;

}

}

**BL Code:** Management staff.cs

class Management\_Staff:MUser

{

public Management\_Staff(string username, string password, string role) : base(username, password, role)

{

}

public Management\_Staff(string username, string password) : base(username, password)

{

}

}

**BL Code:** Pharmacist.cs

class Pharmacist:MUser

{

public Pharmacist(string username, string password, string role) : base(username, password, role)

{

}

public Pharmacist(string username, string password) : base(username, password)

{

}

}

**BL Code:** Doctor.cs

class Doctor:MUser

{

public Doctor(string username, string password, string role,string Scheduledate,string Doctorname,string profession,string startingtime,string endingtime):base(username,password,role)

{

this.Scheduledate = Scheduledate;

this.Doctorname = Doctorname;

this.profession = profession;

this.startingtime = startingtime;

this.endingtime = endingtime;

}

public Doctor() { }

private string Scheduledate, Doctorname, profession, startingtime, endingtime;

public List<Appointment> myAppointments = new List<Appointment>();

public string getScheduledate() { return Scheduledate; }

public string getDoctorname() { return Doctorname; }

public string getProfession() { return profession; }

public string getStartingtime() { return startingtime; }

public string getEndingtime() { return endingtime; }

public void setScheduledate(string Scheduledate) { this.Scheduledate = Scheduledate; }

public void setDoctorname(string Doctorname) { this.Doctorname = Doctorname; }

public void setProfession(string profession) { this.profession = profession; }

public void setStartingtime(string startingtime) { this.startingtime = startingtime; }

public void setEndingtime(string endingtime) { this.endingtime = endingtime; }

public void AddData\_inAppointmentList(List<Appointment> appointments)

{

bool check = false;

foreach(var appointment in appointments)

{

foreach (var Appoinment in myAppointments)

{

if (Appoinment.getName() != appointment.getName() || Appoinment.getDoctor() != appointment.getDoctor())

{

check = false;

}

else

{

check = true;

break;

}

}

if (check == false)

{

if (Doctorname == appointment.getDoctor())

{

myAppointments.Add(appointment);

}

}

}

}

}

**BL Code:** Appointment.cs

class Appointment

{

private string name;

private string dateappoint;

private string doctor;

private string timeappoint;

public string getName() { return name; }

public string getDateappoint() { return dateappoint; }

public string getDoctor() { return doctor; }

public string getTimeappoint() { return timeappoint; }

public void setName(string name) { this.name = name; }

public void setDateappoint(string dateappoint) { this.dateappoint = dateappoint; }

public void setDoctor(string doctor) { this.doctor = doctor; }

public void setTimeappoint(string timeappoint) { this.timeappoint = timeappoint; }

public Appointment() { }

public Appointment(string name, string dateappoint, string doctor)

{

this.name = name;

this.dateappoint = dateappoint;

this.doctor = doctor;

}

public Appointment(string name, string dateappoint, string doctor,string timeappoint)

{

this.name = name;

this.dateappoint = dateappoint;

this.doctor = doctor;

this.timeappoint = timeappoint;

}

public int Converttime(string timeappoint)

{

int time = 0;

if (timeappoint[0] == '0')

{

time = 0;

}

if (timeappoint[0] == '1')

{

time = 1;

}

if (timeappoint[0] == '2')

{

time = 2;

}

if (timeappoint[0] == '3')

{

time = 3;

}

if (timeappoint[0] == '4')

{

time = 4;

}

if (timeappoint[0] == '5')

{

time = 5;

}

if (timeappoint[0] == '6')

{

time = 6;

}

if (timeappoint[0] == '7')

{

time = 7;

}

if (timeappoint[0] == '8')

{

time = 8;

}

if (timeappoint[0] == '9')

{

time = 9;

}

if (timeappoint[0] == '1' && timeappoint[1] == '0')

{

time = 10;

}

if (timeappoint[0] == '1' && timeappoint[1] == '1')

{

time = 11;

}

if (timeappoint[0] == '1' && timeappoint[1] == '2')

{

time = 12;

}

return time;

}

**BL Code:** PatientBill.cs

class PatientBill

{

//private string PatientnameBill;

private string services;

private int daysPatientstayed;

private float bedcharge;

private float labtest;

private float total\_bill;

public string getServices() { return services; }

public int getDaysPatientstayed() { return daysPatientstayed; }

public float getBedcharge() { return bedcharge; }

public float getLabtest() { return labtest; }

public float getTotalbill() { return total\_bill; }

public void setServices(string services) { this.services = services; }

public void setDayspatientstayed(int daysPatientstayed) { this.daysPatientstayed = daysPatientstayed; }

public void setBedcharge(float bedcharge) { this.bedcharge = bedcharge; }

public void setLabtest(float labtest) { this.labtest = labtest; }

public void setTotalbill(float total\_bill) { this.total\_bill = total\_bill; }

public int ServiceCheck()

{

int hospitalizationcharge = 0;

int emergency\_room\_charge = 0;

int final\_charges;

if (services == "hospitalization")

{

hospitalizationcharge = 500;

}

if (services == "emergency room service")

{

emergency\_room\_charge = 2000;

}

return final\_charges = hospitalizationcharge + emergency\_room\_charge;

}

public void calculate\_totalBill(int hospitalizationcharge,int emergency\_room\_charge,float bedcharge,float totalpricemedicine)

{

total\_bill = (hospitalizationcharge \* daysPatientstayed) + (emergency\_room\_charge \* daysPatientstayed) + (bedcharge \* daysPatientstayed) + (labtest + totalpricemedicine);

}

}

**BL Code:** Report.cs

class Report

{

private string laboratorytest;

private int cholestrolLevel;

private string cholestrol;

private string typhoid;

private string malaria;

private string anemia;

private string diagnostictest;

public Report() { }

public Report(string laboratorytest, int cholestrolLevel,string cholestrol,string typhoid,string malaria,string anemia,string diagnostictest)

{

this.laboratorytest = laboratorytest;

this.cholestrolLevel = cholestrolLevel;

this.cholestrol = cholestrol;

this.typhoid = typhoid;

this.malaria = malaria;

this.anemia = anemia;

this.diagnostictest = diagnostictest;

}

public string getLaboratorytest() { return laboratorytest; }

public int getCholestrolLevel() { return cholestrolLevel; }

public string getCholestrol() { return cholestrol; }

public string getTyphoid() { return typhoid; }

public string getMalaria() { return malaria; }

public string getAnemia() { return anemia; }

public string getDiagnostictest() { return diagnostictest; }

public void setLaboratorytest(string laboratorytest) { this.laboratorytest = laboratorytest; }

public void setCholestrolLevel(int cholestrolLevel) { this.cholestrolLevel = cholestrolLevel; }

public void setCholestrol(string cholestrol) { this.cholestrol = cholestrol; }

public void setTyphoid(string typhoid) { this.typhoid = typhoid; }

public void setMalaria(string malaria) { this.malaria = malaria; }

public void setAnemia(string anemia) { this.anemia = anemia; }

public void setDiagnostictest(string diagnostictest) { this.diagnostictest = diagnostictest; }

public void Generatemedicalreport()

{

if (laboratorytest == "blood test")

{

if (cholestrolLevel < 100)

{

cholestrol = "Optimum";

}

if (cholestrolLevel >= 100 && cholestrolLevel < 130)

{

cholestrol = "Fairly Good";

}

if (cholestrolLevel >= 130 && cholestrolLevel < 160)

{

cholestrol = "Borederline High";

}

if (cholestrolLevel >= 160 && cholestrolLevel < 190)

{

cholestrol = "High";

}

if (cholestrolLevel >= 190)

{

cholestrol = "Very High";

}

}

}

}

**BL Code:** Medicine.cs

class Medicine

{

private string addmedicine;

private int quantity;

private string expirydate;

private float price;

private int demandcheck;

public string getMedicinename() { return addmedicine; }

public int getQuantity() { return quantity; }

public string getExpirydate() { return expirydate; }

public float getPrice() { return price; }

public int getDemandcheck() { return demandcheck; }

public void setMedicinename(string addmedicine) { this.addmedicine = addmedicine; }

public void setQuantity(int quantity) { this.quantity = quantity; }

public void setExpirydate(string expirydate) { this.expirydate = expirydate; }

public void setPrice(float price) { this.price = price; }

public void setDemandcheck(int demandcheck) { this.demandcheck = demandcheck; }

public Medicine()

{

addmedicine = "";

quantity = 0;

expirydate = "";

price = 0.0F;

demandcheck = 0;

}

public Medicine(string addmedicine, int quantity, string expirydate, float price, int demandcheck)

{

this.addmedicine = addmedicine;

this.quantity = quantity;

this.expirydate = expirydate;

this.price = price;

this.demandcheck = demandcheck;

}

}

**BL Code:** Customer.cs

class Customer

{

private string customerName;

private List<Medicine> PurchaseMedicines = new List<Medicine>();

public Customer(string customerName)

{

this.customerName = customerName;

}

public Customer()

{

}

public string getcustomerName()

{

return customerName;

}

public void setcustomerName(string customerName)

{

this.customerName = customerName;

}

public List<Medicine> getPurchasedMedicines()

{

return PurchaseMedicines;

}

public void setPurchasedMedicines(List<Medicine> PurchaseMedicines)

{

this.PurchaseMedicines = PurchaseMedicines;

}

public void AddMedicineInList(Medicine PurchaseMed)

{

PurchaseMedicines.Add(PurchaseMed);

}

public Medicine purchasemedicine(List<Medicine> med, Medicine purchasemed1)

{

foreach (Medicine i in med)

{

if (purchasemed1.getMedicinename() == i.getMedicinename())

{

if (purchasemed1.getQuantity() <= i.getQuantity())

{

int final\_quantity = i.getQuantity() - purchasemed1.getQuantity();

i.setQuantity(final\_quantity);

purchasemed1.setExpirydate(i.getExpirydate());

purchasemed1.setPrice(i.getPrice());

purchasemed1.setDemandcheck(i.getDemandcheck());

return i;

}

}

}

return null;

}

public float Calculate\_Bill()

{

float total\_price = 0F;

foreach(var med in PurchaseMedicines)

{

float perPrice = med.getPrice() \* med.getQuantity();

total\_price = total\_price + perPrice;

}

return total\_price;

}

}

**BL Code:** ExpiredMedicine.cs

class ExpiredMedicine

{

private string expiredmedicine;

public string getExpiredmedicine() { return expiredmedicine; }

public void setExpiredmedicine(string expiredmedicine) { this.expiredmedicine = expiredmedicine; }

public ExpiredMedicine() { }

public ExpiredMedicine(string expiredmedicine)

{

this.expiredmedicine = expiredmedicine;

}

public static int convertDatetodays(string date)

{

char[] datedays = { date[0], date[1], '\0' };

char[] datemonths = { date[3], date[4], '\0' };

char[] dateyears = { date[6], date[7], date[8], date[9], '\0' };

string dateDays = new string(datedays);

string dateMonths = new string(datemonths);

string dateYears = new string(dateyears);

return int.Parse(dateDays) + int.Parse(dateMonths) \* 30 + (int.Parse(dateYears) - 2000) \* 365;

}

public static bool isExpired(string currentDate, string expiryDate)

{

int currentDays = convertDatetodays(currentDate);

int expiryDays = convertDatetodays(expiryDate);

return (currentDays > expiryDays);

}

}

**DL Code:** MUserCRUD.cs

class MUser\_CRUD

{

public static List<MUser> users = new List<MUser>();

public static bool readData(string path)

{

if (File.Exists(path))

{

StreamReader fileVariable = new StreamReader(path);

string record;

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

{

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

string username = line[0];

string password = line[1];

string role = line[2];

MUser user = new MUser(username, password, role);

storeDataInList(user);

}

fileVariable.Close();

return true;

}

return false;

}

public static void storeDataInList(MUser user)

{

users.Add(user);

}

public static void storeDataInFile(string path, MUser user)

{

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(user.getUsername() + "," + user.getPassword() + "," + user.getRole());

file.Flush();

file.Close();

}

public static MUser signIn(MUser user)

{

foreach (MUser storedUser in users)

{

if (user.getUsername() == storedUser.getUsername() && user.getPassword() == storedUser.getPassword())

{

return storedUser;

}

}

return null;

}

}

**DL Code:** PatientCRUD.cs

class PatientCRUD

{

public static int bed = 150;

public static List<Patient> patients = new List<Patient>();

public static List<bool> bedAvaiables = new List<bool>();

public static void Add\_Patient(Patient p)

{

patients.Add(p);

}

public static bool addPatient\_Dl(Patient p)

{

bool isbool;

isbool = true;

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

{

if (patients[i].getCnic() == p.getCnic())

{

isbool = false;

}

}

bed--;

return isbool;

}

static string path\_patient()

{

string path = "path\\Patient.txt";

return path;

}

public static void save\_patientdata(Patient p)

{

string path = path\_patient();

StreamWriter patient\_file = new StreamWriter(path, true);

patient\_file.WriteLine(p.getName() + "," + p.getAge() + "," + p.getContactno() + "," + p.getCnic() + "," + p.getAddress() + "," + p.getVisitdate() + "," + p.getMedicalhistory());

patient\_file.Flush();

patient\_file.Close();

}

public static void load\_patientdata()

{

string path = path\_patient();

string line;

StreamReader patient\_file = new StreamReader(path);

while ((line = patient\_file.ReadLine()) != null)

{

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

string name = record[0];

int age = int.Parse(record[1]);

string contactNo = record[2];

string cnic = record[3];

string address = record[4];

string visit\_date = record[5];

string medicalHistory = record[6];

Patient p = new Patient();

p.setName(name);

p.setAge(age);

p.setContactno(contactNo);

p.setCnic(cnic);

p.setAddress(address);

p.setVisitdate(visit\_date);

p.setMedicalhistory(medicalHistory);

patients.Add(p);

}

patient\_file.Close();

}

public static Patient update\_patient\_Dl(string patient\_name, string cnicno)

{

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

{

if (patients[i].getName() == patient\_name && patients[i].getCnic() == cnicno)

{

return patients[i];

}

}

return null;

}

public static void update\_patientdata()

{

string path = "path\\Patient.txt";

StreamWriter patient\_file = new StreamWriter(path);

foreach (var Patient in patients)

{

patient\_file.WriteLine(Patient.getName() + "," + Patient.getAge() + "," + Patient.getContactno() + "," + Patient.getCnic() + "," + Patient.getAddress() + "," + Patient.getVisitdate() + "," + Patient.getMedicalhistory());

}

patient\_file.Flush();

patient\_file.Close();

}

public static void add\_medicalReport(Patient p)

{

string path = "path\\medicalReport.txt";

StreamWriter Report\_file = new StreamWriter(path,true);

Report\_file.Write(p.getName() + ",");

Report\_file.Write(p.PatientReport.getLaboratorytest() + ",");

if (p.PatientReport.getLaboratorytest() == "blood test")

{

Report\_file.Write(p.PatientReport.getCholestrolLevel() + ",");

Report\_file.Write(p.PatientReport.getCholestrol() + ",");

Report\_file.Write(p.PatientReport.getTyphoid() + ",");

Report\_file.Write(p.PatientReport.getMalaria() + ",");

Report\_file.Write(p.PatientReport.getAnemia() + ",");

}

Report\_file.Write(p.PatientReport.getDiagnostictest());

Report\_file.WriteLine();

Report\_file.Flush();

Report\_file.Close();

}

public static void load\_medicalReport()

{

string Name, laboratory\_test, Cholestrol, Typhoid, Malaria, Anemia, diagnostic\_test;

int cholestrol\_level;

string line;

string path = "path\\medicalReport.txt";

int i = 0;

StreamReader patient\_file = new StreamReader(path);

while ((line=patient\_file.ReadLine())!=null)

{

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

Name = record[0];

laboratory\_test = record[1];

cholestrol\_level = int.Parse(record[2]);

Cholestrol = record[3];

Typhoid = record[4];

Malaria = record[5];

Anemia = record[6];

diagnostic\_test = record[7];

patients[i].PatientReport.setLaboratorytest(laboratory\_test);

patients[i].PatientReport.setCholestrolLevel(cholestrol\_level);

patients[i].PatientReport.setCholestrol(Cholestrol);

patients[i].PatientReport.setTyphoid(Typhoid);

patients[i].PatientReport.setMalaria(Malaria);

patients[i].PatientReport.setAnemia(Anemia);

patients[i].PatientReport.setDiagnostictest(diagnostic\_test);

i++;

}

patient\_file.Close();

}

public static bool dischargePatient(string discharge\_patient, string dischargepatient\_cnic)

{

bool discharge\_check = false;

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

{

if (patients[i].getCnic() == dischargepatient\_cnic && discharge\_patient == patients[i].getName())

{

discharge\_check = true;

patients.RemoveAt(i);

if (bed >= 1)

{

bed++;

}

break;

}

}

return discharge\_check;

}

public static void dataPatient\_afterDelete()

{

string path = "path\\Patient.txt";

StreamWriter patient\_file = new StreamWriter(path);

foreach (var Patient in patients)

{

patient\_file.WriteLine(Patient.getName() + "," + Patient.getAge() + "," + Patient.getContactno() + "," + Patient.getCnic() + "," + Patient.getAddress() + "," + Patient.getVisitdate() + "," + Patient.getMedicalhistory());

}

patient\_file.Flush();

patient\_file.Close();

}

public static void DataPatientReport\_afterdelete()

{

string path = "path\\medicalReport.txt";

StreamWriter Report\_file = new StreamWriter(path);

foreach (var Patient in patients)

{

Report\_file.Write(Patient.getName() + ",");

Report\_file.Write(Patient.PatientReport.getLaboratorytest() + ",");

if (Patient.PatientReport.getLaboratorytest() == "blood test")

{

Report\_file.Write(Patient.PatientReport.getCholestrolLevel() + ",");

Report\_file.Write(Patient.PatientReport.getCholestrol() + ",");

Report\_file.Write(Patient.PatientReport.getTyphoid() + ",");

Report\_file.Write(Patient.PatientReport.getMalaria() + ",");

Report\_file.Write(Patient.PatientReport.getAnemia() + ",");

}

Report\_file.Write(Patient.PatientReport.getDiagnostictest());

Report\_file.WriteLine();

}

Report\_file.Flush();

Report\_file.Close();

}

public static void dataPatientBill\_afterdelete()

{

string path = "path\\Patient\_bill.txt";

StreamWriter billfile = new StreamWriter(path);

foreach (var Patient in patients)

{

billfile.WriteLine(Patient.getName() + "," + Patient.bill.getDaysPatientstayed() + "," + Patient.bill.getBedcharge() + "," + Patient.bill.getLabtest() + "," + Patient.bill.getTotalbill());

}

billfile.Flush();

billfile.Close();

}

public static void dataafter\_deletefile()

{

dataPatient\_afterDelete();

dataPatientBill\_afterdelete();

DataPatientReport\_afterdelete();

}

public static void add\_billing(Patient p)

{

string path = "path\\Patient\_bill.txt";

StreamWriter billfile = new StreamWriter(path,true);

billfile.WriteLine(p.getName() + "," + p.bill.getDaysPatientstayed() + "," + p.bill.getBedcharge() + "," + p.bill.getLabtest() + "," + p.bill.getTotalbill());

billfile.Flush();

billfile.Close();

}

public static void load\_billing()

{

string patientnamebill;

int dayspatientstayed;

float bedCharge, lab\_test, total\_Bill;

string line;

string path = "path\\Patient\_bill.txt";

int countbill = 0;

StreamReader file = new StreamReader(path);

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

{

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

patientnamebill = record[0];

dayspatientstayed = int.Parse(record[1]);

bedCharge = float.Parse(record[2]);

lab\_test = float.Parse(record[3]);

total\_Bill = float.Parse(record[4]);

//patients[countbill].PatientnameBill = patientnamebill;

patients[countbill].bill.setDayspatientstayed(dayspatientstayed);

patients[countbill].bill.setBedcharge(bedCharge);

patients[countbill].bill.setLabtest(lab\_test);

patients[countbill].bill.setTotalbill(total\_Bill);

countbill++;

}

file.Close();

}

public static void saveBedChecker()

{

string path = "path\\BedAvailability.txt";

StreamWriter file = new StreamWriter(path);

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

file.WriteLine(bedAvaiables[i]);

}

file.Flush();

file.Close();

}

public static void loadBedChecker()

{

string path = "path\\BedAvailability.txt";

string line;

StreamReader file = new StreamReader(path);

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

{

bool record = bool.Parse(line);

bedAvaiables.Add(record);

}

file.Close();

}

}

**DL Code:** DoctorCRUD.cs

class Doctor\_CRUD

{

public static List<Doctor> doctors = new List<Doctor>();

public static void Add\_Doctor(Doctor doctor)

{

doctors.Add(doctor);

}

public static void add\_DoctorSchedule(Doctor doctor)

{

string path = "path\\doctor\_schedule.txt";

StreamWriter Doctor\_schedule\_file = new StreamWriter(path, true);

Doctor\_schedule\_file.WriteLine(doctor.getScheduledate() + "," + doctor.getDoctorname() + "," + doctor.getProfession() + "," + doctor.getStartingtime() + "," + doctor.getEndingtime());

Doctor\_schedule\_file.Flush();

Doctor\_schedule\_file.Close();

}

public static void load\_DoctorScheduleData()

{

string path = "path\\doctor\_schedule.txt";

string scheduledate, doctorname, Profession, starting\_time, ending\_time;

string line;

StreamReader file = new StreamReader(path);

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

{

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

scheduledate = record[0];

doctorname = record[1];

Profession = record[2];

starting\_time = record[3];

ending\_time = record[4];

string Scheduledate = scheduledate;

string Doctorname = doctorname;

string profession = Profession;

string startingtime = starting\_time;

string endingtime = ending\_time;

Doctor d = new Doctor();

d.setScheduledate(Scheduledate);

d.setDoctorname(Doctorname);

d.setProfession(profession);

d.setStartingtime(startingtime);

d.setEndingtime(endingtime);

doctors.Add(d);

}

file.Close();

}

public static void put\_DoctorAppointment\_inList(List<Appointment> appointments)

{

foreach(var Doctor in doctors)

{

Doctor.AddData\_inAppointmentList(appointments);

}

}

}

**DL Code:** AppointmentCRUD.cs

class Appointment\_CRUD

{

public static List<Appointment> appointments = new List<Appointment>();

public static void Add\_Appointment(Appointment appointment)

{

appointments.Add(appointment);

}

public static void canceledappointment(ref string cancelpatient)

{

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

{

if (cancelpatient == appointments[i].getName())

{

appointments.RemoveAt(i);

break;

}

}

}

public static void shiftedappointment(ref string ShiftedPatientname,ref string shiftdate,ref string shifttime)

{

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

{

if (ShiftedPatientname == appointments[i].getName())

{

appointments[i].setDateappoint(shiftdate);

appointments[i].setTimeappoint(shifttime);

}

}

}

public static void aftershiftedORcancelappointment(int num)

{

string path = "path\\appointment.txt";

StreamWriter appointmentFile = new StreamWriter(path);

appointmentFile.WriteLine(appointments[num].getName() + ","+ appointments[num].getDateappoint()+","+ appointments[num].getDoctor()+","+ appointments[num].getTimeappoint());

appointmentFile.Flush();

appointmentFile.Close();

}

public static void DATAaftershiftedORcancelappointment()

{

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

{

aftershiftedORcancelappointment(i);

}

}

public static void add\_appointmentinFile(Appointment appointment)

{

string path = "path\\appointment.txt";

StreamWriter appointmentFile = new StreamWriter(path, true);

appointmentFile.WriteLine(appointment.getName() + "," + appointment.getDateappoint() + "," + appointment.getDoctor() + "," + appointment.getTimeappoint());

appointmentFile.Flush();

appointmentFile.Close();

}

public static void load\_appointment()

{

string appointmentDoctor, appointmentPatientname, Dateappoint, Timeappoint;

string line;

string path = "path\\appointment.txt";

StreamReader appointmentFile = new StreamReader(path);

while ((line = appointmentFile.ReadLine()) != null)

{

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

appointmentPatientname = record[0];

Dateappoint = record[1];

appointmentDoctor = record[2];

Timeappoint = record[3];

string name = appointmentPatientname;

string dateappoint = Dateappoint;

string doctor = appointmentDoctor;

string timeappoint = Timeappoint;

Appointment ap = new Appointment(name, dateappoint, doctor, timeappoint);

appointments.Add(ap);

Doctor\_CRUD.put\_DoctorAppointment\_inList(appointments);

}

appointmentFile.Close();

}

}

**DL Code:** MedicineCRUD.cs

class Medicine\_CRUD

{

public static List<Medicine> meds = new List<Medicine>();

public static void Addmedicine(Medicine med1)

{

meds.Add(med1);

}

public static string pathOfMedicineFile()

{

string path = "path\\medicine.txt";

return path;

}

public static void addMedicineinFile()

{

string path = pathOfMedicineFile();

StreamWriter medicineFile = new StreamWriter(path, true);

medicineFile.WriteLine(meds[meds.Count - 1].getMedicinename() + "," + meds[meds.Count - 1].getQuantity() + "," + meds[meds.Count - 1].getDemandcheck() + "," + meds[meds.Count - 1].getExpirydate() + "," + meds[meds.Count - 1].getPrice());

medicineFile.Flush();

medicineFile.Close();

}

public static void load\_medicineFile()

{

string Addmedicine, Expirydate;

int Quantity, Demandcheck;

float Price;

string path = pathOfMedicineFile();

string record;

StreamReader file = new StreamReader(path);

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

{

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

Addmedicine = line[0];

Quantity = int.Parse(line[1]);

Demandcheck = int.Parse(line[2]);

Expirydate = line[3];

Price = float.Parse(line[4]);

Medicine med1 = new Medicine();

med1.setMedicinename(Addmedicine);

med1.setQuantity(Quantity);

med1.setDemandcheck(Demandcheck);

med1.setExpirydate(Expirydate);

med1.setPrice(Price);

meds.Add(med1);

}

file.Close();

}

public static Medicine Updatemedicine\_Check(string updatemedicine)

{

foreach (Medicine i in meds)

{

if (updatemedicine == i.getMedicinename())

{

return i;

}

}

return null;

}

public static void SaveChange()

{

string path = pathOfMedicineFile();

StreamWriter medicineFile = new StreamWriter(path);

foreach (var med in meds)

{

medicineFile.WriteLine(med.getMedicinename() + "," + med.getQuantity() + "," + med.getDemandcheck() + "," + med.getExpirydate() + "," + med.getPrice());

}

medicineFile.Flush();

medicineFile.Close();

}

public static void saveUpdatedmedicine\_data()

{

string path = pathOfMedicineFile();

StreamWriter medicineFile = new StreamWriter(path);

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

{

updatemedicinedata(i,medicineFile);

}

medicineFile.Flush();

medicineFile.Close();

}

public static void updatemedicinedata(int num,StreamWriter medicineFile)

{

medicineFile.WriteLine(meds[num].getMedicinename() + "," + meds[num].getQuantity() + "," + meds[num].getDemandcheck() + "," + meds[num].getExpirydate() + "," + meds[num].getPrice());

}

public static bool deletemedicine(string delmedicine)

{

bool delmedicineCheck = false;

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

{

if (delmedicine == meds[i].getMedicinename())

{

delmedicineCheck = true;

meds.RemoveAt(i);

break;

}

}

return delmedicineCheck;

}

public static void AFTERDeleteMedicine(int num,StreamWriter medicineFile)

{

medicineFile.WriteLine(meds[num].getMedicinename() + "," + meds[num].getQuantity() + "," + meds[num].getDemandcheck() + "," + meds[num].getExpirydate() + "," + meds[num].getPrice());

}

public static void DataAFTERDeleteMedicine()

{

string path = pathOfMedicineFile();

StreamWriter medicineFile = new StreamWriter(path);

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

{

AFTERDeleteMedicine(i,medicineFile);

}

medicineFile.Flush();

medicineFile.Close();

}

}

**DL Code:** CustomerCRUD.cs

class Customer\_CRUD

{

public static List<Customer> customers = new List<Customer>();

public static void AddCustomerinList(Customer c)

{

customers.Add(c);

}

public static void add\_purchaseMedicineinFile()

{

string path = "path\\purchaseMedicine.txt";

StreamWriter purchasefile = new StreamWriter(path, true);

purchasefile.Write(customers[customers.Count - 1].getcustomerName() + ":");

List<Medicine> purchasemed = customers[customers.Count - 1].getPurchasedMedicines();

for(int i = 0; i < purchasemed.Count - 1; i++)

{

purchasefile.Write(purchasemed[i].getMedicinename() + "," + purchasemed[i].getQuantity() + "," + purchasemed[i].getPrice());

purchasefile.Write("-");

}

purchasefile.Write(purchasemed[purchasemed.Count-1].getMedicinename() + "," + purchasemed[purchasemed.Count - 1].getQuantity() + "," + purchasemed[purchasemed.Count - 1].getPrice());

purchasefile.WriteLine();

purchasefile.Flush();

purchasefile.Close();

}

public static void load\_purchaseMedicine()

{

string path = "path\\purchaseMedicine.txt";

StreamReader purchasefile = new StreamReader(path);

string line, Customername, purchasemedicine, Quantitypurchase, total\_Price;

string Purchasedmedicines,separatePurchaseMed;

while ((line = purchasefile.ReadLine()) != null)

{

Customer customer = new Customer();

string[] record = line.Split(':');

Customername = record[0];

customer.setcustomerName(Customername);

Purchasedmedicines = record[1];

string[] record1 = Purchasedmedicines.Split('-');

for(int i = 0; i < record1.Length; i++)

{

separatePurchaseMed = record1[i];

string[] record3 = separatePurchaseMed.Split(',');

purchasemedicine = record3[0];

Quantitypurchase = record3[1];

total\_Price = record3[2];

Medicine purchase\_medicine = new Medicine();

purchase\_medicine.setMedicinename(purchasemedicine);

purchase\_medicine.setQuantity(int.Parse(Quantitypurchase));

purchase\_medicine.setPrice(float.Parse(total\_Price));

customer.AddMedicineInList(purchase\_medicine);

}

customers.Add(customer);

}

purchasefile.Close();

}

}

**DL Code:** ExpiredMedicineCRUD.cs

class ExpiredMedicine\_CRUD

{

public static List<ExpiredMedicine> expiredmed = new List<ExpiredMedicine>();

public static void addExpiredmedicine()

{

string path = "path\\ExpiredMedicine.txt";

StreamWriter expirefile = new StreamWriter(path);

for (int x = 0; x < expiredmed.Count; x++)

{

expirefile.WriteLine(expiredmed[x].getExpiredmedicine());

}

expirefile.Flush();

expirefile.Close();

}

}

**Muser Forms code:** Main Form.cs

public partial class HMSApplication : Form

{

public HMSApplication()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

Form newform = new SignUp\_Form();

newform.Show();

this.Hide();

}

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

{

Form newForm = new SignIn\_Form();

newForm.Show();

this.Hide();

}

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

{

}

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

{

button1.BackColor = Color.Black;

button1.ForeColor = Color.White;

}

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

{

button1.BackColor = Color.FromArgb(128, 255, 255);

button1.ForeColor = Color.Black;

}

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

{

button2.BackColor = Color.Black;

button2.ForeColor = Color.White;

}

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

{

button2.BackColor = Color.FromArgb(128, 255, 255);

button2.ForeColor = Color.Black;

}

}

**Muser Forms code:** SignUp Form.cs

public partial class SignUp\_Form : Form

{

public SignUp\_Form()

{

InitializeComponent();

}

private void ClearDataFromForm()

{

txtusername.Text = "";

txtpassword.Text = "";

txtrole.Text = "";

}

private void SignUp\_Form\_Load(object sender, EventArgs e)

{

}

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

{

string username = txtusername.Text;

string password = txtpassword.Text;

string role = txtrole.Text;

string path = "data.txt";

MUser user = new MUser(username, password, role);

MUser\_CRUD.addUserIntoList(user);

MUser\_CRUD.storeIntoFile(path, user);

MessageBox.Show("User Added Successfully");

ClearDataFromForm();

}

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

{

Form newform = new HMSApplication();

newform.Show();

this.Hide();

}

private void chkboxshowpasscode\_CheckedChanged(object sender, EventArgs e)

{

if (chkboxshowpasscode.Checked)

{

txtpassword.UseSystemPasswordChar = true;

}

else

{

txtpassword.UseSystemPasswordChar = false;

}

}

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

{

button2.BackColor = Color.Black;

button2.ForeColor = Color.White;

}

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

{

button2.BackColor = Color.FromArgb(128, 255, 255);

button2.ForeColor = Color.Black;

}

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

{

button1.BackColor = Color.Black;

button1.ForeColor = Color.White;

}

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

{

button1.BackColor = Color.FromArgb(128, 255, 255);

button1.ForeColor = Color.Black;

}

}

**Muser Forms code:** SignIn Form.cs

public partial class SignIn\_Form : Form

{

public SignIn\_Form()

{

InitializeComponent();

}

private void ClearDataFromForm()

{

txtusername.Text = "";

txtpassword.Text = "";

}

private void SignIn\_Form\_Load(object sender, EventArgs e)

{

}

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

{

string username = txtusername.Text;

string password = txtpassword.Text;

MUser user = new MUser(username, password);

MUser validUser = MUser\_CRUD.SignIn(user);

if (validUser != null)

{

if (validUser.UserRole == "Admin" || validUser.UserRole == "admin")

{

Form newform = new Admin\_menu\_Form();

newform.Show();

}else if (validUser.UserRole == "patient" || validUser.UserRole == "Patient")

{

Form newform = new Patient\_menu\_form();

newform.Show();

}else if(validUser.UserRole == "Pharmacist" || validUser.UserRole == "pharmacist") {

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

}

this.Hide();

}

else

{

MessageBox.Show("User is Invalid");

}

ClearDataFromForm();

}

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

{

Form newform = new HMSApplication();

newform.Show();

this.Hide();

}

private void chkboxShowPasscode\_CheckedChanged(object sender, EventArgs e)

{

if (chkboxShowPasscode.Checked)

{

txtpassword.UseSystemPasswordChar = true;

}

else

{

txtpassword.UseSystemPasswordChar = false;

}

}

private void Back\_MouseEnter(object sender, EventArgs e)

{

Back.BackColor = Color.Black;

Back.ForeColor = Color.White;

}

private void Back\_MouseLeave(object sender, EventArgs e)

{

Back.BackColor = Color.FromArgb(128, 255, 255);

Back.ForeColor = Color.Black;

}

private void SignInbtn\_MouseEnter(object sender, EventArgs e)

{

SignInbtn.BackColor = Color.Black;

SignInbtn.ForeColor = Color.White;

}

private void SignInbtn\_MouseLeave(object sender, EventArgs e)

{

SignInbtn.BackColor = Color.FromArgb(128, 255, 255);

SignInbtn.ForeColor = Color.Black;

}

}

**Patient Forms code:** Add Patient.cs

public partial class Add\_Patient\_Form : Form

{

public Add\_Patient\_Form()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

string patient\_name = txtpatient.Text;

int age = int.Parse(txtage.Text);

string phoneno = txtphone.Text;

string cnic = txtcnicno.Text;

string address = txtaddress.Text;

DateTime date = dtvisitdate.Value;

string datevisit = Convert.ToDateTime(date).Date.ToString("d");

string medicalHistory = txtmedicalhistory.Text;

Patient p = new Patient();

p.setName(patient\_name);

p.setAge(age);

p.setContactno(phoneno);

p.setCnic(cnic);

p.setAddress(address);

p.setVisitdate(datevisit);

p.setMedicalhistory(medicalHistory);

ADD\_Patient(p);

}

private void ADD\_Patient(Patient P)

{

bool check = PatientCRUD.addPatient\_Dl(P);

if (check == false)

{

MessageBox.Show("This patient record is already exist");

Empty();

}

else

{

PatientCRUD.Add\_Patient(P);

PatientCRUD.save\_patientdata(P);

MessageBox.Show("The Patient is Added Successfully and Please give proper info by choosing other options for single Patient before adding next Patient");

this.Close();

Form newform = new Admin\_menu\_Form();

newform.Show();

}

}

private void Empty()

{

txtpatient.Text = String.Empty;

txtage.Text = String.Empty;

txtcnicno.Text = String.Empty;

txtphone.Text = String.Empty;

txtaddress.Text = String.Empty;

dtvisitdate.Value = DateTime.Now;

}

private void Add\_Patient\_Form\_Load(object sender, EventArgs e)

{

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnAdd\_MouseEnter(object sender, EventArgs e)

{

btnAdd.BackColor = Color.Black;

btnAdd.ForeColor = Color.White;

}

private void btnAdd\_MouseLeave(object sender, EventArgs e)

{

btnAdd.BackColor=Color.FromArgb(128, 255, 255);

btnAdd.ForeColor=Color.Black;

}

}

**Patient Forms code:** Update Patient.cs

public partial class Update\_Patient\_Form : Form

{

Patient p1;

DataTable table = new DataTable("UpdatePatient");

public Update\_Patient\_Form()

{

InitializeComponent();

}

private void Update\_Patient\_Form\_Load(object sender, EventArgs e)

{

}

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

{

string patientName = txtPatientUpdatename.Text;

string cnicNO = txtCnicnoUpdate.Text;

UPDATE\_Patient\_Check(ref patientName,ref cnicNO);

}

private void UPDATE\_Patient\_Check(ref string patientName,ref string cnicNO)

{

p1 = PatientCRUD.update\_patient\_Dl(patientName, cnicNO);

ifPatientnotExist(p1);

}

private void PatientColumnsHeading()

{

table.Columns.Add("Patient name", Type.GetType("System.String"));

table.Columns.Add("Age", Type.GetType("System.String"));

table.Columns.Add("Phone no", Type.GetType("System.String"));

table.Columns.Add("Cnic", Type.GetType("System.String"));

table.Columns.Add("Address", Type.GetType("System.String"));

}

private void ifPatientnotExist(Patient p)

{

if (p != null)

{

PatientColumnsHeading();

table.Rows.Add(p.getName(), p.getAge(), p.getContactno(), p.getCnic(), p.getAddress());

UpdatePatientGV.DataSource = table;

txtPatientname.Text=p.getName();

txtage.Text=""+p.getAge();

txtPhoneno.Text=p.getContactno();

txtCnicno.Text=p.getCnic();

string date = p.getVisitdate();

dtvisitdate.Value=DateTime.Parse(date);

txtAddress.Text=p.getAddress();

txtmedicalHistory.Text=p.getMedicalhistory();

/\* Admin\_interface.updatepatient\_after(p);

PatientCRUD.update\_patientdata();\*/

}

if(p == null)

{

MessageBox.Show("The patient is not exist in our database");

empty();

}

}

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

{

string name = txtPatientname.Text;

int age = int.Parse(txtage.Text);

string phoneNo = txtPhoneno.Text;

string cnic = txtCnicno.Text;

DateTime date = dtvisitdate.Value;

string datevisit = Convert.ToDateTime(date).Date.ToString("d");

string Address = txtAddress.Text;

string MedicalHistory = txtmedicalHistory.Text;

p1.setName(name);

p1.setAge(age);

p1.setContactno(phoneNo);

p1.setCnic(cnic);

p1.setVisitdate(datevisit);

p1.setAddress(Address);

p1.setMedicalhistory(MedicalHistory);

UpdateMessage();

empty();

}

private void UpdateMessage()

{

MessageBox.Show("Selected Patient data is Updated");

}

private void empty()

{

txtPatientUpdatename.Text=String.Empty;

txtCnicnoUpdate.Text=String.Empty;

txtPatientname.Text = String.Empty;

txtAddress.Text = String.Empty;

txtage.Text = String.Empty;

txtCnicno.Text = String.Empty;

txtmedicalHistory.Text = String.Empty;

txtPhoneno.Text = String.Empty;

dtvisitdate.Value = DateTime.Now;

UpdatePatientGV.DataSource = null;

table.Columns.Clear();

table.Rows.Clear();

}

private void HometoolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnCheck\_MouseEnter(object sender, EventArgs e)

{

btnCheck.BackColor = Color.Black;

btnCheck.ForeColor = Color.White;

}

private void btnCheck\_MouseLeave(object sender, EventArgs e)

{

btnCheck.BackColor = Color.FromArgb(128, 255, 255);

btnCheck.ForeColor = Color.Black;

}

private void btnUpdate\_MouseEnter(object sender, EventArgs e)

{

btnUpdate.BackColor = Color.Black;

btnUpdate.ForeColor = Color.White;

}

private void btnUpdate\_MouseLeave(object sender, EventArgs e)

{

btnUpdate.BackColor = Color.FromArgb(128, 255, 255);

btnUpdate.ForeColor = Color.Black;

}

}

**Patient Forms code:** View Patient.cs

public partial class View\_Patient\_form : Form

{

private string path = "path\\Patient.txt";

DataTable table = new DataTable("PatientTable");

public View\_Patient\_form()

{

InitializeComponent();

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

}

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

{

}

private void View\_Patient\_form\_Load(object sender, EventArgs e)

{

table.Columns.Add("Patientname", Type.GetType("System.String"));

table.Columns.Add("Age", Type.GetType("System.String"));

table.Columns.Add("Phone no", Type.GetType("System.String"));

table.Columns.Add("Cnic", Type.GetType("System.String"));

table.Columns.Add("Address", Type.GetType("System.String"));

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

{

Patient p = PatientCRUD.patients[i];

table.Rows.Add(p.getName(),p.getAge(),p.getContactno(),p.getCnic(),p.getAddress());

}

PatientGV.DataSource = table;

}

private void HometoolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

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

{

DataView dvPatientTable = table.DefaultView;

dvPatientTable.RowFilter = "Patientname Like '%" + txtSearchBox.Text + "%'";

}

private void txtSearchBox\_Enter(object sender, EventArgs e)

{

txtCnicSearch.Text = "Search By Cnic";

txtCnicSearch.ForeColor = Color.DimGray;

if (txtSearchBox.Text=="Search By Name")

{

txtSearchBox.Text = "";

txtSearchBox.ForeColor =Color.Black;

}

}

private void txtSearchBox\_Leave(object sender, EventArgs e)

{

if (txtSearchBox.Text == "")

{

txtSearchBox.Text = "Search By Name";

txtSearchBox.ForeColor = Color.DimGray;

}

}

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

{

DataView dvPatientTable1 = table.DefaultView;

dvPatientTable1.RowFilter = "Cnic Like '%" + txtCnicSearch.Text + "%'";

}

private void txtCnicSearch\_Enter(object sender, EventArgs e)

{

txtSearchBox.Text = "Search By Name";

txtSearchBox.ForeColor = Color.DimGray;

if (txtCnicSearch.Text == "Search By Cnic")

{

txtCnicSearch.Text = "";

txtCnicSearch.ForeColor = Color.Black;

}

}

private void txtCnicSearch\_Leave(object sender, EventArgs e)

{

if (txtCnicSearch.Text == "")

{

txtCnicSearch.Text = "Search By Cnic";

txtCnicSearch.ForeColor = Color.DimGray;

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

}

**Patient forms code:** Discharge Patient.cs

public partial class Discharge\_Patient\_Form : Form

{

public Discharge\_Patient\_Form()

{

InitializeComponent();

}

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

{

string discharge\_patient = txtname.Text;

string dischargepatient\_cnic = txtCnic.Text;

bool discharge\_check = PatientCRUD.dischargePatient(discharge\_patient, dischargepatient\_cnic);

if (discharge\_check == true)

{

MessageBox.Show("The Patient is discharge successfully");

PatientCRUD.dataafter\_deletefile();

}

else

{

MessageBox.Show("The patient does not exist in our database");

}

empty();

}

private void empty()

{

txtname.Text = null;

txtCnic.Text = null;

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnDelete\_MouseEnter(object sender, EventArgs e)

{

btnDelete.BackColor = Color.Black;

btnDelete.ForeColor = Color.White;

}

private void btnDelete\_MouseLeave(object sender, EventArgs e)

{

btnDelete.BackColor = Color.FromArgb(128, 255, 255);

btnDelete.ForeColor = Color.Black;

}

}

**Admin Forms code:** Suggested Medicine.cs

public partial class Suggest\_Medicine\_Form : Form

{

string[] symptoms = { " ", " ", " ", " ", " ", " " };

char check1 = ' ', check2 = ' ', check3 = ' ';

Patient P = PatientCRUD.patients[PatientCRUD.patients.Count - 1];

int count = 0;

public Suggest\_Medicine\_Form()

{

InitializeComponent();

}

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

{

string Medicine = P.GiveMedicine(check1, check2, check3);

MessageBox.Show("Suggested medicines for you is "+Medicine);

this.Close();

Form newform = new Admin\_menu\_Form();

newform.Show();

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnAdd\_MouseEnter(object sender, EventArgs e)

{

btnAdd.BackColor = Color.Black;

btnAdd.ForeColor = Color.White;

}

private void btnAdd\_MouseLeave(object sender, EventArgs e)

{

btnAdd.BackColor = Color.FromArgb(128, 255, 255);

btnAdd.ForeColor = Color.Black;

}

private void btnNext\_MouseEnter(object sender, EventArgs e)

{

btnNext.BackColor = Color.Black;

btnNext.ForeColor = Color.White;

}

private void btnNext\_MouseLeave(object sender, EventArgs e)

{

btnNext.BackColor = Color.FromArgb(128, 255, 255);

btnNext.ForeColor = Color.Black;

}

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

{

if (count < 6)

{

symptoms[count] = txtSymptom.Text;

P.SymptomsCheck(symptoms, count, ref check1, ref check2, ref check3);

count++;

empty();

}

else

{

MessageBox.Show("You have entered all the symptoms");

}

}

private void empty()

{

txtSymptom.Text = String.Empty;

}

}

**Admin Forms code:** Generate Report.cs

public partial class Generate\_medical\_Report\_Form : Form

{

Patient p = PatientCRUD.patients[PatientCRUD.patients.Count - 1];

public Generate\_medical\_Report\_Form()

{

InitializeComponent();

}

private void label3\_Click(object sender, EventArgs e)

{

}

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

{

string LabTestName = txtLabtestName.Text;

int cholestrolLevel = int.Parse(txtCholestrolLevel.Text);

string Typhoid = txtTyphoid.Text;

string Malaria = txtMalaria.Text;

string Anemia = txtAnemia.Text;

string DiagnosticTestname = txtDiagnosticTest.Text;

p.PatientReport.setLaboratorytest(LabTestName);

if (LabTestName == "blood test")

{

p.PatientReport.setCholestrolLevel(cholestrolLevel);

p.PatientReport.Generatemedicalreport();

p.PatientReport.setTyphoid(Typhoid);

p.PatientReport.setMalaria(Malaria);

p.PatientReport.setAnemia(Anemia);

}

p.PatientReport.setDiagnostictest(DiagnosticTestname);

PatientCRUD.add\_medicalReport(p);

MessageBox.Show("Report is generated successfully");

this.Close();

Form newform = new Admin\_menu\_Form();

newform.Show();

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnNext\_MouseEnter(object sender, EventArgs e)

{

btnNext.BackColor = Color.Black;

btnNext.ForeColor = Color.White;

}

private void btnNext\_MouseLeave(object sender, EventArgs e)

{

btnNext.BackColor = Color.FromArgb(128, 255, 255);

btnNext.ForeColor = Color.Black;

}

}

**Admin Forms code:** Billing/Invoicing.cs

public partial class Billing\_Form : Form

{

int count = 1;

int firstCharge = 0;

int secondCharge = 0;

Patient P = PatientCRUD.patients[PatientCRUD.patients.Count - 1];

PatientBill bill = PatientCRUD.patients[PatientCRUD.patients.Count-1].bill;

public Billing\_Form()

{

InitializeComponent();

}

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

{

double totalpricemedicine = 0F;

int days = Convert.ToInt32(Math.Round(numpatientdaysinhospital.Value, 0));

bill.setDayspatientstayed(days);

for (int i = 0; i < Customer\_CRUD.customers.Count; i++)

{

if (P.getName() == Customer\_CRUD.customers[i].getcustomerName())

{

totalpricemedicine = Customer\_CRUD.customers[i].Calculate\_Bill();

break;

}

}

int bedCharge = Convert.ToInt32(Math.Round(numunitdaycharge.Value, 0));

bill.setBedcharge(bedCharge);

int labtest = Convert.ToInt32(Math.Round(numlabbill.Value, 0)); ;

bill.setLabtest(labtest);

bill.calculate\_totalBill(firstCharge, secondCharge, bill.getBedcharge(), totalpricemedicine);

double totalbill = bill.getTotalbill();

txttotalBill.Text = ""+totalbill;

PatientCRUD.add\_billing(P);

}

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

{

string service = txtService.Text;

if (count < 3)

{

if (count == 1)

{

bill.setServices(service);

firstCharge = bill.ServiceCheck();

}

if (count == 2)

{

if (service != bill.getServices())

{

bill.setServices(service);

secondCharge = bill.ServiceCheck();

}

else

{

MessageBox.Show("You Already added this service");

count -= 1;

}

}

}

else { MessageBox.Show("You have added all available services NO MORE"); }

count++;

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnAdd\_MouseEnter(object sender, EventArgs e)

{

btnAdd.BackColor = Color.Black;

btnAdd.ForeColor = Color.White;

}

private void btnAdd\_MouseLeave(object sender, EventArgs e)

{

btnAdd.BackColor = Color.FromArgb(128, 255, 255);

btnAdd.ForeColor = Color.Black;

}

private void btnCalculatebill\_MouseEnter(object sender, EventArgs e)

{

btnCalculatebill.BackColor = Color.Black;

btnCalculatebill.ForeColor = Color.White;

}

private void btnCalculatebill\_MouseLeave(object sender, EventArgs e)

{

btnCalculatebill.BackColor = Color.FromArgb(128, 255, 255);

btnCalculatebill.ForeColor = Color.Black;

}

}

**Doctor Forms code:** Set DoctorSchedule.cs

public partial class Manage\_DoctorSchedule\_Form : Form

{

public Manage\_DoctorSchedule\_Form()

{

InitializeComponent();

}

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

{

DateTime date = dtDoctordate.Value;

string Doctordate= Convert.ToDateTime(date).Date.ToString("d");

string DoctorName = txtDoctorname.Text;

string profession = txtDoctorSpecialty.Text;

string startingTime = txtStartingTime.Text;

string endingTime = txtEndingTime.Text;

Doctor d = new Doctor();

d.setScheduledate(Doctordate);

d.setDoctorname(DoctorName);

d.setProfession(profession);

d.setStartingtime(startingTime);

d.setEndingtime(endingTime);

Doctor\_CRUD.Add\_Doctor(d);

Doctor\_CRUD.add\_DoctorSchedule(d);

ShowDialogbox();

}

private void ShowDialogbox()

{

DialogResult res = MessageBox.Show("Do you want to add another Doctor Schedule", "Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (res == DialogResult.Yes)

{

empty();

}

else

{

this.Close();

}

}

private void empty()

{

txtDoctorname.Text = String.Empty;

txtDoctorSpecialty.Text = String.Empty;

txtEndingTime.Text = String.Empty;

txtStartingTime.Text = String.Empty;

dtDoctordate.Value = DateTime.Now;

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnAdd\_MouseEnter(object sender, EventArgs e)

{

btnAdd.BackColor = Color.Black;

btnAdd.ForeColor = Color.White;

}

private void btnAdd\_MouseLeave(object sender, EventArgs e)

{

btnAdd.BackColor = Color.FromArgb(128, 255, 255);

btnAdd.ForeColor = Color.Black;

}

}

**Admin Forms code:** Manage Doctor Appointment.cs

public partial class ManageAppointments\_form : Form

{

private DataTable appointmentTable = new DataTable("AppointmentTable");

public ManageAppointments\_form()

{

InitializeComponent();

}

private void ManageAppointments\_form\_Load(object sender, EventArgs e)

{

TableColumnsHeading();

ShowAppointments();

table\_fun();

}

public void TableColumnsHeading()

{

appointmentTable.Columns.Clear();

appointmentTable.Rows.Clear();

appointmentTable.Columns.Add("Patient name", Type.GetType("System.String"));

appointmentTable.Columns.Add("Appointment Date", Type.GetType("System.String"));

appointmentTable.Columns.Add("Appointment Doctor", Type.GetType("System.String"));

appointmentTable.Columns.Add("Appointment Time", Type.GetType("System.String"));

}

public void ShowAppointments()

{

for (int i = 0; i < Appointment\_CRUD.appointments.Count; i++)

{

Appointment appointment = Appointment\_CRUD.appointments[i];

appointmentTable.Rows.Add(appointment.getName(), appointment.getDateappoint(), appointment.getDoctor(), appointment.getTimeappoint());

}

}

public void table\_fun()

{

AppointmentsGV.DataSource = appointmentTable;

}

public static void t()

{

ManageAppointments\_form appointments\_Form = new ManageAppointments\_form();

appointments\_Form.table\_fun();

}

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

{

Form newform = new Patient\_Form.ShiftAppointment\_Form();

newform.Tag = this;

newform.Show();

}

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

{

Form newform = new Patient\_Form.CancelAppointment\_Form();

newform.Tag = this;

newform.Show();

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnShiftAppointment\_MouseEnter(object sender, EventArgs e)

{

btnShiftAppointment.BackColor = Color.Black;

btnShiftAppointment.ForeColor = Color.White;

}

private void btnShiftAppointment\_MouseLeave(object sender, EventArgs e)

{

btnShiftAppointment.BackColor = Color.FromArgb(128, 255, 255);

btnShiftAppointment.ForeColor = Color.Black;

}

private void btnCancelAppointment\_MouseEnter(object sender, EventArgs e)

{

btnCancelAppointment.BackColor = Color.Black;

btnCancelAppointment.ForeColor = Color.White;

}

private void btnCancelAppointment\_MouseLeave(object sender, EventArgs e)

{

btnCancelAppointment.BackColor = Color.FromArgb(128, 255, 255);

btnCancelAppointment.ForeColor = Color.Black;

}

}

**Shift Appointment form:**

public partial class ShiftAppointment\_Form : Form

{

public ShiftAppointment\_Form()

{

InitializeComponent();

}

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

{

string ShiftedPatientname = txtname.Text;

DateTime date = dtDateAppointment.Value;

string shiftdate = Convert.ToDateTime(date).Date.ToString("d");

string shiftTime = txttime.Text;

bool exist = Appointment\_CRUD.shiftedappointment(ref ShiftedPatientname, ref shiftdate, ref shiftTime);

if (exist == true) { MessageBox.Show("Selected Appointment is Shifted Successfully"); }

else { MessageBox.Show("This Patient has no appointment write correct patient"); }

Appointment\_CRUD.DATAaftershiftedORcancelappointment();

ManageAppointments\_form nform = this.Tag as ManageAppointments\_form;

nform.TableColumnsHeading();

nform.ShowAppointments();

this.Close();

}

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

{

Environment.Exit(1);

}

private void btnShift\_MouseEnter(object sender, EventArgs e)

{

btnShift.BackColor = Color.Black;

btnShift.ForeColor = Color.White;

}

private void btnShift\_MouseLeave(object sender, EventArgs e)

{

btnShift.BackColor = Color.FromArgb(128, 255, 255);

btnShift.ForeColor = Color.Black;

}

}

**Cancel Appointment Form:**

public partial class CancelAppointment\_Form : Form

{

public CancelAppointment\_Form()

{

InitializeComponent();

}

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

{

string cancelpatient = txtCancelAppointmentPatient.Text;

bool exist = Appointment\_CRUD.canceledappointment(ref cancelpatient);

if (exist == true) { MessageBox.Show("The Appointment of this patient is Canceled"); }

else { MessageBox.Show("This Patient has no Appointment"); }

Appointment\_CRUD.cleanFile();

Appointment\_CRUD.DATAaftershiftedORcancelappointment();

ManageAppointments\_form nform = this.Tag as ManageAppointments\_form;

nform.TableColumnsHeading();

nform.ShowAppointments();

this.Close();

}

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

{

Environment.Exit(1);

}

private void btnCancel\_MouseEnter(object sender, EventArgs e)

{

btnCancel.BackColor = Color.Black;

btnCancel.ForeColor = Color.White;

}

private void btnCancel\_MouseLeave(object sender, EventArgs e)

{

btnCancel.BackColor = Color.FromArgb(128, 255, 255);

btnCancel.ForeColor = Color.Black;

}

}

**Admin Forms code:** Check Doctor Appointment.cs

public partial class Check\_Doctor\_Appointment\_Form : Form

{

DataTable Doctors = new DataTable("DoctorTable");

public Check\_Doctor\_Appointment\_Form()

{

InitializeComponent();

}

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

{

DateTime date = dtDate.Value;

string Date = Convert.ToDateTime(date).Date.ToString("d");

string Doctorname = txtDoctorname.Text;

TablecolumnsHeading();

foreach (var doctor in Doctor\_CRUD.doctors)

{

if (Doctorname == doctor.getDoctorname() && Date == doctor.getScheduledate())

{

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

{

Appointment a = doctor.myAppointments[i];

Doctors.Rows.Add(a.getName(), a.getTimeappoint());

}

break;

}

}

DoctorsGV.DataSource = Doctors;

}

private void TablecolumnsHeading()

{

Doctors.Columns.Add("Patient name", Type.GetType("System.String"));

Doctors.Columns.Add("Appointment Time", Type.GetType("System.String"));

}

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

{

Form newform = new Admin\_menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnCheck\_MouseEnter(object sender, EventArgs e)

{

btnCheck.BackColor = Color.Black;

btnCheck.ForeColor = Color.White;

}

private void btnCheck\_MouseLeave(object sender, EventArgs e)

{

btnCheck.BackColor = Color.FromArgb(128, 255, 255);

btnCheck.ForeColor = Color.Black;

}

}

**Individual PatientData Forms code:** Detail.cs

public partial class DetailForm : Form

{

public static string Patientname;

public static string patientCnic;

public DetailForm()

{

InitializeComponent();

}

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

{

Patientname = txtpatient.Text;

patientCnic = txtcnicno.Text;

bool checkPatient = false;

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

{

if (patientCnic == PatientCRUD.patients[i].getCnic())

{

checkPatient = true;

break;

}

else { checkPatient = false; }

}

if(checkPatient == true) { this.Close(); Form newform = new Patient\_menu\_form(); newform.Show();}

if(checkPatient == false) { MessageBox.Show("This patient is not store in our database"); empty(); }

}

private void empty()

{

txtcnicno.Text = null;

txtpatient.Text = null;

}

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

{

Form newform = new Patient\_Form.Patient\_menu\_form();

newform.Show();

this.Hide();

}

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

{

Form newform = new Patient\_menu\_form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnnext\_MouseEnter(object sender, EventArgs e)

{

btnnext.BackColor = Color.Black;

btnnext.ForeColor = Color.White;

}

private void btnnext\_MouseLeave(object sender, EventArgs e)

{

btnnext.BackColor = Color.FromArgb(128, 255, 128);

btnnext.ForeColor = Color.Black;

}

}

**Individual PatientData Forms code:** View Prescription.cs

public partial class ViewPrescriptionForm : Form

{

public ViewPrescriptionForm()

{

InitializeComponent();

}

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

{

}

private void ViewPrescriptionForm\_Load\_1(object sender, EventArgs e)

{

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

{

if (DetailForm.patientCnic == PatientCRUD.patients[i].getCnic())

{

lblprescription.Text = PatientCRUD.patients[i].getMedicine();

break;

}

}

}

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

{

this.Close();

}

}

**Individual PatientData Forms code:** View MedicalReport.cs

public partial class ViewMedicalReportForm : Form

{

public ViewMedicalReportForm()

{

InitializeComponent();

}

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

{

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

{

if (DetailForm.patientCnic == PatientCRUD.patients[i].getCnic())

{

if (PatientCRUD.patients[i].PatientReport.getLaboratorytest() == "blood test")

{

ShowLABTest(i);

}

Show\_medical\_history(i);

Show\_Diagnostic\_test(i);

}

}

}

private void ShowLABTest(int i)

{

lblCholestrolLevel.Text = ""+PatientCRUD.patients[i].PatientReport.getCholestrolLevel();

lblCholestrol.Text = PatientCRUD.patients[i].PatientReport.getCholestrol();

lbltyphoid.Text = PatientCRUD.patients[i].PatientReport.getTyphoid();

lblmalaria.Text = PatientCRUD.patients[i].PatientReport.getMalaria();

lblanemia.Text = PatientCRUD.patients[i].PatientReport.getAnemia();

}

private void Show\_medical\_history(int i)

{

lblMedicalHistory.Text = PatientCRUD.patients[i].getMedicalhistory();

}

private void Show\_Diagnostic\_test(int i)

{

lbldiagnostic.Text = PatientCRUD.patients[i].PatientReport.getDiagnostictest();

lblnormal.Text = "normal";

}

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

{

this.Close();

}

}

**Individual PatientData Forms code:** Check DoctorSchedule.cs

public partial class CheckDoctorSchedule\_Form : Form

{

DataTable doctors = new DataTable();

public CheckDoctorSchedule\_Form()

{

InitializeComponent();

}

private void CheckDoctorSchedule\_Form\_Load(object sender, EventArgs e)

{

}

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

{

doctors.Columns.Clear();

doctors.Rows.Clear();

DateTime date = dtdate.Value;

string dateavailability = Convert.ToDateTime(date).Date.ToString("d");

doctors.Columns.Add("Doctor Name", Type.GetType("System.String"));

doctors.Columns.Add("Doctor Specialty", Type.GetType("System.String"));

doctors.Columns.Add("Starting Time", Type.GetType("System.String"));

doctors.Columns.Add("Ending Time", Type.GetType("System.String"));

foreach (var doctor in Doctor\_CRUD.doctors)

{

if (dateavailability == doctor.getScheduledate())

{

doctors.Rows.Add(doctor.getDoctorname(), doctor.getProfession(), doctor.getStartingtime(), doctor.getEndingtime());

}

}

doctorGV.DataSource = doctors;

}

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

{

this.Close();

}

private void btnCheck\_MouseEnter(object sender, EventArgs e)

{

btnCheck.BackColor = Color.Black;

btnCheck.ForeColor = Color.White;

}

private void btnCheck\_MouseLeave(object sender, EventArgs e)

{

btnCheck.BackColor = Color.FromArgb(128, 255, 128);

btnCheck.ForeColor = Color.Black;

}

}

**Individual PatientData Forms code:** Check Bill.cs

public partial class CheckBill\_Form : Form

{

public CheckBill\_Form()

{

InitializeComponent();

}

private void CheckBill\_Form\_Load(object sender, EventArgs e)

{

double totalpricemedicine = 0F;

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

{

if (DetailForm.Patientname == PatientCRUD.patients[i].getName())

{

ShowRoomCharges(i);

for (int j = 0; j < Customer\_CRUD.customers.Count; j++)

{

if (PatientCRUD.patients[i].getName() == Customer\_CRUD.customers[j].getcustomerName())

{

totalpricemedicine = Customer\_CRUD.customers[j].Calculate\_Bill();

break;

}

}

Showbills(totalpricemedicine, i);

break;

}

}

}

private void ShowRoomCharges(int i)

{

lblroomcharge.Text=""+PatientCRUD.patients[i].bill.getBedcharge() \* PatientCRUD.patients[i].bill.getDaysPatientstayed();

}

private void Showbills(double totalpricemedicine, int i)

{

lblmedicinebill.Text=""+totalpricemedicine;

lblLabtestbill.Text =""+PatientCRUD.patients[i].bill.getLabtest();

lbltotalbill.Text=""+PatientCRUD.patients[i].bill.getTotalbill();

}

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

{

this.Close();

}

}

**Patient Forms code:** Book Appointment.cs

public partial class BookAppointmentForm : Form

{

Appointment appointment;

bool check = false;

int timeSt = 0;

int timeEnd = 0;

public BookAppointmentForm()

{

InitializeComponent();

}

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

{

DateTime dateAppoint = dtDate.Value;

string date = Convert.ToDateTime(dateAppoint).Date.ToString("d");

string doctorname = txtdoctorName.Text;

string name = txtname.Text;

appointment = new Appointment(name,date,doctorname);

for (int i = 0; i < Doctor\_CRUD.doctors.Count; i++)

{

if (appointment.getDateappoint() == Doctor\_CRUD.doctors[i].getScheduledate() && appointment.getDoctor() == Doctor\_CRUD.doctors[i].getDoctorname())

{

lblavailableTime.Text = "This doctor is available from: " + Doctor\_CRUD.doctors[i].getStartingtime() + " to " + Doctor\_CRUD.doctors[i].getEndingtime();

string startingtime\_final = Doctor\_CRUD.doctors[i].getStartingtime();

string endingtime\_final = Doctor\_CRUD.doctors[i].getEndingtime();

timeSt = appointment.Converttime(startingtime\_final);

timeEnd = appointment.Converttime(endingtime\_final);

check = true;

break;

}

else

{

check = false;

}

}

bookAppointment\_DoctorAvailable(check);

}

private void bookAppointment\_DoctorAvailable(bool check)

{

if (check == false)

{

MessageBox.Show("This doctor is not available");

empty();

}

}

private void empty()

{

txtdoctorName.Text = null;

txtname.Text = null;

dtDate.Value = DateTime.Now;

}

private void Fullclean()

{

empty();

lblavailableTime.Text = null;

txttimetakeAppointment.Text = null;

}

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

{

if (check == true)

{

string takeAppointment = txttimetakeAppointment.Text;

appointment.setTimeappoint(takeAppointment);

int time = appointment.Converttime(appointment.getTimeappoint());

if (time >= timeSt && time <= timeEnd)

{

MessageBox.Show("Appointment set");

Appointment\_CRUD.add\_appointmentinFile(appointment);

Appointment\_CRUD.Add\_Appointment(appointment);

Doctor\_CRUD.put\_DoctorAppointment\_inList(Appointment\_CRUD.appointments);

}

else

{

MessageBox.Show("This doctor is not available");

}

}

else { MessageBox.Show("We can't book this Appointment"); }

this.Close();

}

private void btncheck\_MouseEnter(object sender, EventArgs e)

{

btncheck.BackColor = Color.Black;

btncheck.ForeColor = Color.White;

}

private void btncheck\_MouseLeave(object sender, EventArgs e)

{

btncheck.BackColor = Color.FromArgb(128, 255, 128);

btncheck.ForeColor = Color.Black;

}

private void btnbookappointment\_MouseEnter(object sender, EventArgs e)

{

btnbookappointment.BackColor = Color.Black;

btnbookappointment.ForeColor = Color.White;

}

private void btnbookappointment\_MouseLeave(object sender, EventArgs e)

{

btnbookappointment.BackColor = Color.FromArgb(128, 255, 128);

btnbookappointment.ForeColor = Color.Black;

}

}

**Pharmacy Forms code:** Add medicine.cs

public partial class AddMedicine\_Form : Form

{

public AddMedicine\_Form()

{

InitializeComponent();

}

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

{

string MedicineName = txtMedicineName.Text;

DateTime date = dtExpiryDate.Value;

string Expirydate = Convert.ToDateTime(date).Date.ToString("d");

int quantity = Convert.ToInt32(Math.Round(numQuantity.Value, 0));

double price = Convert.ToDouble(Math.Round(numPrice.Value, 0));

int demandcheck = quantity;

Medicine med1 = new Medicine(MedicineName, quantity, Expirydate, price, demandcheck);

Medicine\_CRUD.Addmedicine(med1);

Medicine\_CRUD.addMedicineinFile();

empty();

}

private void empty()

{

txtMedicineName.Text = null;

numQuantity.Value = 0;

numPrice.Value = 0;

dtExpiryDate.Value = DateTime.Now;

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnAdd\_MouseEnter(object sender, EventArgs e)

{

btnAdd.BackColor = Color.Black;

btnAdd.ForeColor = Color.White;

}

private void btnAdd\_MouseLeave(object sender, EventArgs e)

{

btnAdd.BackColor = Color.FromArgb(128, 255, 128);

btnAdd.ForeColor = Color.Black;

}

}

**Pharmacy Forms code:** Update Medicine.cs

public partial class UpdateMedicine\_Form : Form

{

Medicine medicine;

string change;

public UpdateMedicine\_Form()

{

InitializeComponent();

}

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

{

string medicineName = txtMedicine.Text;

medicine = Medicine\_CRUD.Updatemedicine\_Check(medicineName);

if (medicine == null)

{

MessageBox.Show("This medicine is not in our database");

txtMedicine.Text = null;

}

else { MessageBox.Show("Medicine found"); }

}

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

{

change = txtchange.Text;

if (change == "medicine name" || change == "quantity" || change == "Quantity" || change == "expiry date" || change == "price" || change == "Prize")

{

lbldepend.Text = change + ":";

}

else { MessageBox.Show("Enter valid change"); txtchange.Text = null; }

}

private void empty()

{

txtchange.Text = null;

txtMedicine.Text = null;

txtupdatebox.Text = null;

}

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

{

if (change == "medicine name")

{

medicine.setMedicinename(txtupdatebox.Text);

}

if (change == "quantity" || change == "Quantity")

{

medicine.setQuantity(int.Parse(txtupdatebox.Text));

}

if (change == "expiry date")

{

medicine.setExpirydate(txtupdatebox.Text);

}

if (change == "price" || change == "Prize")

{

medicine.setPrice(double.Parse(txtupdatebox.Text));

}

Medicine\_CRUD.saveUpdatedmedicine\_data();

empty();

}

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

{

this.Close();

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnCheck\_MouseEnter(object sender, EventArgs e)

{

btnCheck.BackColor = Color.Black;

btnCheck.ForeColor = Color.White;

}

private void btnCheck\_MouseLeave(object sender, EventArgs e)

{

btnCheck.BackColor = Color.FromArgb(128, 255, 128);

btnCheck.ForeColor = Color.Black;

}

private void btncontinue\_MouseEnter(object sender, EventArgs e)

{

btncontinue.BackColor = Color.Black;

btncontinue.ForeColor = Color.White;

}

private void btncontinue\_MouseLeave(object sender, EventArgs e)

{

btncontinue.BackColor = Color.FromArgb(128, 255, 128);

btncontinue.ForeColor = Color.Black;

}

private void btnUpdate\_MouseEnter(object sender, EventArgs e)

{

btnUpdate.BackColor = Color.Black;

btnUpdate.ForeColor = Color.White;

}

private void btnUpdate\_MouseLeave(object sender, EventArgs e)

{

btnUpdate.BackColor = Color.FromArgb(128, 255, 128);

btnUpdate.ForeColor = Color.Black;

}

}

**Pharmacy Forms code:** View Medicine.cs

public partial class ViewMedicine\_Form : Form

{

DataTable medicines = new DataTable();

public ViewMedicine\_Form()

{

InitializeComponent();

}

private void ViewMedicine\_Form\_Load(object sender, EventArgs e)

{

medicines.Columns.Add("Medicinename", Type.GetType("System.String"));

medicines.Columns.Add("Quantity", Type.GetType("System.String"));

medicines.Columns.Add("Expiry Date", Type.GetType("System.String"));

medicines.Columns.Add("Price", Type.GetType("System.String"));

for(int i = 0; i < Medicine\_CRUD.meds.Count; i++)

{

Medicine med = Medicine\_CRUD.meds[i];

medicines.Rows.Add(med.getMedicinename(), med.getQuantity(), med.getExpirydate(), med.getPrice());

}

medicinesGV.DataSource = medicines;

}

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

{

this.Close();

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

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

{

DataView dvPatientTable = medicines.DefaultView;

dvPatientTable.RowFilter = "Medicinename Like '%" + txtSearchBox.Text + "%'";

}

private void txtSearchBox\_Enter\_1(object sender, EventArgs e)

{

if (txtSearchBox.Text == "Medicine Name")

{

txtSearchBox.Text = "";

txtSearchBox.ForeColor = Color.Black;

}

}

private void txtSearchBox\_Leave\_1(object sender, EventArgs e)

{

if (txtSearchBox.Text == "")

{

txtSearchBox.Text = "Medicine Name";

txtSearchBox.ForeColor = Color.DimGray;

}

}

}

**Pharmacy Forms code:** Purchase Medicine.cs

public partial class PurchaseMedicine\_Form : Form

{

Customer customer;

public PurchaseMedicine\_Form()

{

InitializeComponent();

}

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

{

string textcustomer = txtCustomername.Text;

customer = new Customer();

customer.setcustomerName(textcustomer);

}

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

{

Medicine purchasemed = new Medicine();

string MedicineName = txtmedicinename.Text;

purchasemed.setMedicinename(MedicineName);

string quantityPurchaseInString = txtquantity.Text;

purchasemed.setQuantity(int.Parse(quantityPurchaseInString));

purchasemed = customer.purchasemedicine(Medicine\_CRUD.meds, purchasemed);

if (purchasemed != null)

{

customer.AddMedicineInList(purchasemed);

MessageBox.Show("Added");

}

else

{

MessageBox.Show("This medicine is out of stock");

}

txtmedicinename.Text = null;

txtquantity.Text = null;

}

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

{

Customer\_CRUD.AddCustomerinList(customer);

double totalMedicine\_Bill = customer.Calculate\_Bill();

lblmedicineExpense.Text = ""+totalMedicine\_Bill;

Customer\_CRUD.add\_purchaseMedicineinFile();

Medicine\_CRUD.SaveChange();

MessageBox.Show("Purchased");

empty();

}

private void empty()

{

txtCustomername.Text = null;

txtmedicinename.Text = null;

txtquantity.Text = null;

lblmedicineExpense.Text = null;

}

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

{

this.Close();

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnnext\_MouseEnter(object sender, EventArgs e)

{

btnnext.BackColor = Color.Black;

btnnext.ForeColor = Color.White;

}

private void btnnext\_MouseLeave(object sender, EventArgs e)

{

btnnext.BackColor = Color.FromArgb(128, 255, 128);

btnnext.ForeColor = Color.Black;

}

private void btnadd\_MouseEnter(object sender, EventArgs e)

{

btnadd.BackColor = Color.Black;

btnadd.ForeColor = Color.White;

}

private void btnadd\_MouseLeave(object sender, EventArgs e)

{

btnadd.BackColor = Color.FromArgb(128, 255, 128);

btnadd.ForeColor = Color.Black;

}

private void btnpurchase\_MouseEnter(object sender, EventArgs e)

{

btnpurchase.BackColor = Color.Black;

btnpurchase.ForeColor = Color.White;

}

private void btnpurchase\_MouseLeave(object sender, EventArgs e)

{

btnpurchase.BackColor = Color.FromArgb(128, 255, 128);

btnpurchase.ForeColor = Color.Black;

}

}

**Pharmacy Forms code:** Delete Medicine.cs

public partial class DeleteMedicine\_Form : Form

{

public DeleteMedicine\_Form()

{

InitializeComponent();

}

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

{

string deleteMedicine = txtdeleteMedicine.Text;

bool delmedicineCheck = Medicine\_CRUD.deletemedicine(deleteMedicine);

if (delmedicineCheck == true)

{

MessageBox.Show("Requested medicine is deleted");

Medicine\_CRUD.DataAFTERDeleteMedicine();

}

else

{

MessageBox.Show("The medicine does not exist in our database");

}

txtdeleteMedicine.Text = null;

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btndelete\_MouseEnter(object sender, EventArgs e)

{

btndelete.BackColor = Color.Black;

btndelete.ForeColor = Color.White;

}

private void btndelete\_MouseLeave(object sender, EventArgs e)

{

btndelete.BackColor = Color.FromArgb(128, 255, 128);

btndelete.ForeColor = Color.Black;

}

}

**Pharmacy Forms code:** View Demand Medicine.cs

public partial class ViewDemandMedicine\_Form : Form

{

public ViewDemandMedicine\_Form()

{

InitializeComponent();

}

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

{

this.Close();

}

private void ViewDemandMedicine\_Form\_Load(object sender, EventArgs e)

{

int check;

string demandmed = "";

foreach (Medicine j in Medicine\_CRUD.meds)

{

check = j.getQuantity();

if (check <= j.getDemandcheck() / 2)

{

demandmed = demandmed + j.getMedicinename()+",";

}

}

for(int j = 0; j < demandmed.Length - 1; j++)

{

lblDemandmed.Text = lblDemandmed.Text + demandmed[j];

}

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

}

**Pharmacy Forms code:** View Expired Medicine.cs

public partial class ViewExpiredMedicine\_Form : Form

{

public ViewExpiredMedicine\_Form()

{

InitializeComponent();

}

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

{

string currentdate;

int check = 0;

bool expiredmedicine;

string expiredMed="";

DateTime date = dtDateExpired.Value;

currentdate = Convert.ToDateTime(date).Date.ToString("d");

foreach (Medicine i in Medicine\_CRUD.meds)

{

expiredmedicine = ExpiredMedicine.isExpired(currentdate, i.getExpirydate());

if (expiredmedicine == true)

{

check = 1;

string expiredmedicines1 = i.getMedicinename();

ExpiredMedicine expiredmedicine1 = new ExpiredMedicine(expiredmedicines1);

ExpiredMedicine\_CRUD.expiredmed.Add(expiredmedicine1);

}

}

ExpiredMedicine\_CRUD.addExpiredmedicine();

if (check == 1)

{

foreach (ExpiredMedicine i in ExpiredMedicine\_CRUD.expiredmed)

{

expiredMed = expiredMed + i.getExpiredmedicine() + ",";

}

for (int j = 0; j < expiredMed.Length - 1; j++)

{

lblexpiredmed.Text = lblexpiredmed.Text + expiredMed[j];

}

}

else

{

MessageBox.Show("No medicine is expired");

}

}

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

{

this.Close();

}

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

{

Form newform = new Pharmacy\_Menu\_Form();

newform.Show();

this.Hide();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Environment.Exit(1);

}

private void btnCheck\_MouseEnter(object sender, EventArgs e)

{

btnCheck.BackColor = Color.Black;

btnCheck.ForeColor = Color.White;

}

private void btnCheck\_MouseLeave(object sender, EventArgs e)

{

btnCheck.BackColor = Color.FromArgb(128, 255, 128);

btnCheck.ForeColor = Color.Black;

}

}

**Pharmacy Main Menu Form:** Main.cs

public partial class Pharmacy\_Menu\_Form : Form

{

public Pharmacy\_Menu\_Form()

{

InitializeComponent();

}

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

{

Form newform = new AddMedicine\_Form();

newform.Show();

this.Hide();

}

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

{

Form newform = new ViewMedicine\_Form();

newform.Show();

this.Hide();

}

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

{

Form newform = new UpdateMedicine\_Form();

newform.Show();

this.Hide();

}

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

{

Form newform = new PurchaseMedicine\_Form();

newform.Show();

this.Hide();

}

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

{

Form newform = new ViewExpiredMedicine\_Form();

newform.Show();

this.Hide();

}

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

{

Form newform = new ViewDemandMedicine\_Form();

newform.Show();

this.Hide();

}

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

{

Form newform = new DeleteMedicine\_Form();

newform.Show();

this.Hide();

}

private void Pharmacy\_Menu\_Form\_Load(object sender, EventArgs e)

{

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

this.Hide();

Form newform = new HMSApplication();

newform.Show();

}

private void menuStrip2\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

}

**Admin Main Menu Form:** Main.cs

public partial class Admin\_menu\_Form : Form

{

public Admin\_menu\_Form()

{

InitializeComponent();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

this.Hide();

Form newform = new HMSApplication();

newform.Show();

}

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

{

Form newForm = new Patient\_Form.Add\_Patient\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.View\_Patient\_form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Update\_Patient\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Issue\_Bed\_form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Suggest\_Medicine\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Manage\_DoctorSchedule\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.ManageAppointments\_form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Generate\_medical\_Report\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Check\_Doctor\_Appointment\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Billing\_Form();

newForm.Show();

this.Hide();

}

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

{

Form newForm = new Patient\_Form.Discharge\_Patient\_Form();

newForm.Show();

this.Hide();

}

private void Admin\_menu\_Form\_Load(object sender, EventArgs e)

{

}

}

**Patient Main Menu Form:** Main.cs

public partial class Patient\_menu\_form : Form

{

public Patient\_menu\_form()

{

InitializeComponent();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form newform = new HMSApplication();

newform.Show();

this.Hide();

}

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

{

Form newform = new DetailForm();

newform.Show();

this.Hide();

}

private void Patient\_menu\_form\_Load(object sender, EventArgs e)

{

}

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

{

Form newform = new ViewPrescriptionForm();

newform.Show();

}

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

{

Form newform = new ViewCommonDiseasesForm();

newform.Show();

}

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

{

Form newform = new ViewMedicalReportForm();

newform.Show();

}

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

{

Form newform = new CheckBill\_Form();

newform.Show();

}

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

{

Form newform = new CheckBedAvailabilityForm();

newform.Show();

}

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

{

Form newform = new CheckDoctorSchedule\_Form();

newform.Show();

}

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

{

Form newform = new BookAppointmentForm();

newform.Show();

}

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

{

Form newform = new PatientSatisfactionSurveyForm();

newform.Show();

}

private void menuStrip2\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

}

# Class Details:

**Muser:** Inside this class I have added attributes i.e. username, password and role variable and I have added the getter setter functions to access and set the user attributes.

**Patient:** This class has attributes name,cnic, contact no, address, medical history, and getter setter function are also included. Other functions related to the patient are also included.

**Doctor:** This class has attributes of Doctor name, Profession, Schedule date, ending, and starting time. It includes my appointment list.

**Customer who buys medicine:** This class has the attribute Customer name and purchase medicine list.

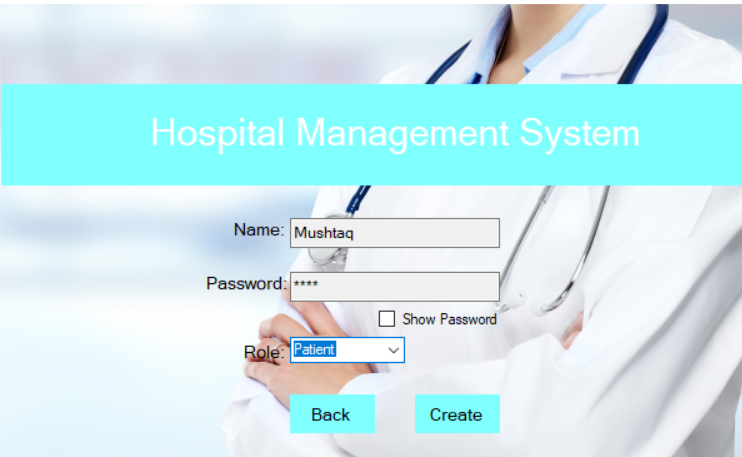
**Medicine:** This class has attributes of medicine name, quantity, expiry date, and price. It also includes the getter and setter functions.

**DL classes:** Inside DL classes I have applied all the CRUD functions in the classes to the respective list they have some kind of association and composition with the BL classes.

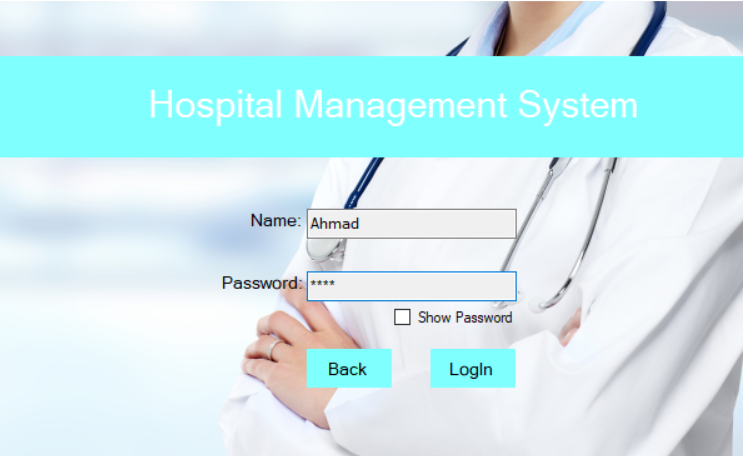
**UI classes:** In User UI I have all print and take input functions. It prints all the functions which may use with BL and DL classes.

# WireFrames:

**SignIn&SignUp Form:**

**SignUp Form:**

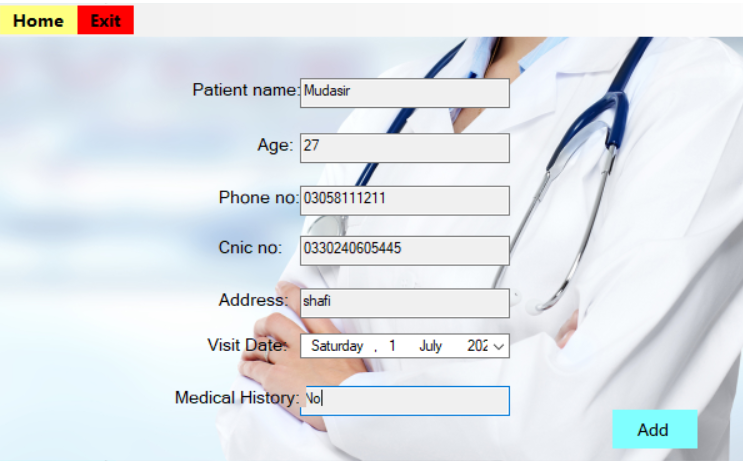
**SignIn Form:**

****

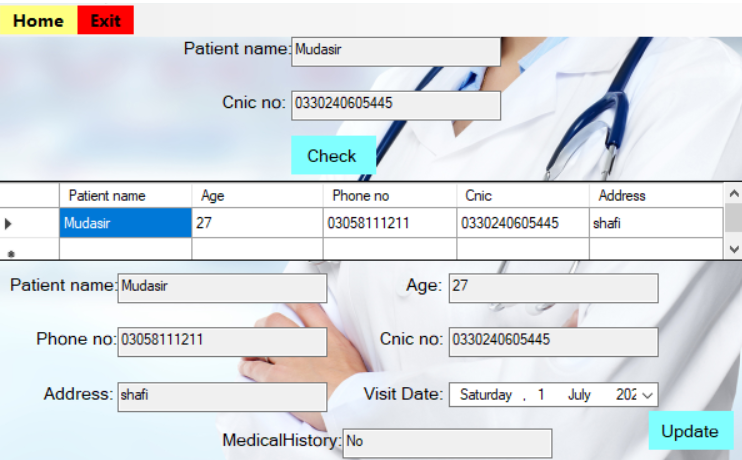
**AdminMenu MainForm:**

****

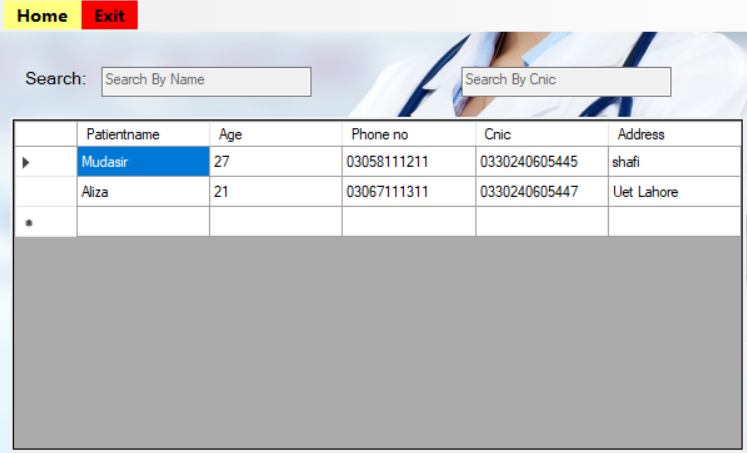
**Add Patient Form:**

****

**UpdatePatient Form:**

****

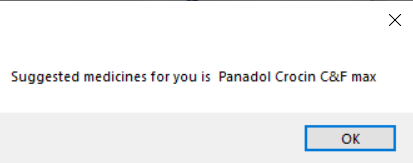
**View Patient Form:**

****

**Symptoms Form:**

****

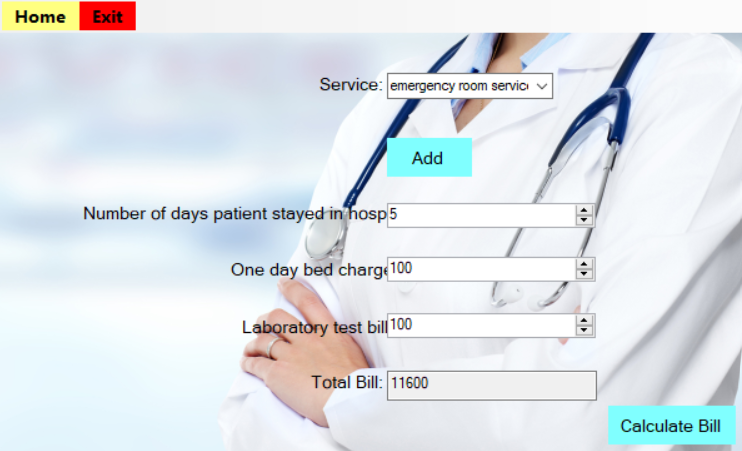
**Suggested Medicine According to the Symptoms:**

****

**Generate Report Form:**

****

**Billing/Invoicing Form:**

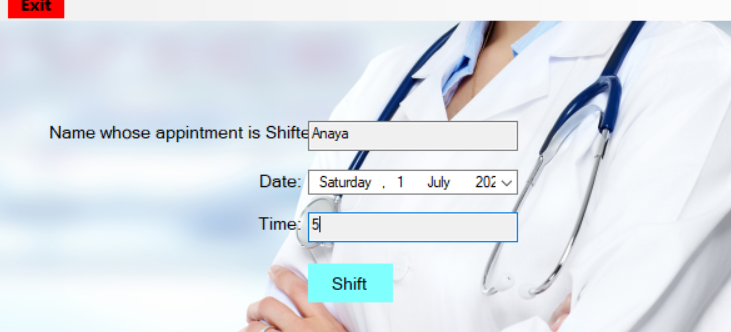
****

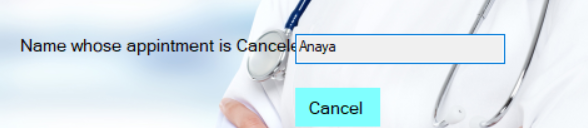
**Set Doctor Schedule Form:**

****

**Manage Appointment Form:**

****

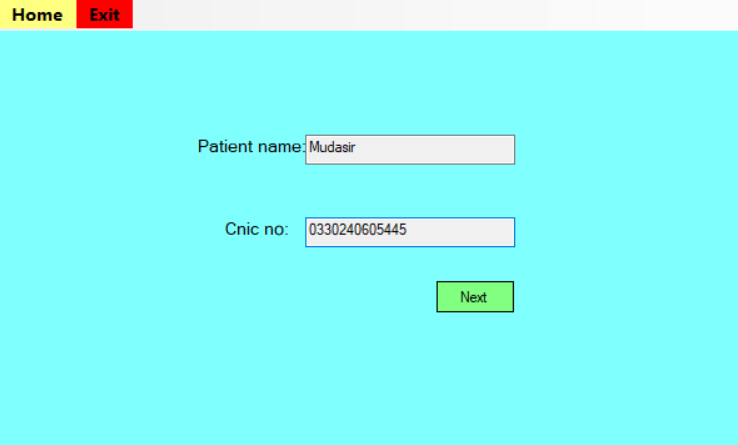
****

****

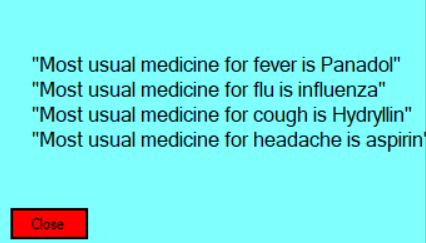
**PatientMenu MainForm:**

****

**Detail Form:**

****

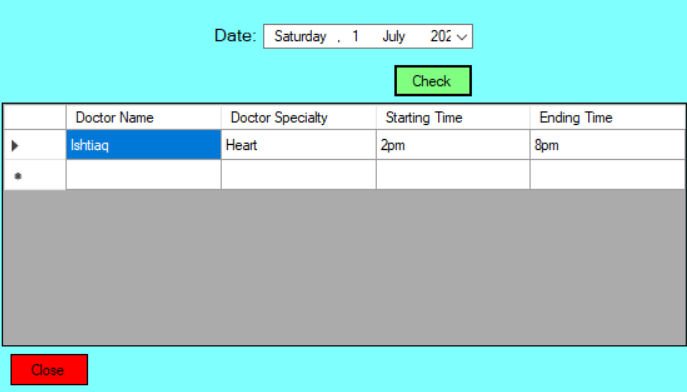
**View Common Diseases:**

****

**View Medical Report Form:**

****

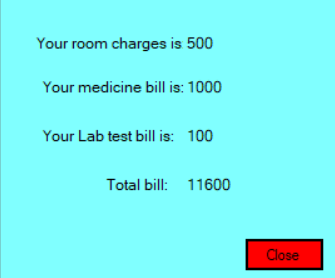
**Check Doctor Schedule Form:**

****

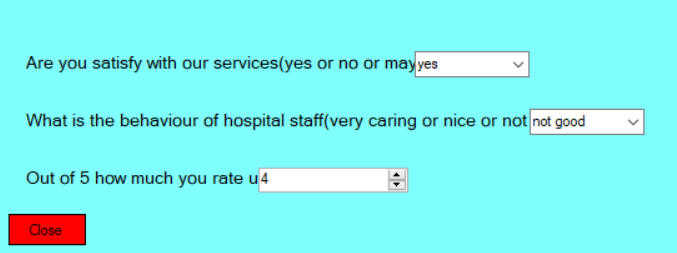
**Book Appointment Form:**

****

**Check Bill:**

****

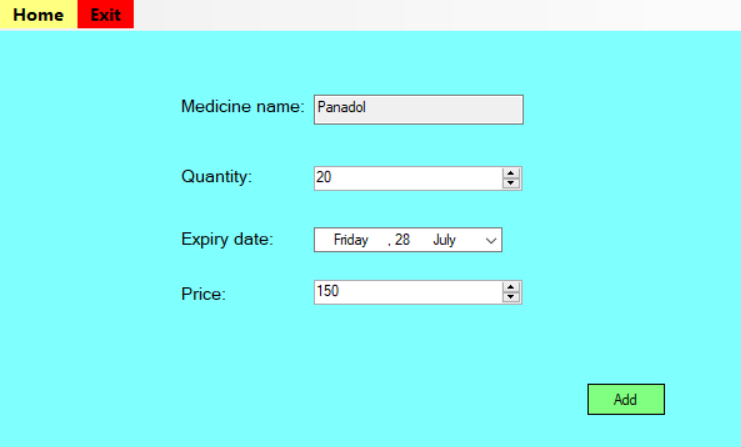
**Patient Satisfaction Survey:**

****

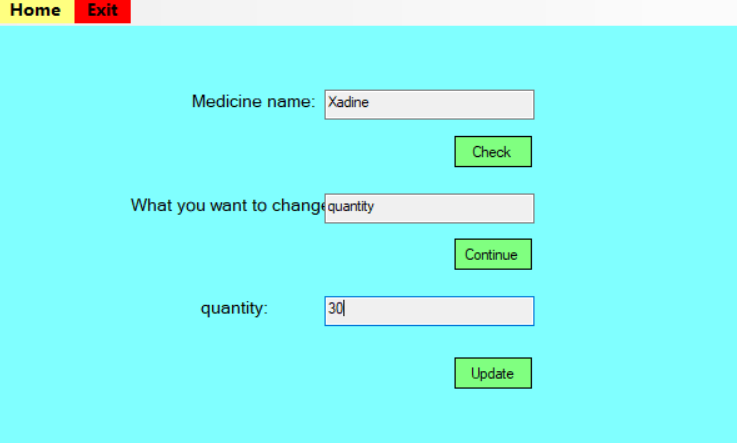
**PharmacyMenu MainForm:**

****

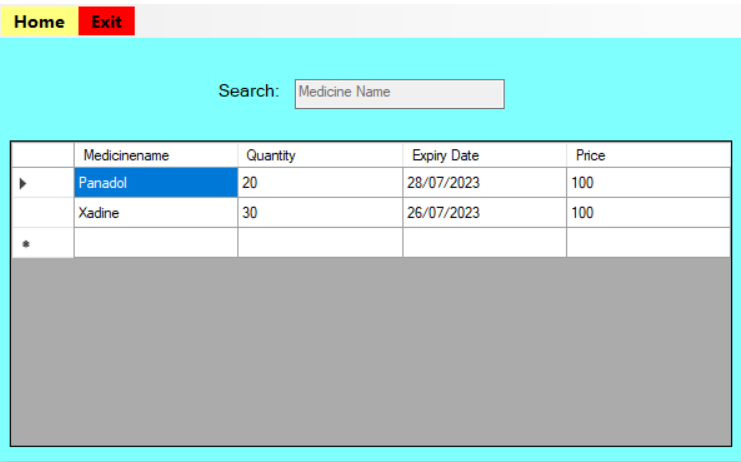
**Add Medicine Form:**

****

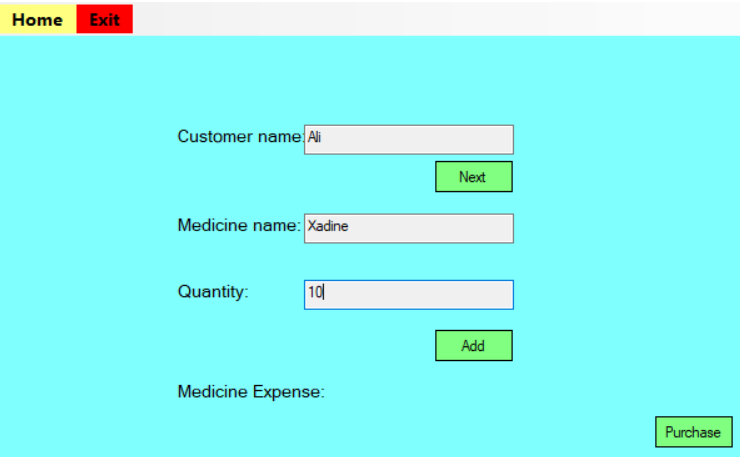
**Update Medicine Form:**

****

**View Medicine Form:**

****

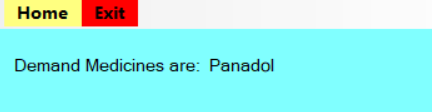
**Purchase Medicine Form:**

****

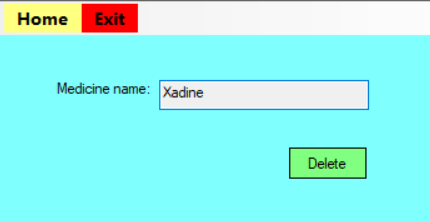
**View Expired Medicine Form:**

****

**View Demand Medicine Form:**

****

**Delete Medicine Form:**

****

# Conclusion:

The Hospital Management System project in C# using OOP aimed to automate various hospital operations. I faced challenges in understanding the complex domain, designing the data model, creating a user-friendly interface, implementing business logic, and ensuring system scalability. I tried to fulfill all the requirements. However, through research, learning, and problem-solving, I successfully completed the project.