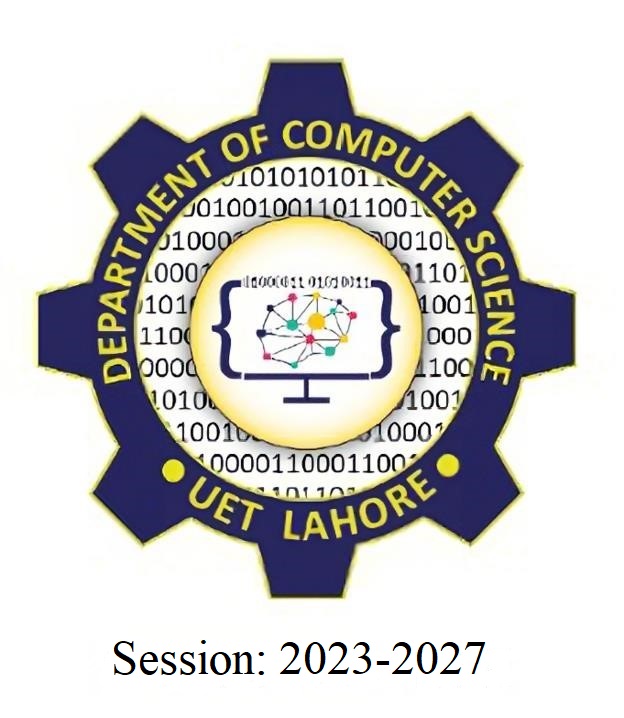
**Pysantas Medi Harmony**



**Submitted by:**

Muhammad Owais 2023-CS-131

**Supervised by:**

Ma’am Maida Shahid

Department of Computer Science

**University of Engineering and Technology Lahore Pakistan**

Table of Contents

[ Contribution toward CS: 4](#_Toc153630928)

[ Why Medi Harmony: 4](#_Toc153630929)

[ What to Expect from Medi Harmony: 5](#_Toc153630930)

[❖ User Types on Medi Harmony: 5](#_Toc153630931)

[I. Admin: 5](#_Toc153630932)

[II. Pharmacist: 5](#_Toc153630933)

[III. Patient: 5](#_Toc153630934)

[IV. Doctor: 5](#_Toc153630935)

[V. Equipment manager: 5](#_Toc153630936)

[❖ Functional Requirements of Medi Harmony: 5](#_Toc153630937)

[❖ Wireframe of Medi Harmony: 8](#_Toc153630938)

[ Basic (For All types of Users): 8](#_Toc153630939)

[ Startup Interface: 8](#_Toc153630940)

[ Branch Menu: 8](#_Toc153630941)

[ Islamabad Branch Interface: 10](#_Toc153630942)

[ Lahore Branch Interface 10](#_Toc153630943)

[ Multan Branch Interface: 11](#_Toc153630944)

[ Login Menu: 11](#_Toc153630945)

[ Signup Menu 12](#_Toc153630946)

[ Sign in menu: 12](#_Toc153630947)

[ Admin Menu: 13](#_Toc153630948)

[ Add Patient: 13](#_Toc153630949)

[ View Patient: 14](#_Toc153630950)

[ Discharge Patient: 14](#_Toc153630951)

[ Profit Loss Calculation 15](#_Toc153630952)

[ Changing Salary: 15](#_Toc153630953)

[ Add Bed: 16](#_Toc153630954)

[ Add Doctor: 16](#_Toc153630955)

[ View Doctor: 17](#_Toc153630956)

[ Add Equipment-Manager: 17](#_Toc153630957)

[ Change Equipment-Manager: 18](#_Toc153630958)

[ Update Patient: 18](#_Toc153630959)

[ Doctor Menu: 19](#_Toc153630960)

[ View Admitted Patient: 19](#_Toc153630961)

[ View Available Beds: 20](#_Toc153630962)

[ View Doctor’s List: 20](#_Toc153630963)

[ Patient Menu: 21](#_Toc153630964)

[ Select Doctor: 21](#_Toc153630965)

[ Pay Bills: 22](#_Toc153630966)

[ View Prescription: 22](#_Toc153630967)

[ Patient Review: 23](#_Toc153630968)

[ Buy Medicines: 23](#_Toc153630969)

[ Pharmacist Menu: 24](#_Toc153630970)

[ Add Medicine: 24](#_Toc153630971)

[ View Medicines: 25](#_Toc153630972)

[ Update Medicine: 25](#_Toc153630973)

[ Expired medicine: 26](#_Toc153630974)

[ Mostly Used Medicines: 26](#_Toc153630975)

[ Remove Medicines: 27](#_Toc153630976)

[ Add Employ: 27](#_Toc153630977)

[ Remove Employ: 28](#_Toc153630978)

[ View Employ: 28](#_Toc153630979)

[ Equipment Manager: 29](#_Toc153630980)

[ Add Equipment: 29](#_Toc153630981)

[ View Equipment: 30](#_Toc153630982)

[ Remove Equipment: 30](#_Toc153630983)

[❖ Data Structure: 31](#_Toc153630984)

[❖ Function Prototypes: 38](#_Toc153630985)

[❖ Code: 40](#_Toc153630986)

[❖ Weakness: 127](#_Toc153630987)

[❖ Future Directions: 127](#_Toc153630988)

Table of Figures

[Figure 1. Startup Interface 10](#_Toc153863875)

[Figure 2. Islamabad Branch Interface 10](#_Toc153863876)

[Figure 3-Lahore Branch Interface 11](#_Toc153863877)

[Figure 4-Multan Branch Interface 11](#_Toc153863878)

[Figure 5- Login Menu- 12](#_Toc153863879)

[Figure 6-Sign up menu 13](#_Toc153863880)

[Figure 7-Signin Menu 13](#_Toc153863881)

[Figure 8-Admin Menu 14](#_Toc153863882)

[Figure 9-Add Patient Menu 14](#_Toc153863883)

[Figure 10-View Patient Menu 15](#_Toc153863884)

[Figure 11-Discharge Menu 15](#_Toc153863885)

[Figure 12-Profit Loss Menu 16](#_Toc153863886)

[Figure 13-Changing Salary Menu 16](#_Toc153863887)

[Figure 14-Add Bed Menu 17](#_Toc153863888)

[Figure 15-Add Doctor Menu 17](#_Toc153863889)

[Figure 16-View Doctor Menu 18](#_Toc153863890)

[Figure 17-Add Manager Menu 18](#_Toc153863891)

[‘ Figure 18-Change Manger Menu 19](#_Toc153863892)

[Figure 19-Update Patient Menu 19](#_Toc153863893)

[Figure 20-Doctor Menu 20](#_Toc153863894)

[Figure 21-Add Doctor Menu 20](#_Toc153863895)

[Figure 22-View Available Beds Menu 21](#_Toc153863896)

[Figure 23-View Available Doctor Menu 21](#_Toc153863897)

[Figure 24-Patient Menu 22](#_Toc153863898)

[Figure 25-Select Doctor Menu 22](#_Toc153863899)

[Figure 26-Bill Menu 23](#_Toc153863900)

[Figure 27-View Prescription Menu 23](#_Toc153863901)

[Figure 28-Patient Review Menu 24](#_Toc153863902)

[Figure 29-Buy Medicines Menu 24](#_Toc153863903)

[Figure 30-Pharmacist Menu 25](#_Toc153863904)

[Figure 31-Add Medicine Menu 25](#_Toc153863905)

[Figure 32-View Medicine Menu 26](#_Toc153863906)

[Figure 33-Update Medicine Menu 26](#_Toc153863907)

[Figure 34-Expired Medicine Menu 27](#_Toc153863908)

[Figure 35-Medicine in used Menu 27](#_Toc153863909)

[Figure 36-Remove Medicine Menu 28](#_Toc153863910)

[Figure 37-Add Employ Menu 28](#_Toc153863911)

[Figure 38-Remove Employ Menu 29](#_Toc153863912)

[Figure 39-View Employ Menu 29](#_Toc153863913)

[Figure 40-Equipment Manager Menu 30](#_Toc153863914)

[Figure 41-Add Equipment Menu 30](#_Toc153863915)

[Figure 42-View Equipment Menu 31](#_Toc153863916)

[Figure 43-Remove Equipment Menu 31](#_Toc153863917)

❖ **About Medi Harmony:**

Medi Harmony is a user-friendly application, built to help the patient, doctors, equipment manager, admin and pharmacist to approach different contents in the hospital. As the users can see the progress of the all the things like stock etc. They can also view the details of the equipment’s. The patient can take appointment, view medicines and all the other things. The patient sees the available beds in the hospital, see the bills online and pays it. Give the satisfactory report, buy medicines. The pharmacist adds, delete and view medicines stock. Also, he sees the expired medicines details from his stock.

##  Contribution toward CS:

Medi Harmony is a business application that can helps the patient to take appointment and to save his/her own precocious time. This application run the whole application in C++ and it will run accurately. This is the main thing that I can run all the things in this as a computer scientist. This application is capable of running a whole hospital management system easily.

##  Why Medi Harmony:

Medi Harmony a one of the credible applications all over country people like to work and share things with it. As the application is running very smoothly and there is no chance of fraud and any other misunderstanding as the is a big privacy before all the work. There is a copy right mark in this Medi Harmony.

##  What to Expect from Medi Harmony:

The Medi Harmony is a big application and trustable so feel free to contact us. Also, you can register his/her own bed in this application. So, the chance of misunderstanding will be ended. This is why, the major population of our country using this application.

# ❖ User Types on Medi Harmony:

Medi Harmony provides different access for different type of users. The users are divided into three categories, patient, pharmacist and Owner (Admin). Each user type has access to different types of command related to their need and requirements. User can login and verify their type by inputting the username and password for authentication.

The hierarchy and functionality of user types is as under: -

## I. Admin:

Owner or admin has access to add patients, to view the patients and add the patient, discharging the patient, add beds, add doctors, viewing the doctors list, add and remove the manager, change the salary and view the profit and loss. Also, he has to access to all the internal functions of the hospital.

## II. Pharmacist:

Pharmacist has access to add medicines, remove medicines, view medicines list, updating the medicines list, view the expired medicines, add the employ to the shop and remove the employ. He also gave prescription to some common diseases. He also views the medicines which are in demand all over the country.

## III. Patient:

Patient is the main user of the application. The main purpose of this application is to serve the patient as much as a possible. Patient have ability to view doctors schedule, available beds, selecting doctors, bill check, buying medicines. Also, patient can view the doctor schedule. Patient also gives the feedback to the staff if he likes the medication.

## IV. Doctor:

Doctor can view the patients added in the hospital and he also view that he hired by someone or not. To suggest the patient medication, he can also view the number of beds available in the hospital.

## V. Equipment manager:

There’s a manager in the hospital that can has the details of the equipment. He adds the equipment in the stock, view the admitted equipment, remove the broken equipment from the system. He also updated the added equipment.

# ❖ Functional Requirements of Medi Harmony:

Some of the functional requirements expected from Medi Harmony are as under:

|  |  |  |
| --- | --- | --- |
| ***User Type*** | ***Required Function to be Performed*** | ***Result of Action Performed*** |
| ***ADMIN*** | Add Patient | Admin can add the patient which come for treatment. |
| View Patient | Admin can also view the patients list which were added. |
| Discharge Patient | Admin has ability to discharge the patient after treatment. |
| Profit | Admin can view the profit by simply adding the patient. |
| Changing salary | Admin can change the salary of the doctors as well as pharmacist and manager. |
| Add Bed | Admin can add new beds if needed. For the suffering patients. |
| Add Doctors | Admin can add the doctors for the patients. |
| View Doctors List | Admin has ability to view the doctor’s list. |
| Add Equipment’s Manager | Admin can add manager. |
| Update patient | Admin have ability to update patient if any wrong information is entered. |
| ***Pharmacist*** | Add Medicine Form | Pharmacist can add the medicine. |
| View Medicines | Pharmacist can view the medicines list. |
| Update Medicines Form | Pharmacist can also view the medicines list. |
| View Employ | Pharmacist has ability to view the employ in the pharmacy department. |

|  |  |  |
| --- | --- | --- |
|  | Medicines in Demand | Pharmacist can view the medicines mostly used for diseases. |
|  | Expired Medicines | Pharmacist can also view the expired medicines. |
| Remove Medicines | Pharmacist can delete and exchange the unused medicines. |
| Add Employ | Pharmacist can also add the employ. |
| Remove Employ | Pharmacist can also remove employ. |
| ***Patient*** | View Doctors Schedule | Patient can view the doctors of the hospital. |
| View Prescription | Patient can also view the prescription to some diseases. |
| Billing and invoice | Patient can pay bills for the treatment. |
| View doctors list | Patient can view the doctors list. |
| Available Beds | Patient can see the beds. |
| Buy Medicines | Patient can buy the medicines. |
| ***Doctor*** | View Patient | Doctor can view patient which come for cure |
| View Available beds | Doctor can view the available beds to suggest patient. |
| View Doctor’s List | Doctor can view record list to know whether he is hired by someone or not. |
| ***Equipment Manager*** | Add Equipment | Manager can add the equipment for the surgery of the patients. |
| View Equipment | Manager can also access to the equipment record. |
| Remove Equipment | Manager can remove the equipment from the stock. |
| Update Medicines | Manager can also exchange the equipment which he wants. |

# ❖ Wireframe of Medi Harmony:

The following is the wireframe of Medi Harmony.

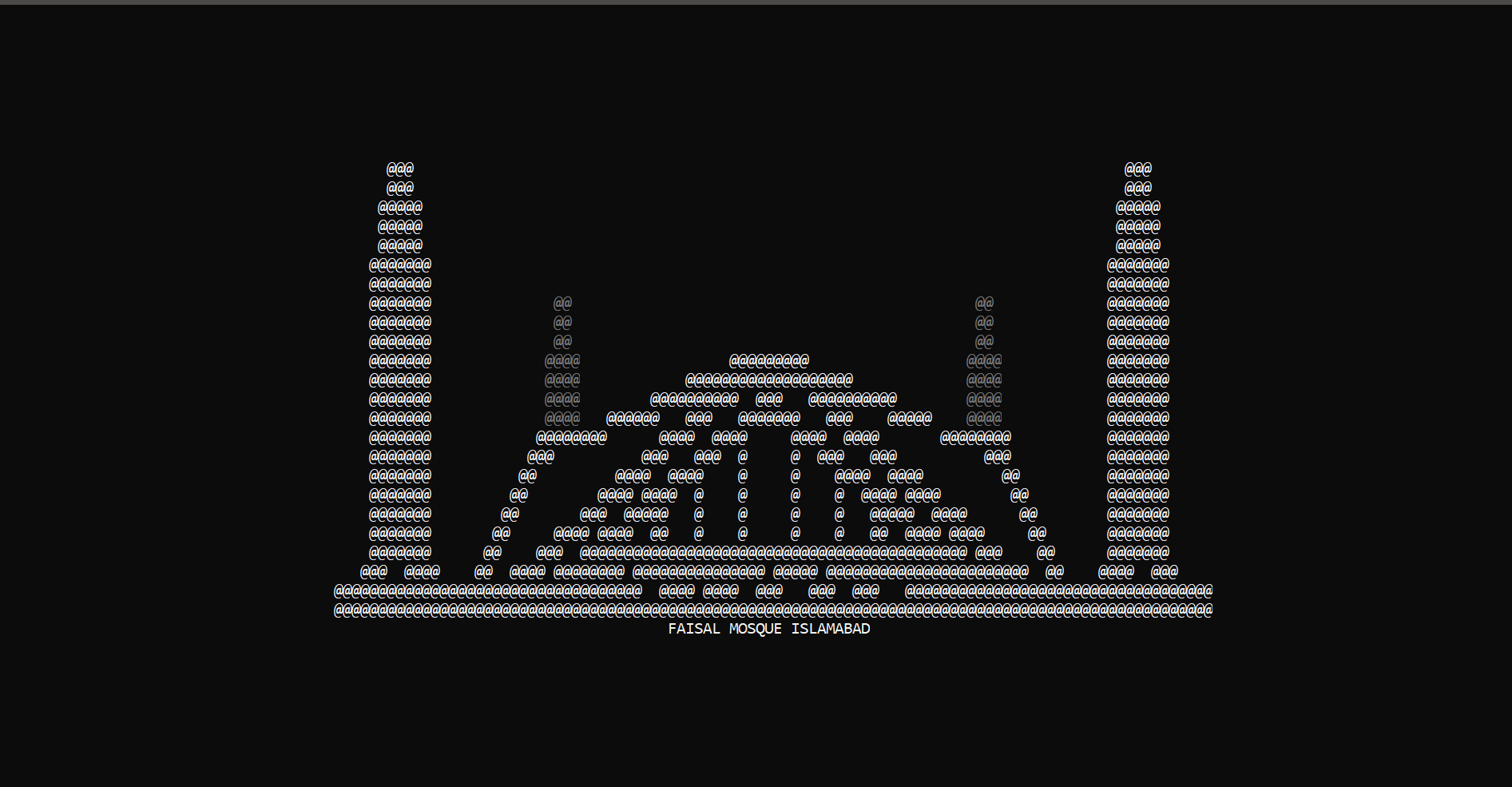
##  Basic (For All types of Users):

### Startup Interface:



Figure . Startup Interface

### Islamabad Branch Interface:

 Figure . Islamabad Branch Interface

* Lahore Branch Interface**:**



Figure -Lahore Branch Interface

### Multan Branch Interface:

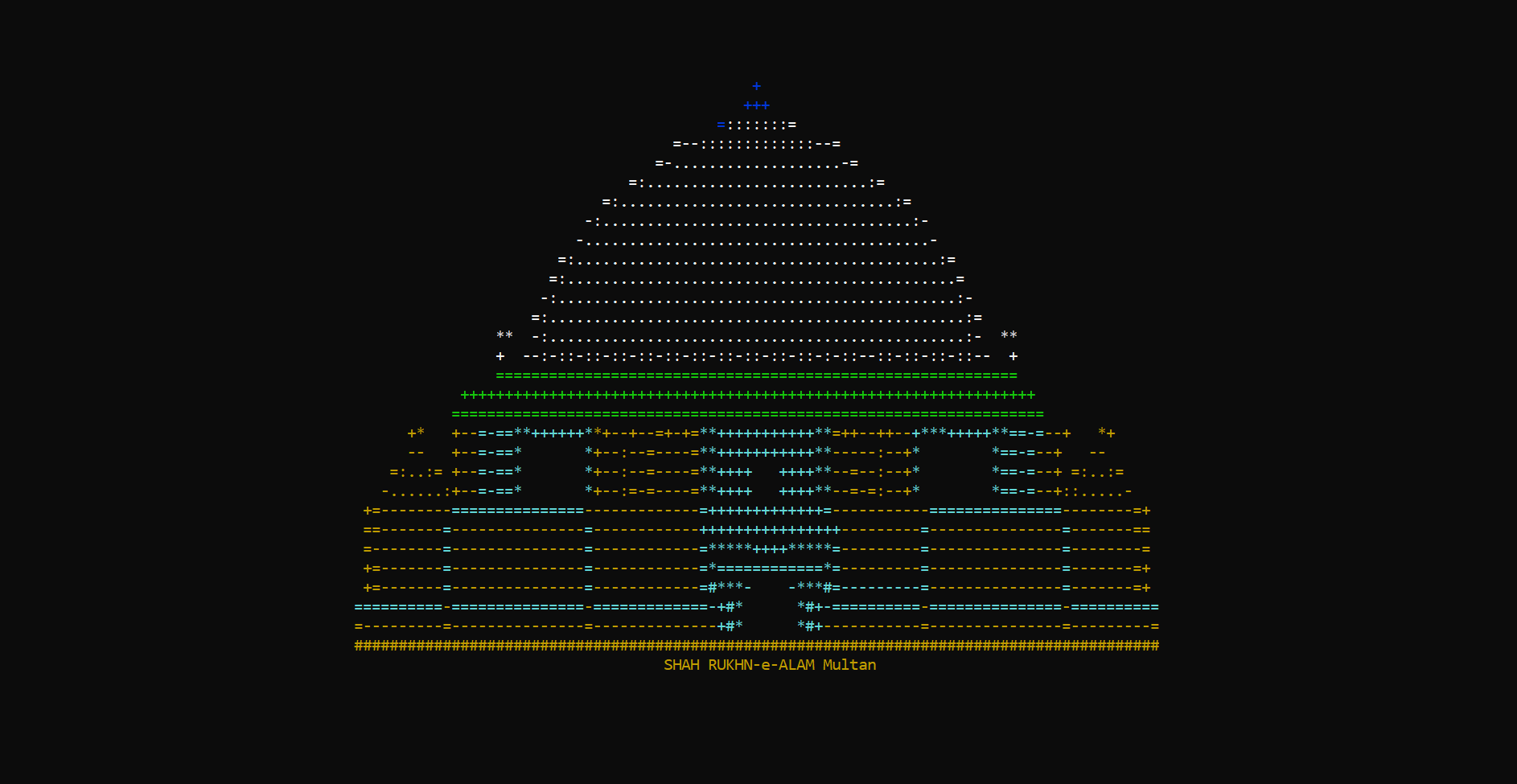


Figure -Multan Branch Interface

Login Menu:

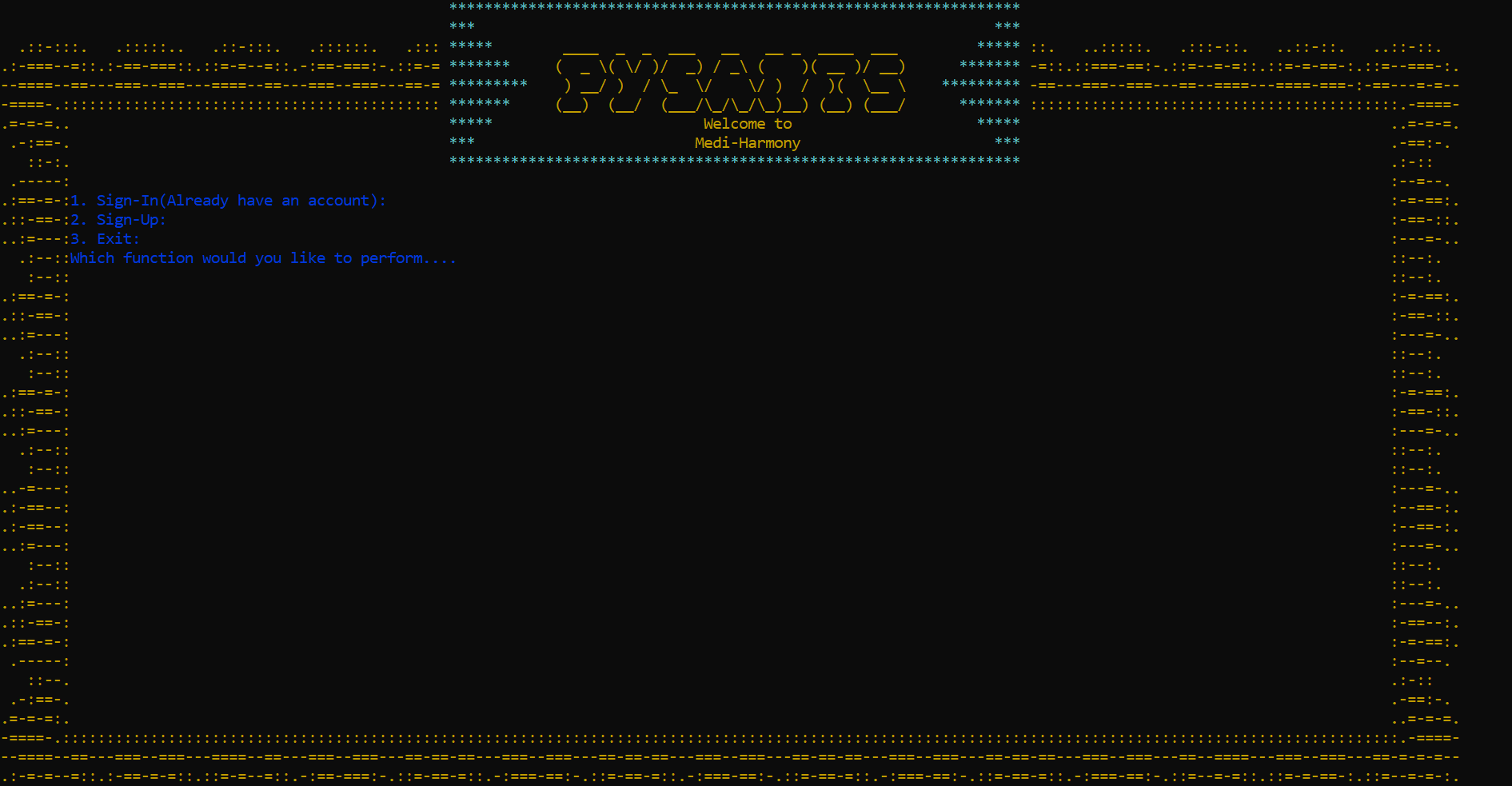


Figure - Login Menu-

* Signup Menu**:**

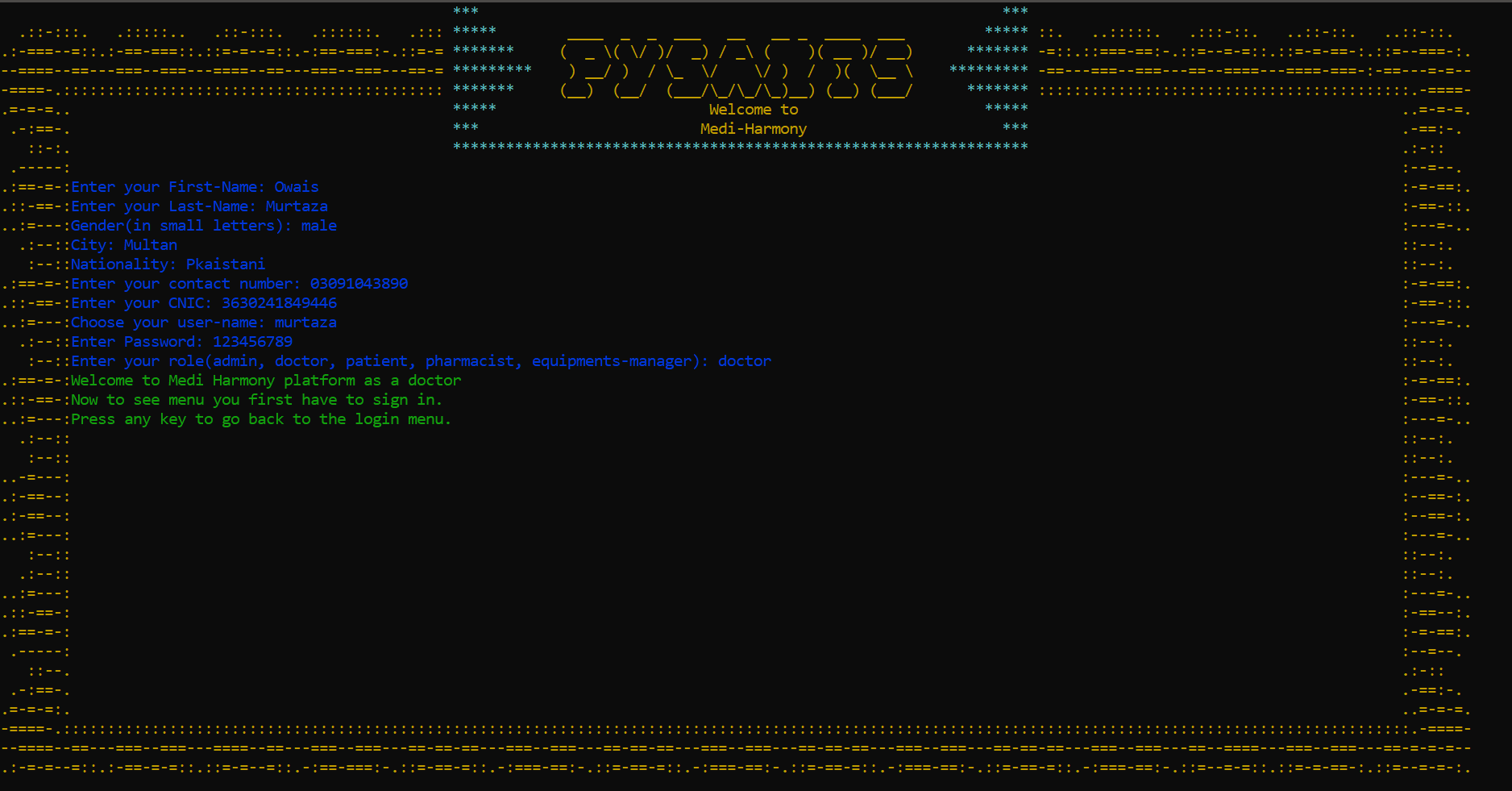


Figure -Sign up menu

### Sign in menu:



Figure -Signin Menu

### Admin Menu:

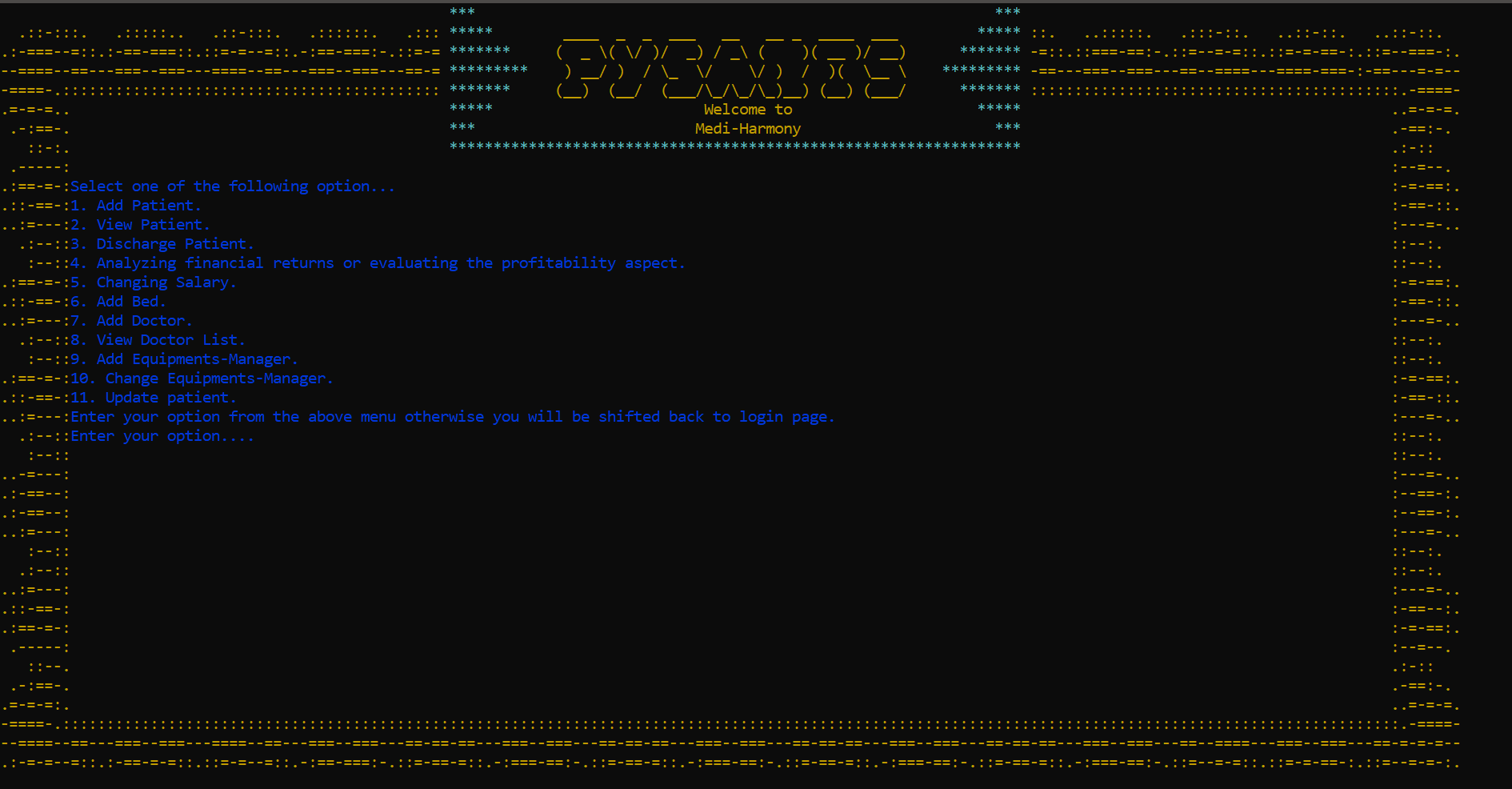


Figure -Admin Menu

### Add Patient:

****

Figure -Add Patient Menu

### View Patient:

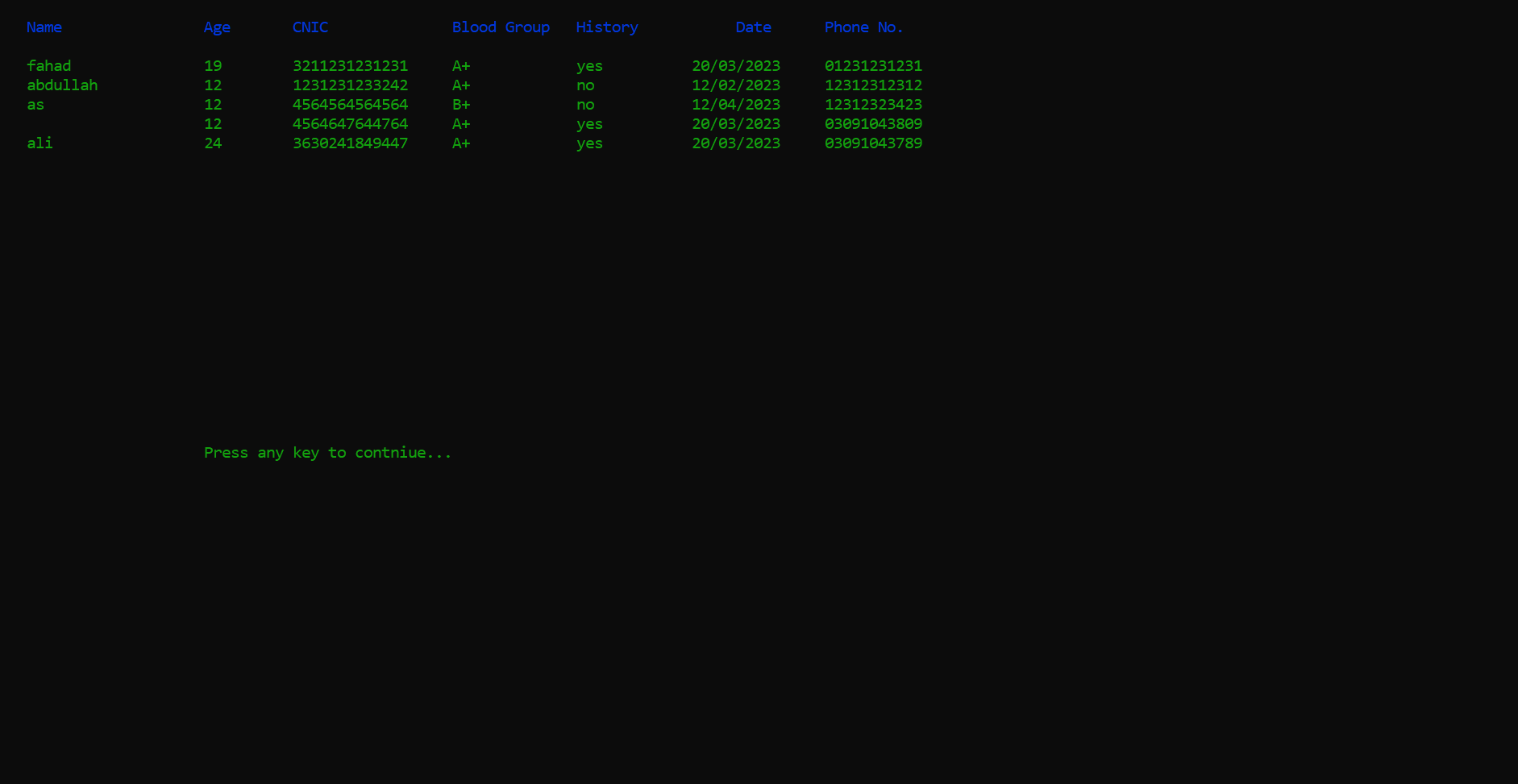


Figure -View Patient Menu

### Discharge Patient:

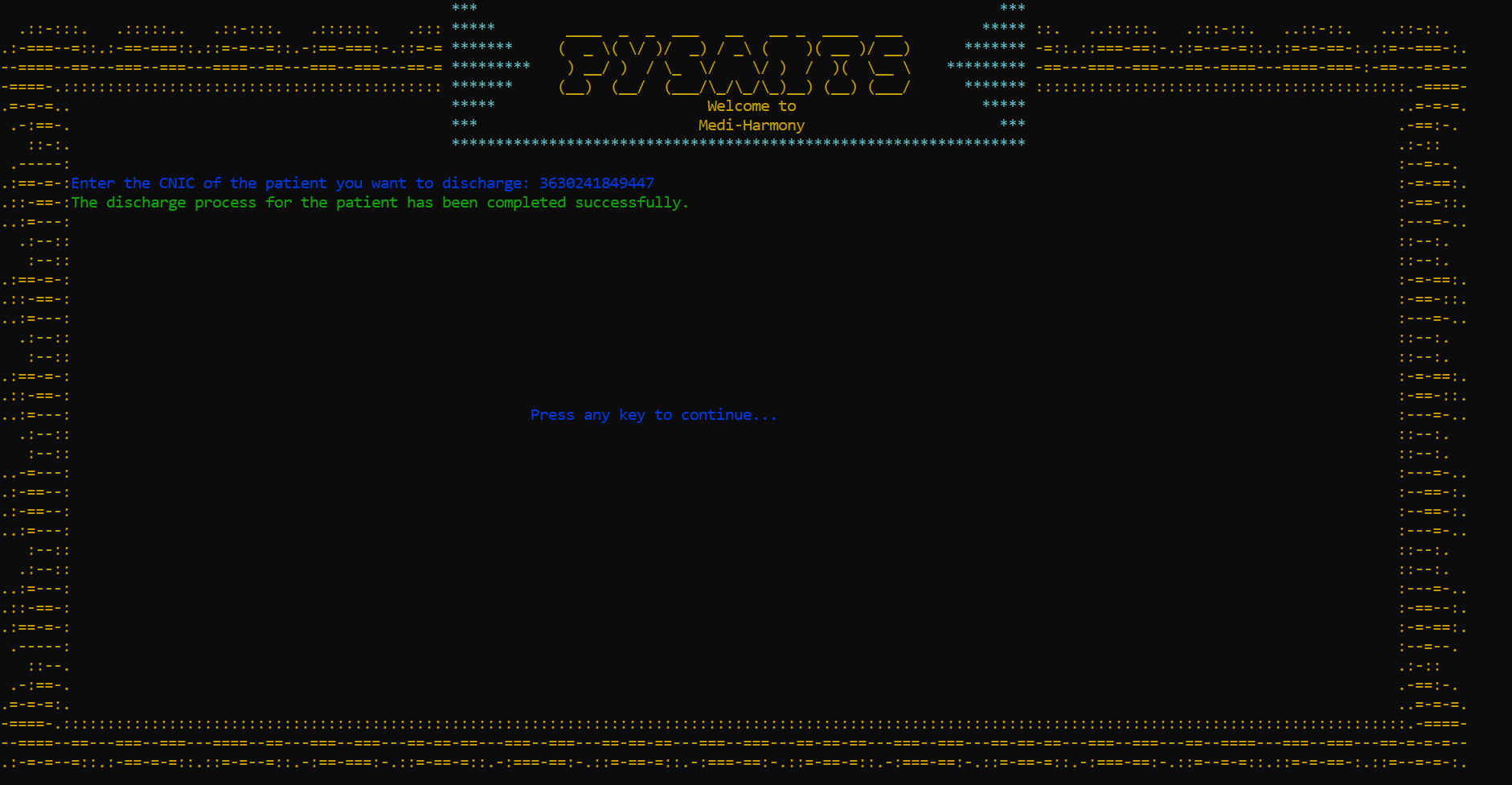


Figure -Discharge Menu

* Profit Loss Calculation**:**



Figure -Profit Loss Menu

### Changing Salary:



Figure -Changing Salary Menu

**Add Bed:**

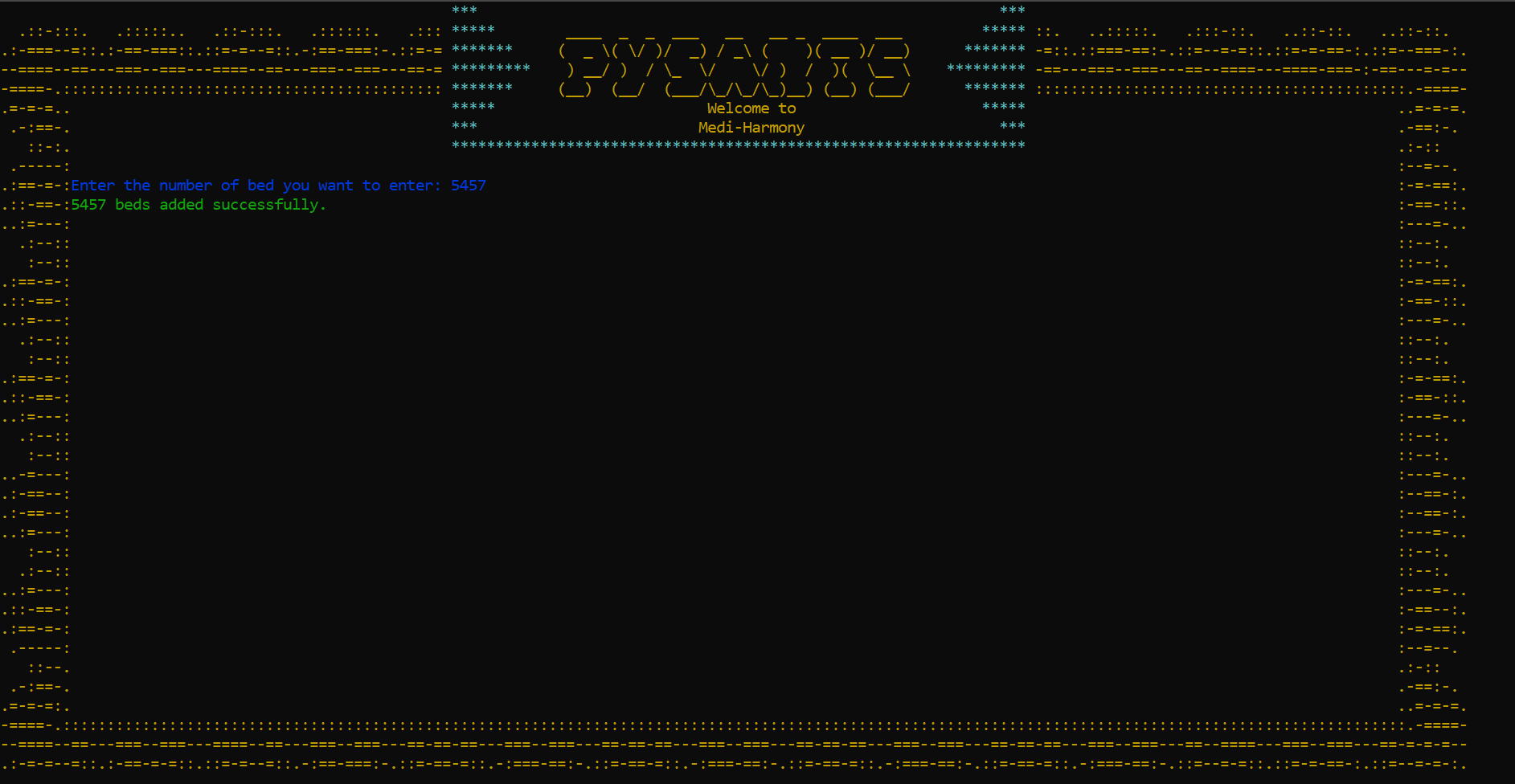


Figure -Add Bed Menu

### Add Doctor:



Figure -Add Doctor Menu

### View Doctor:

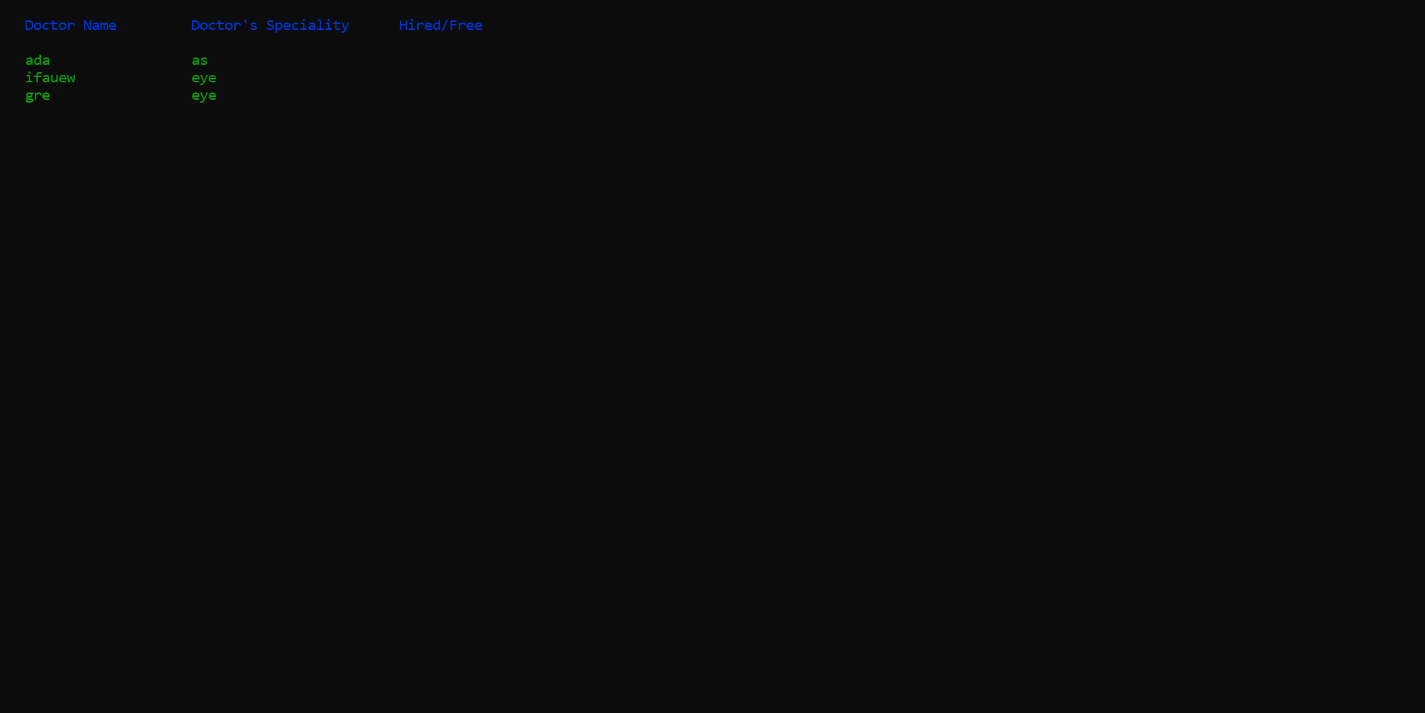


Figure -View Doctor Menu

### Add Equipment-Manager:

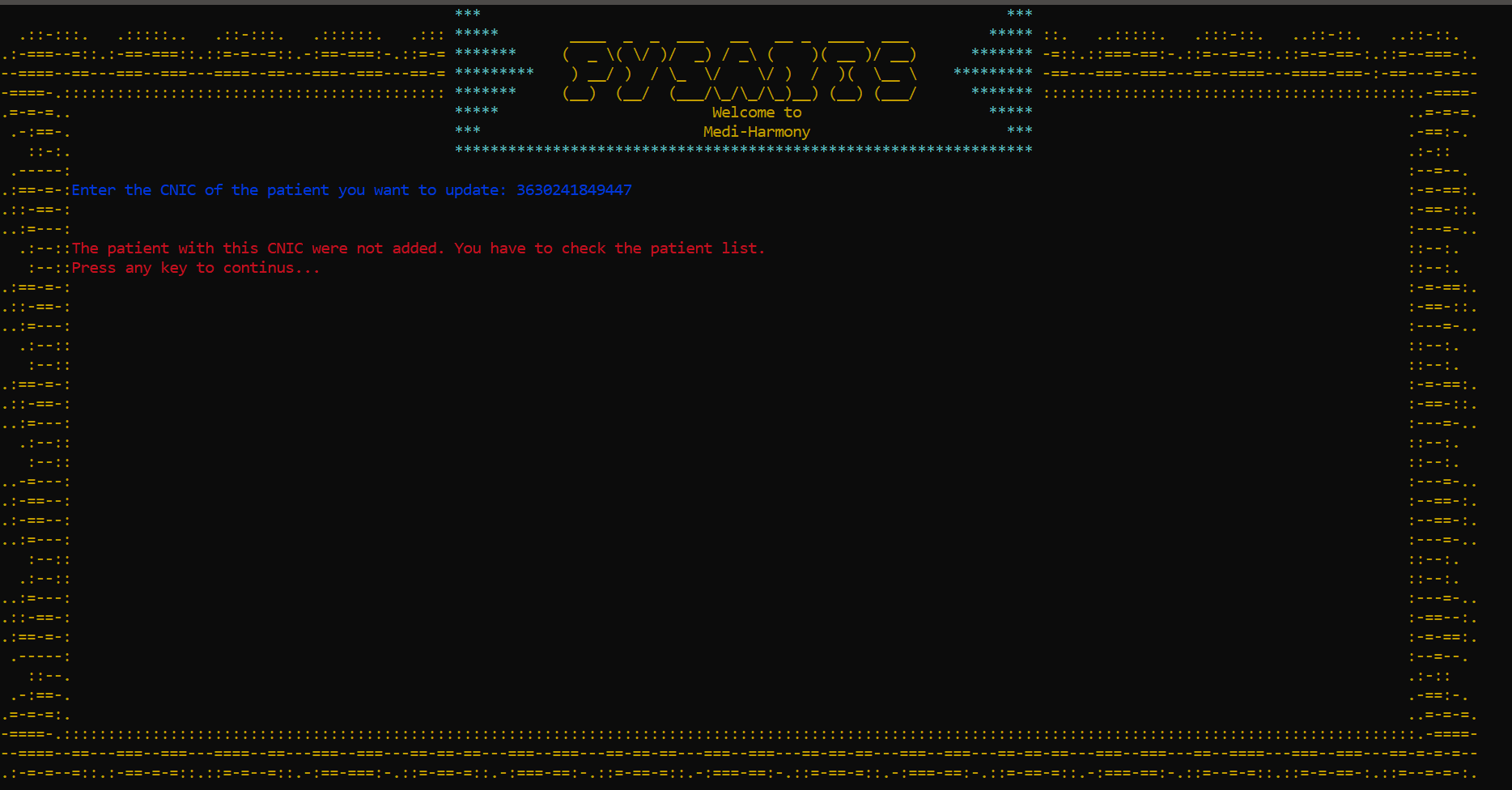


Figure -Add Manager Menu

**Change Equipment-Manager:**

‘ Figure -Change Manger Menu

### Update Patient:

Figure -Update Patient Menu

### Doctor Menu:



Figure -Doctor Menu

### View Admitted Patient:

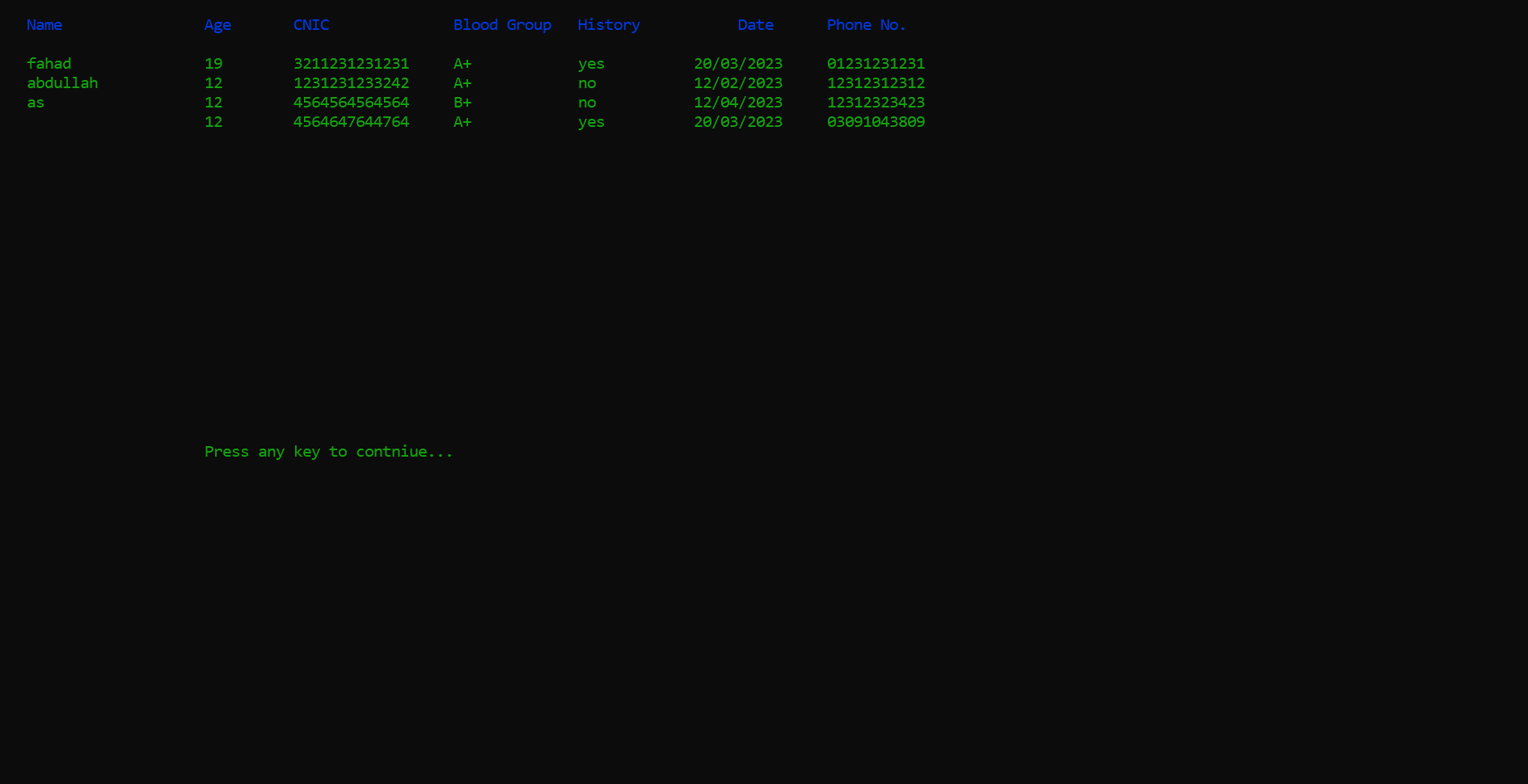


Figure -Add Doctor Menu

### View Available Beds:

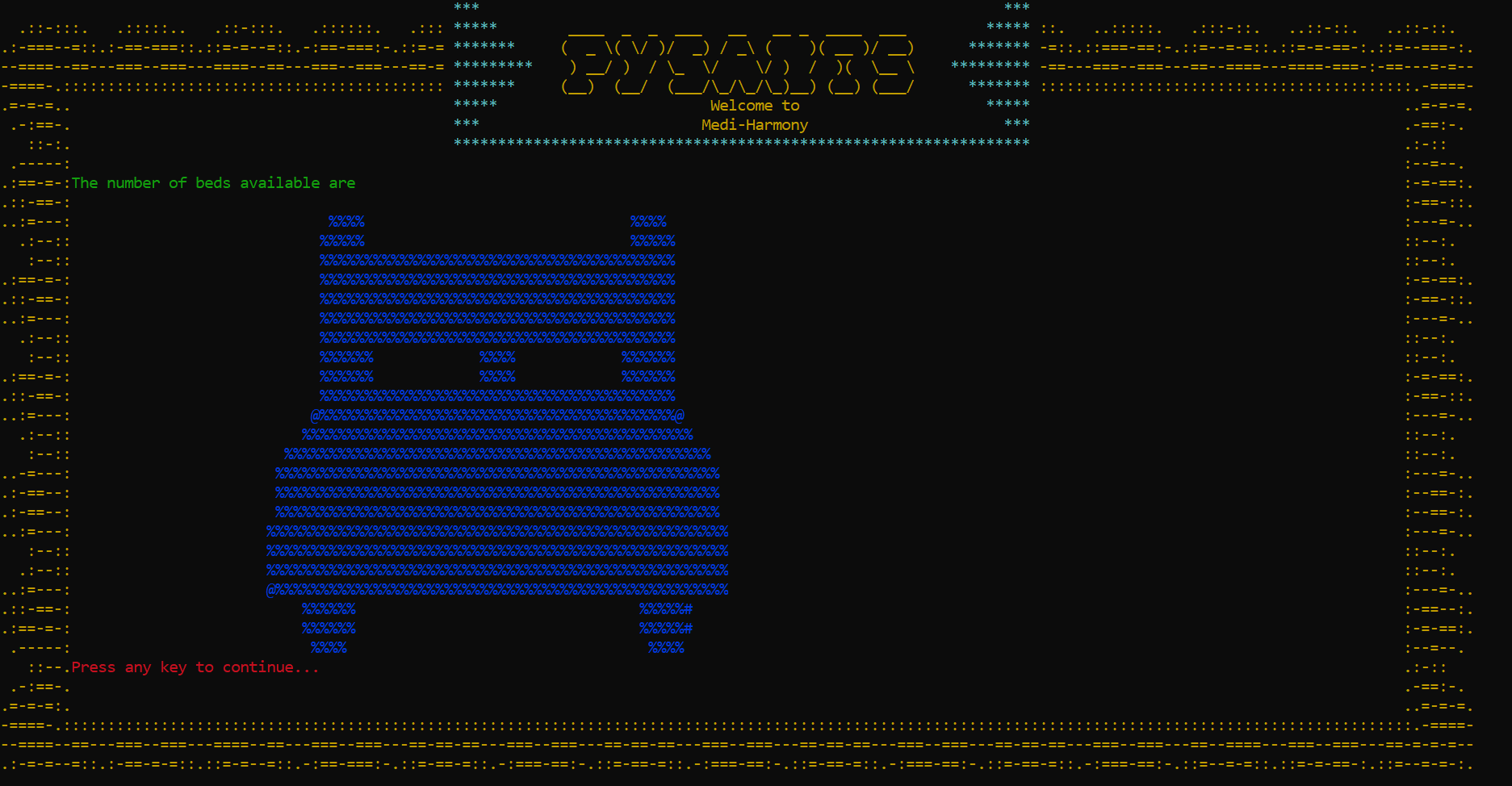


Figure -View Available Beds Menu

### View Doctor’s List:

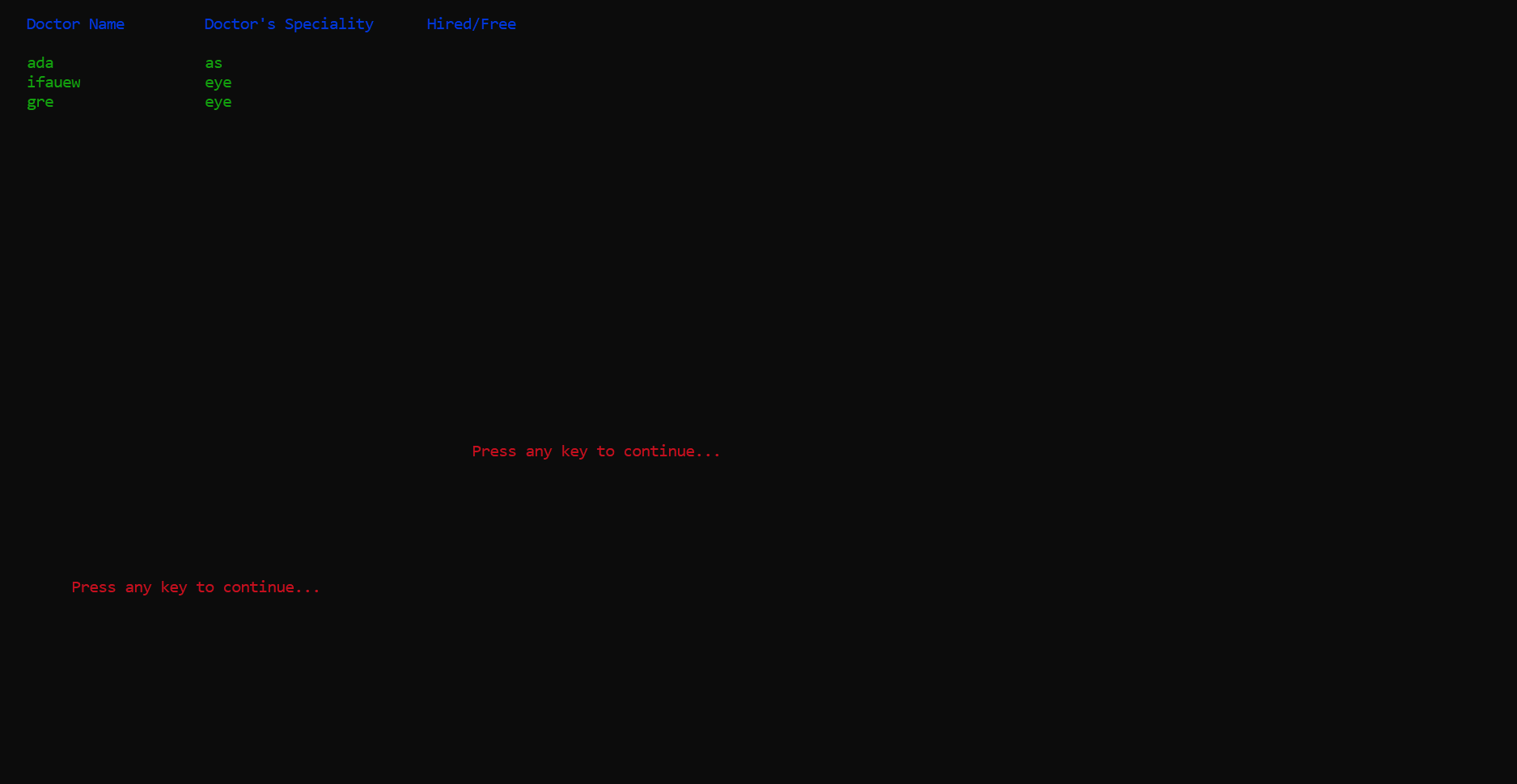


Figure -View Available Doctor Menu

### Patient Menu:

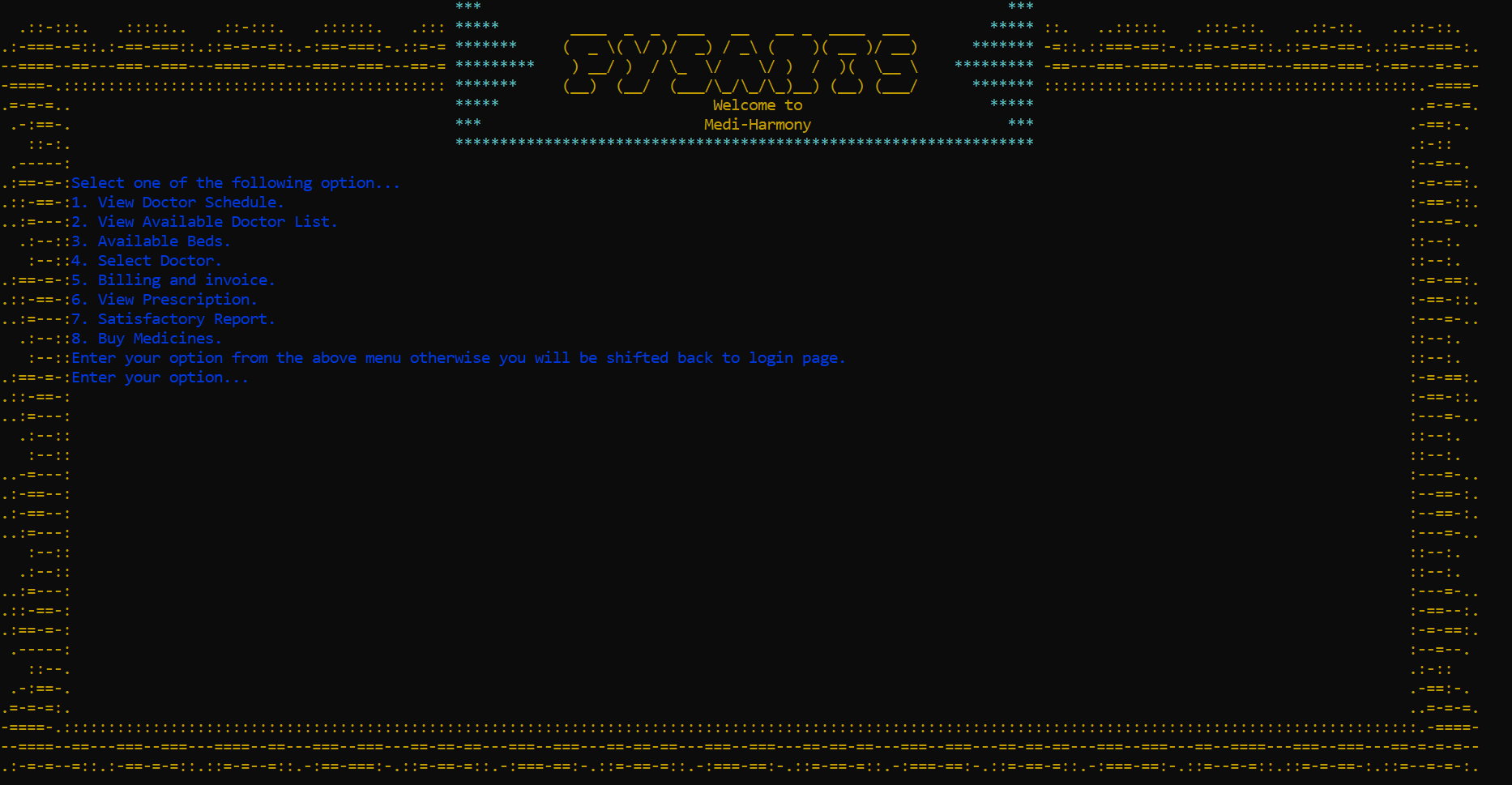


Figure -Patient Menu

### Select Doctor:



Figure -Select Doctor Menu

### Pay Bills:

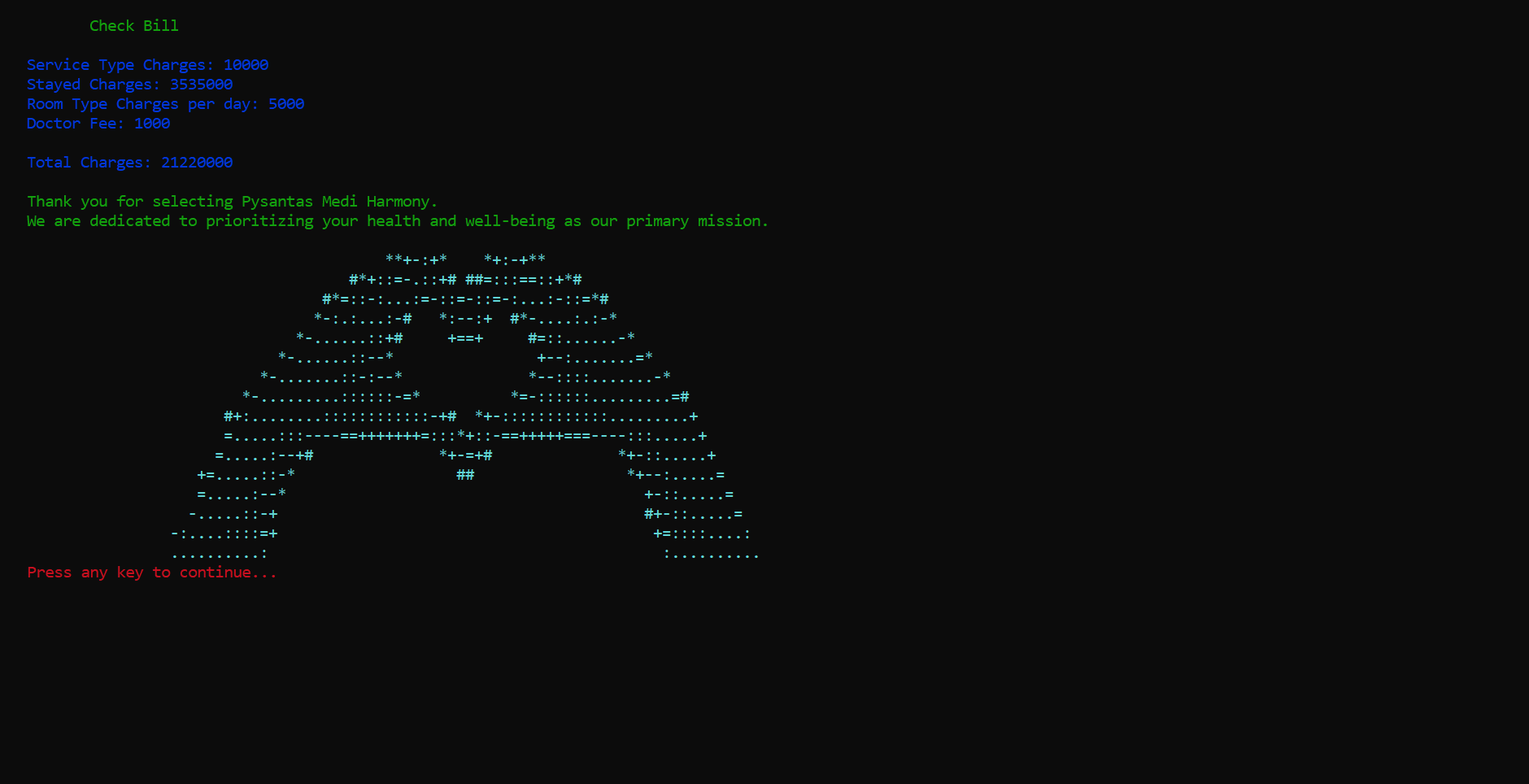


Figure -Bill Menu

### View Prescription:

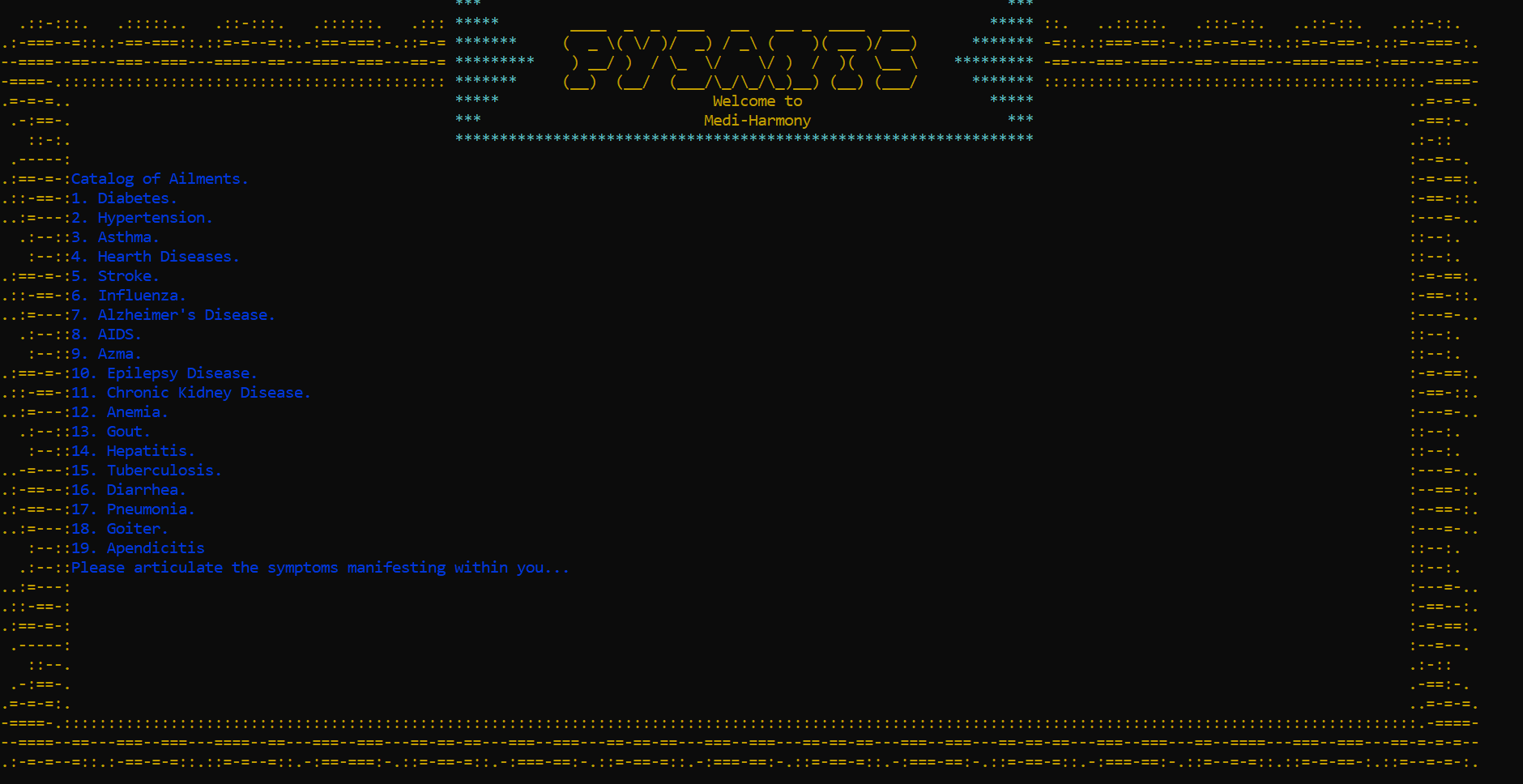


Figure -View Prescription Menu

### Patient Review:



Figure -Patient Review Menu

### Buy Medicines:

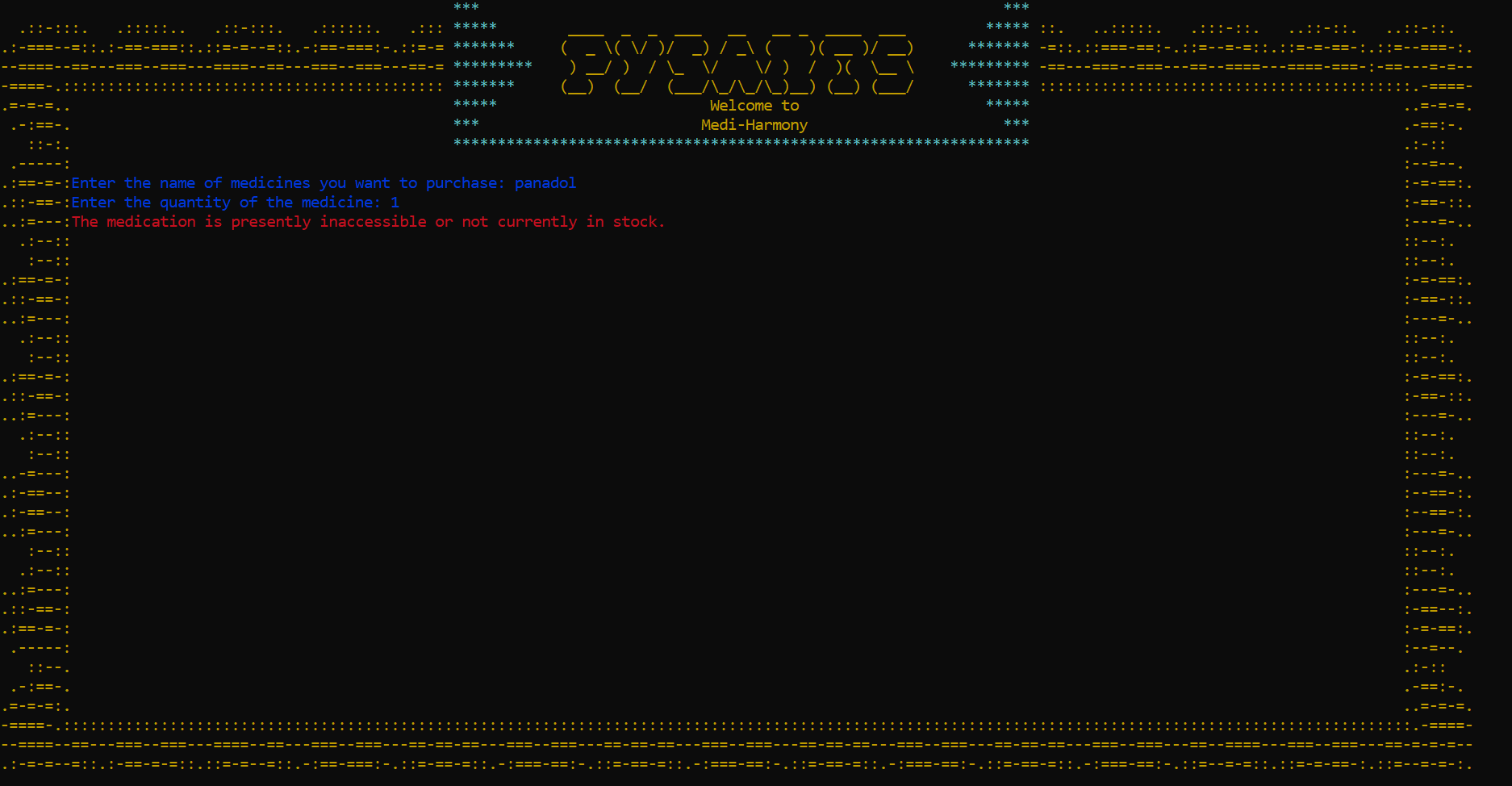


Figure -Buy Medicines Menu

### Pharmacist Menu:

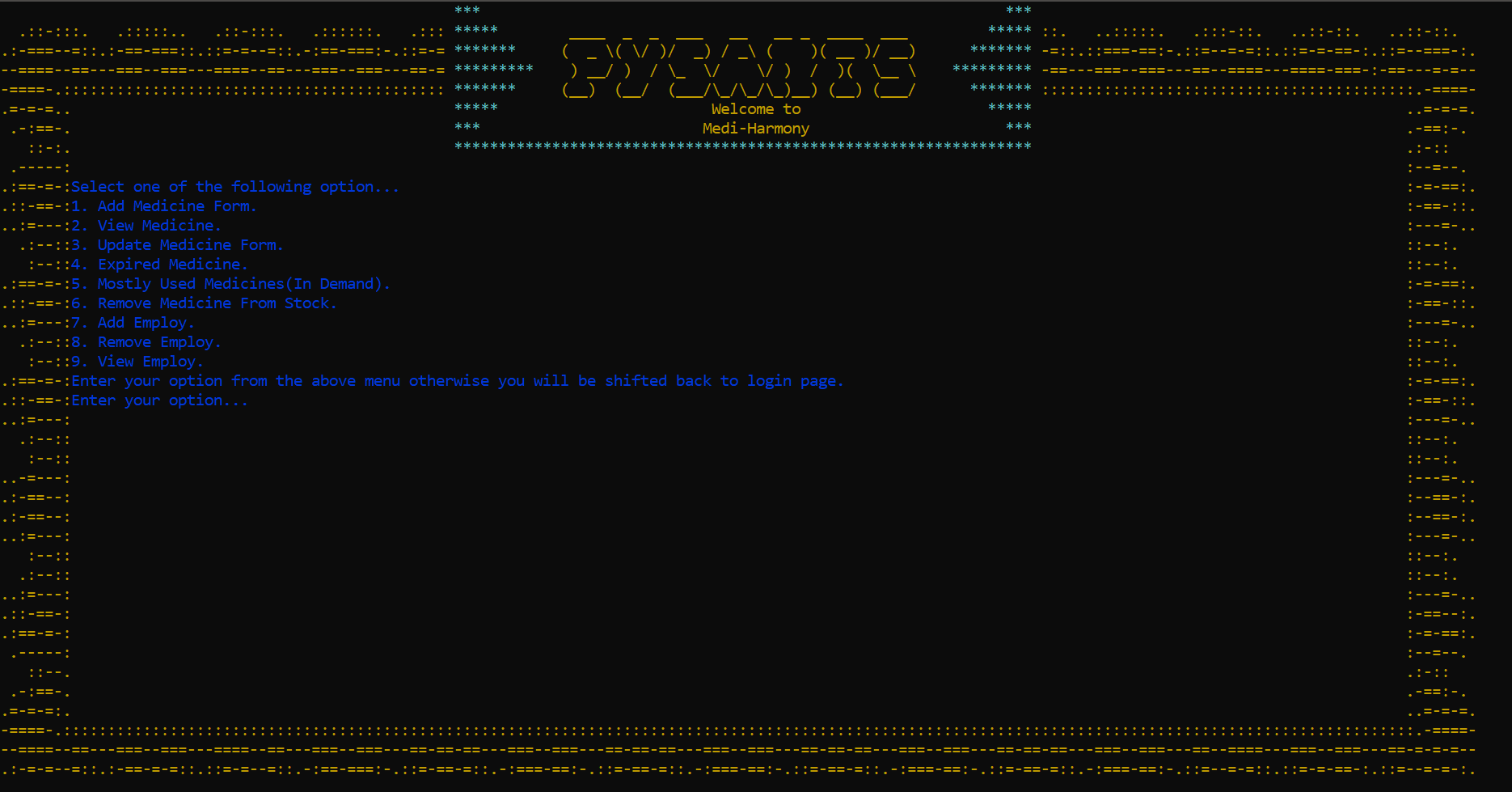


Figure -Pharmacist Menu

### Add Medicine:

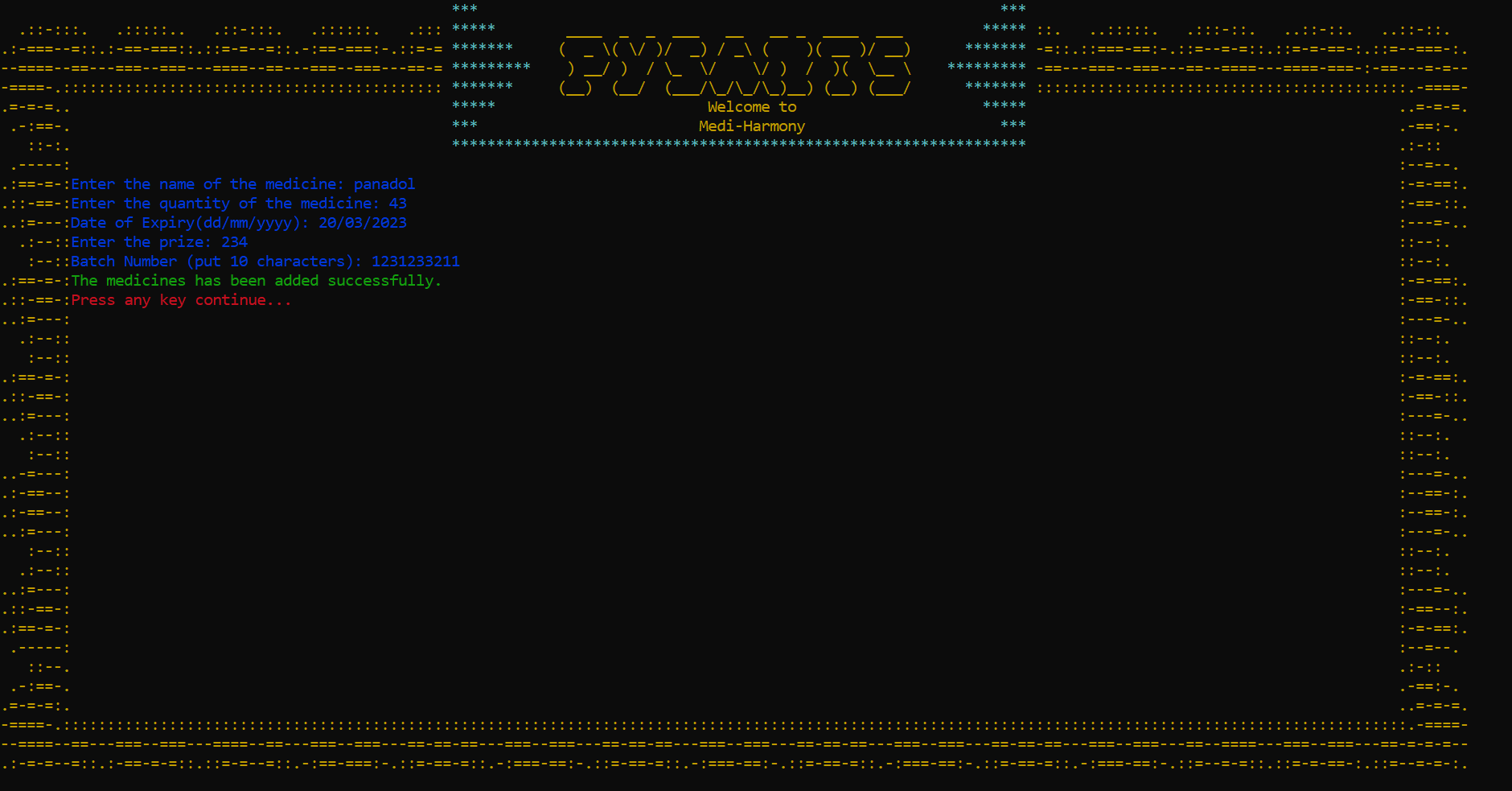


Figure -Add Medicine Menu

### View Medicines:

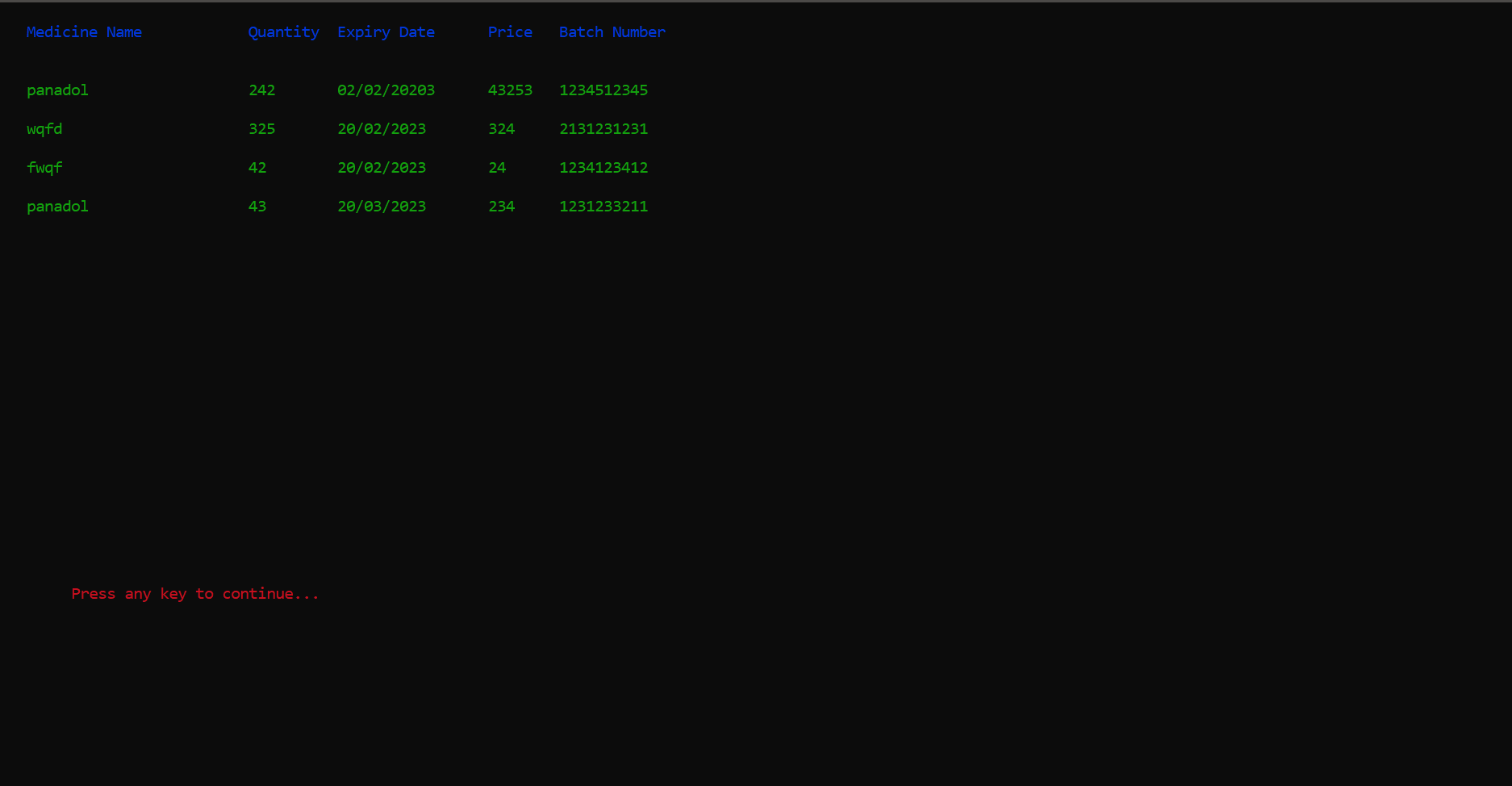


Figure -View Medicine Menu

### Update Medicine:



Figure -Update Medicine Menu

### Expired medicine:



Figure -Expired Medicine Menu

### Mostly Used Medicines:

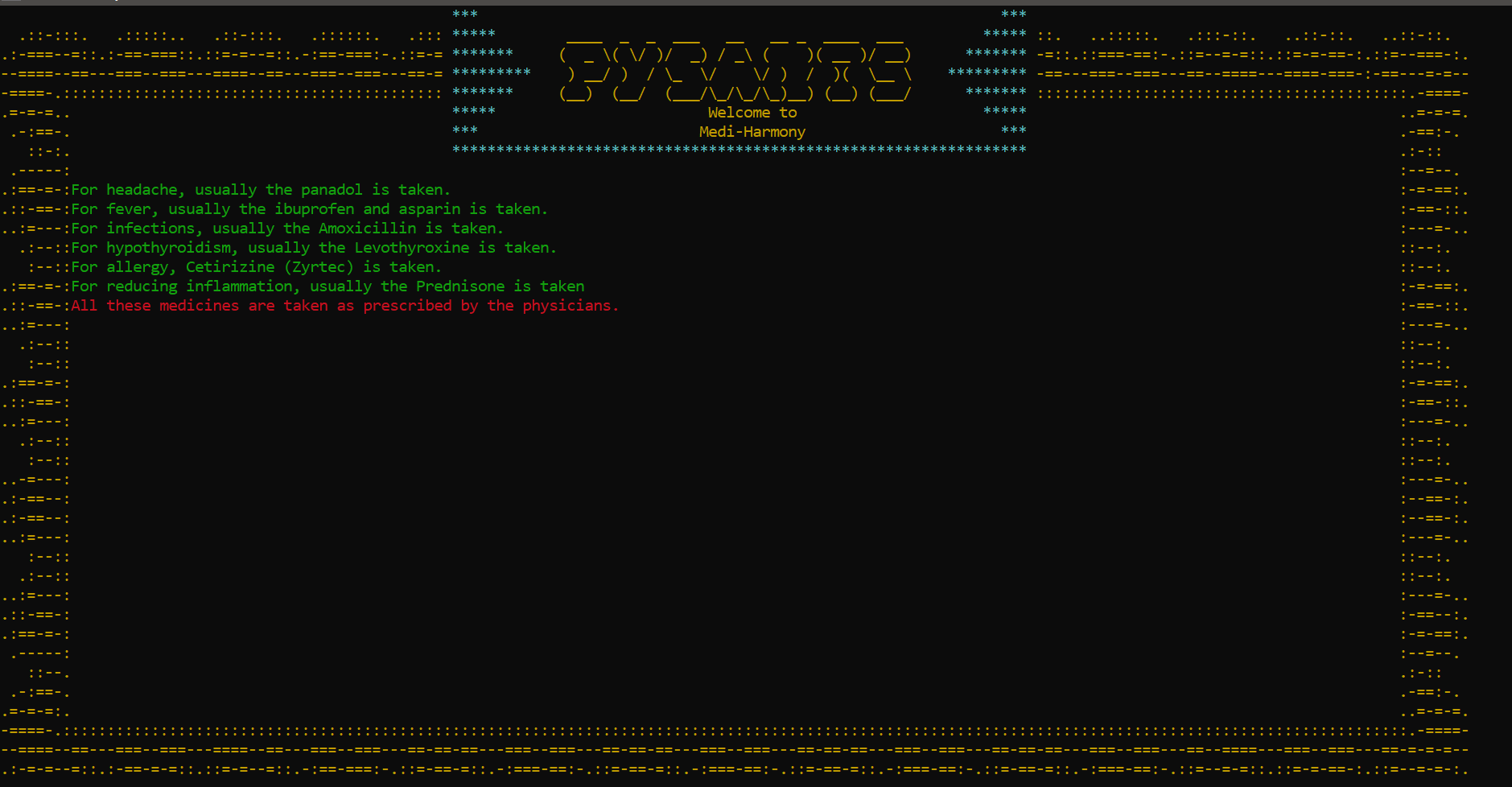


Figure -Medicine in used Menu

### Remove Medicines:

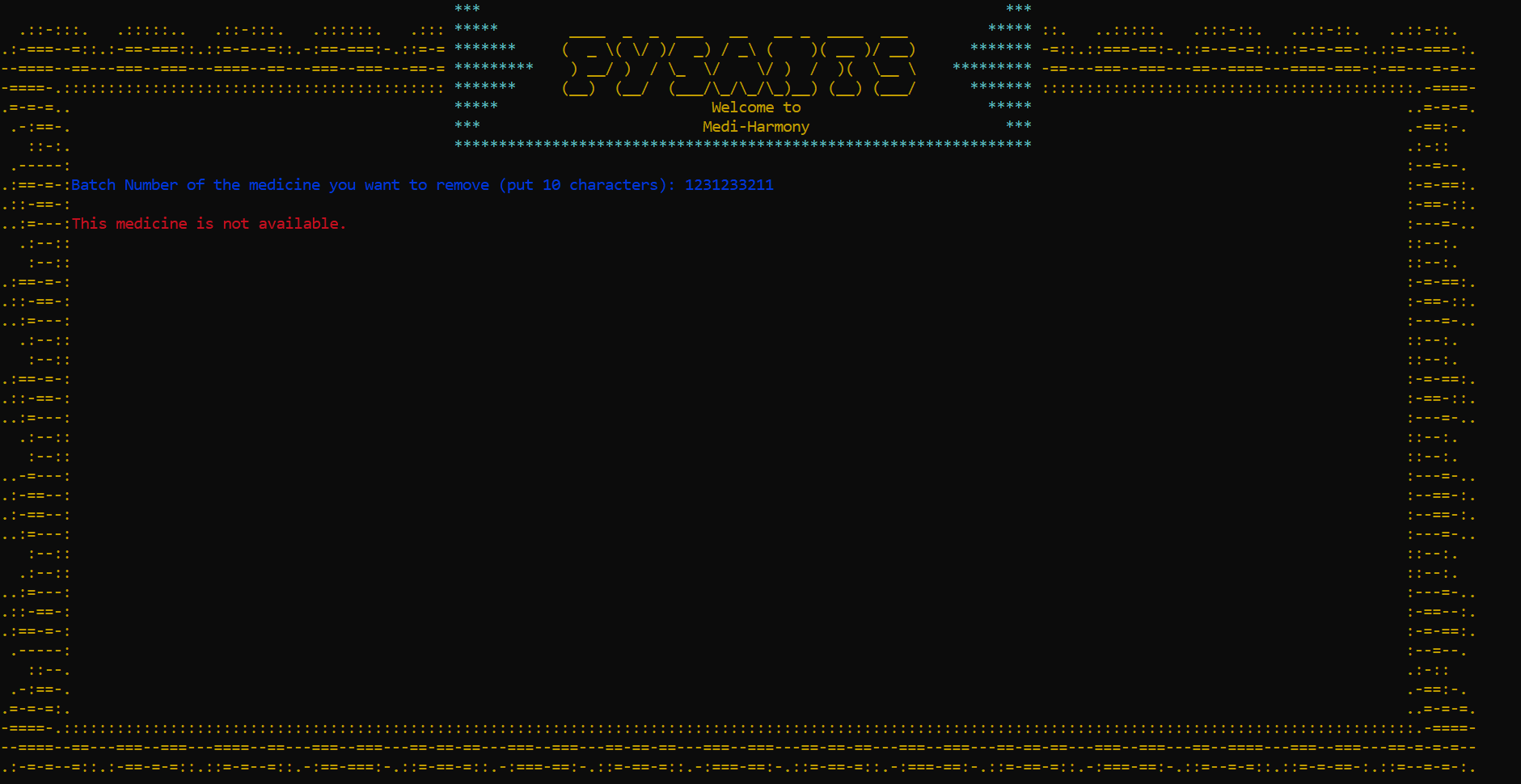


Figure -Remove Medicine Menu

### Add Employ:

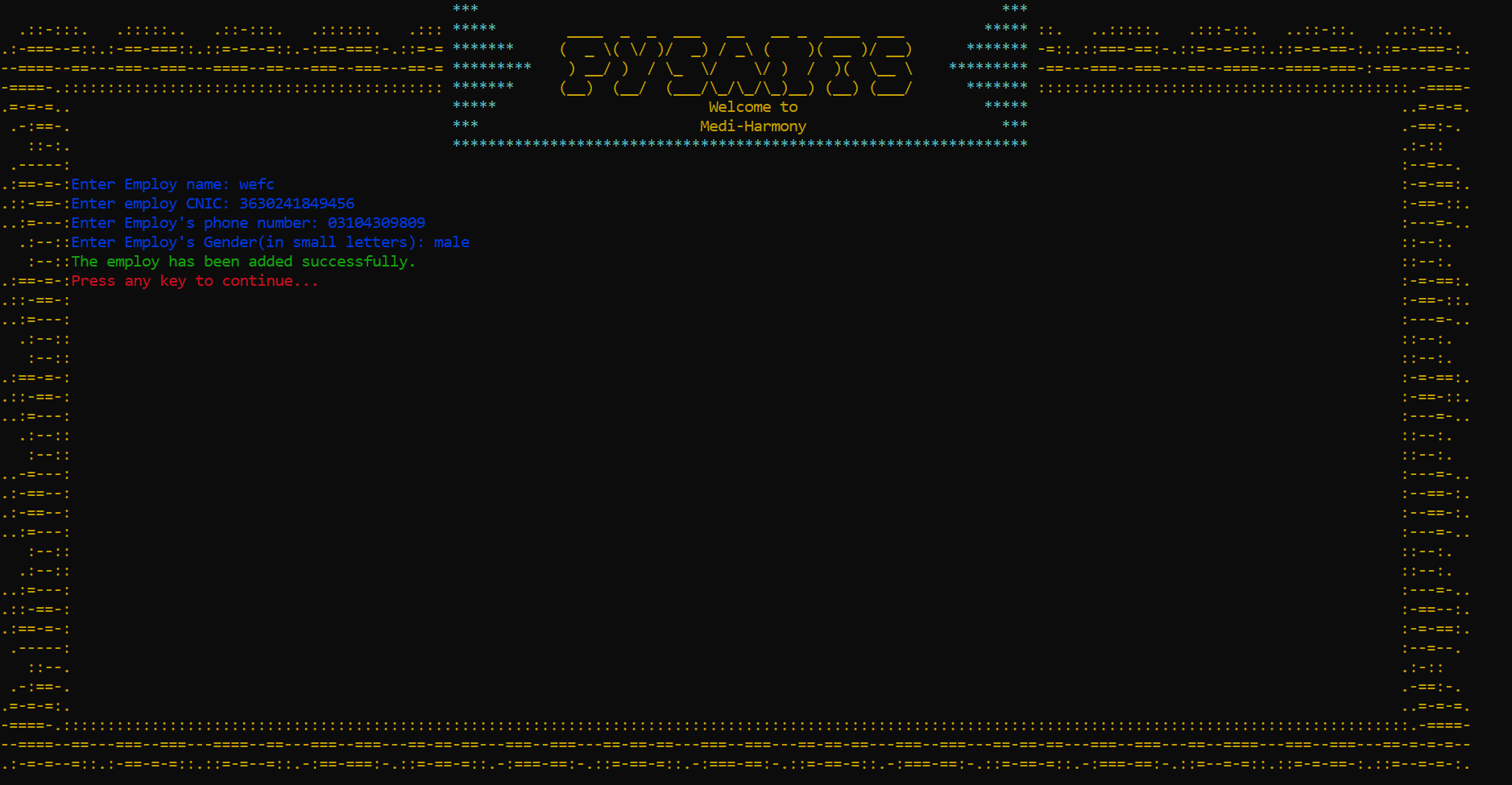


Figure -Add Employ Menu

### Remove Employ:

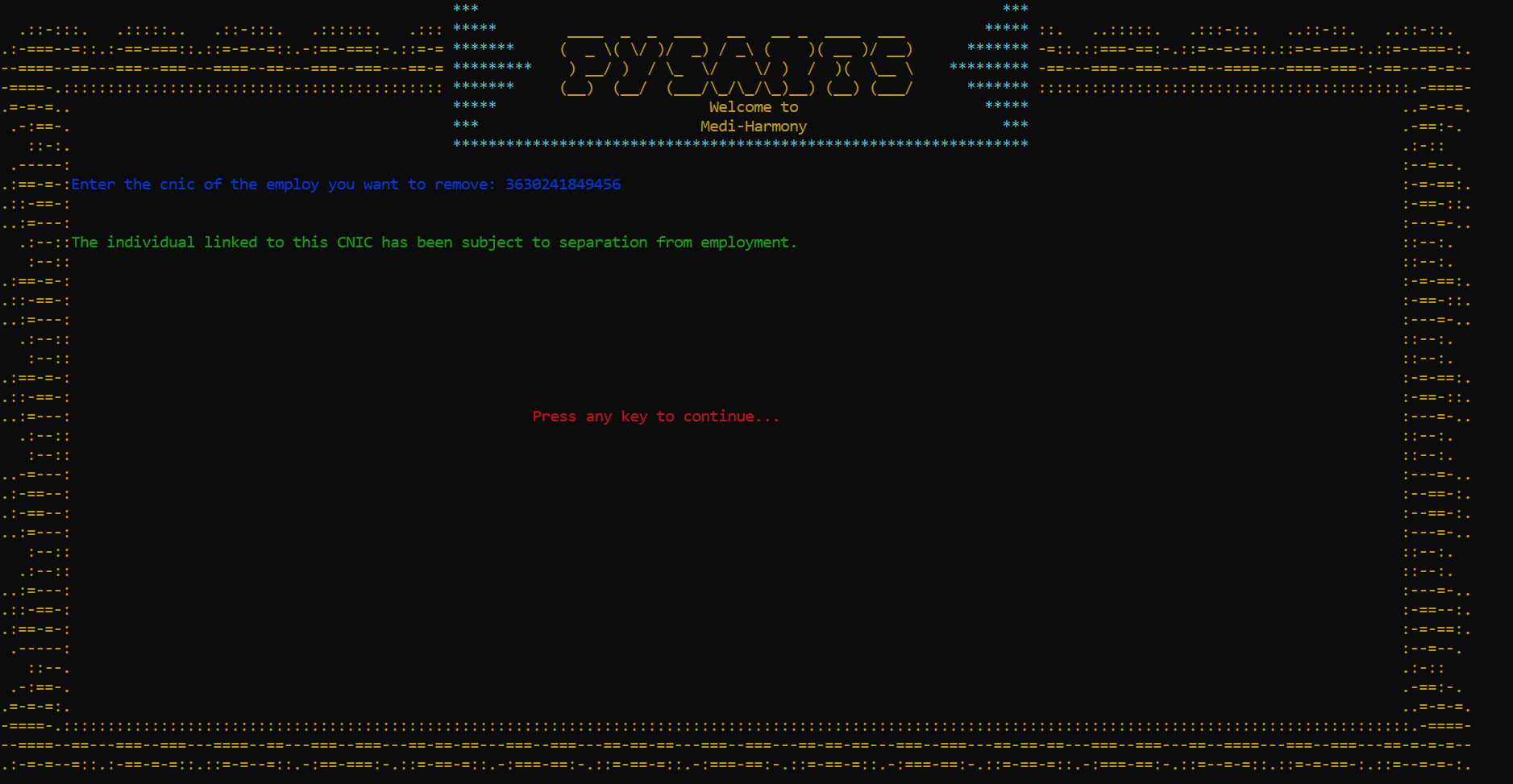


Figure -Remove Employ Menu

### View Employ:



Figure -View Employ Menu

### Equipment Manager:

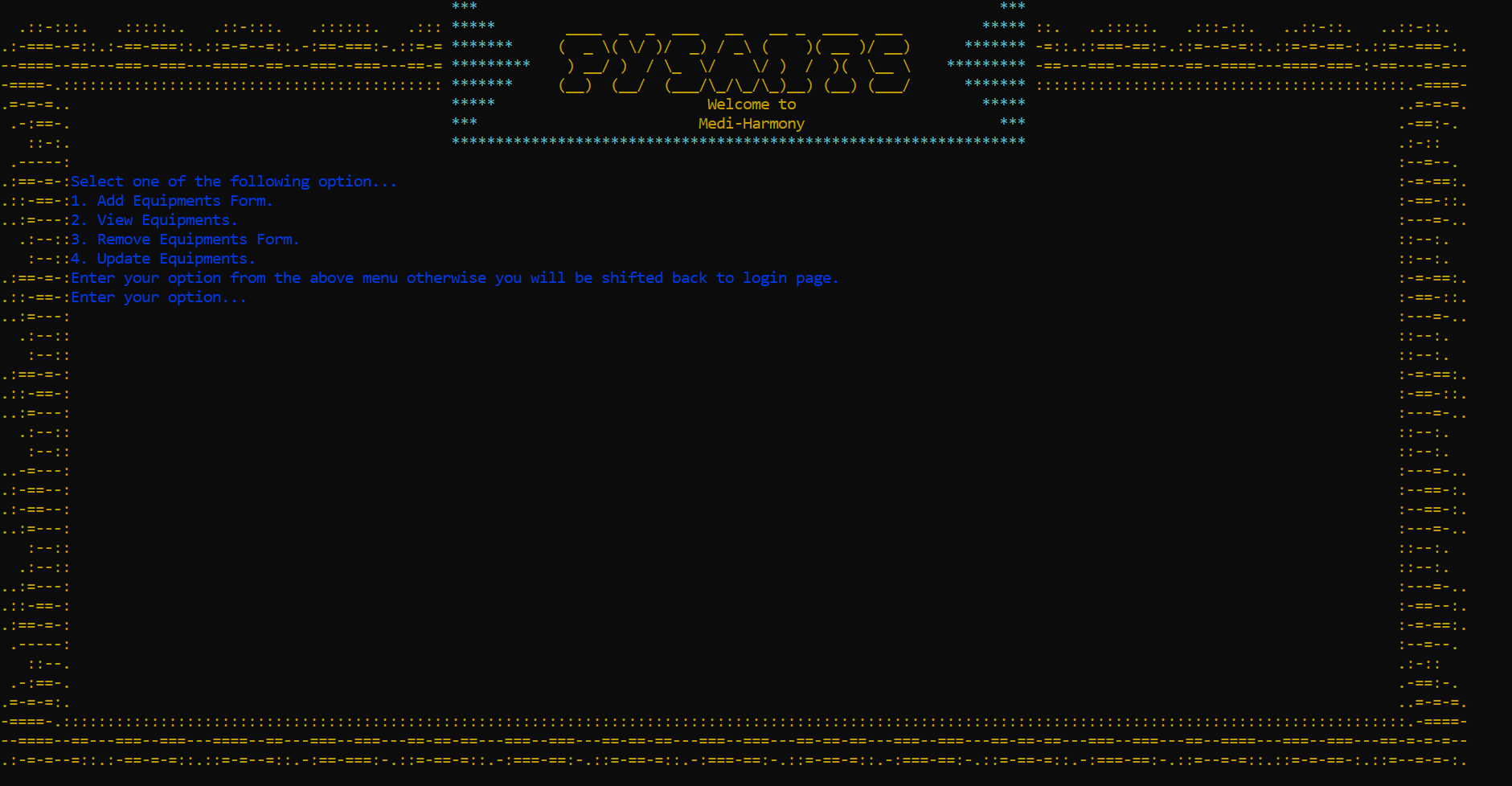


Figure -Equipment Manager Menu

### Add Equipment:



Figure -Add Equipment Menu

### View Equipment:

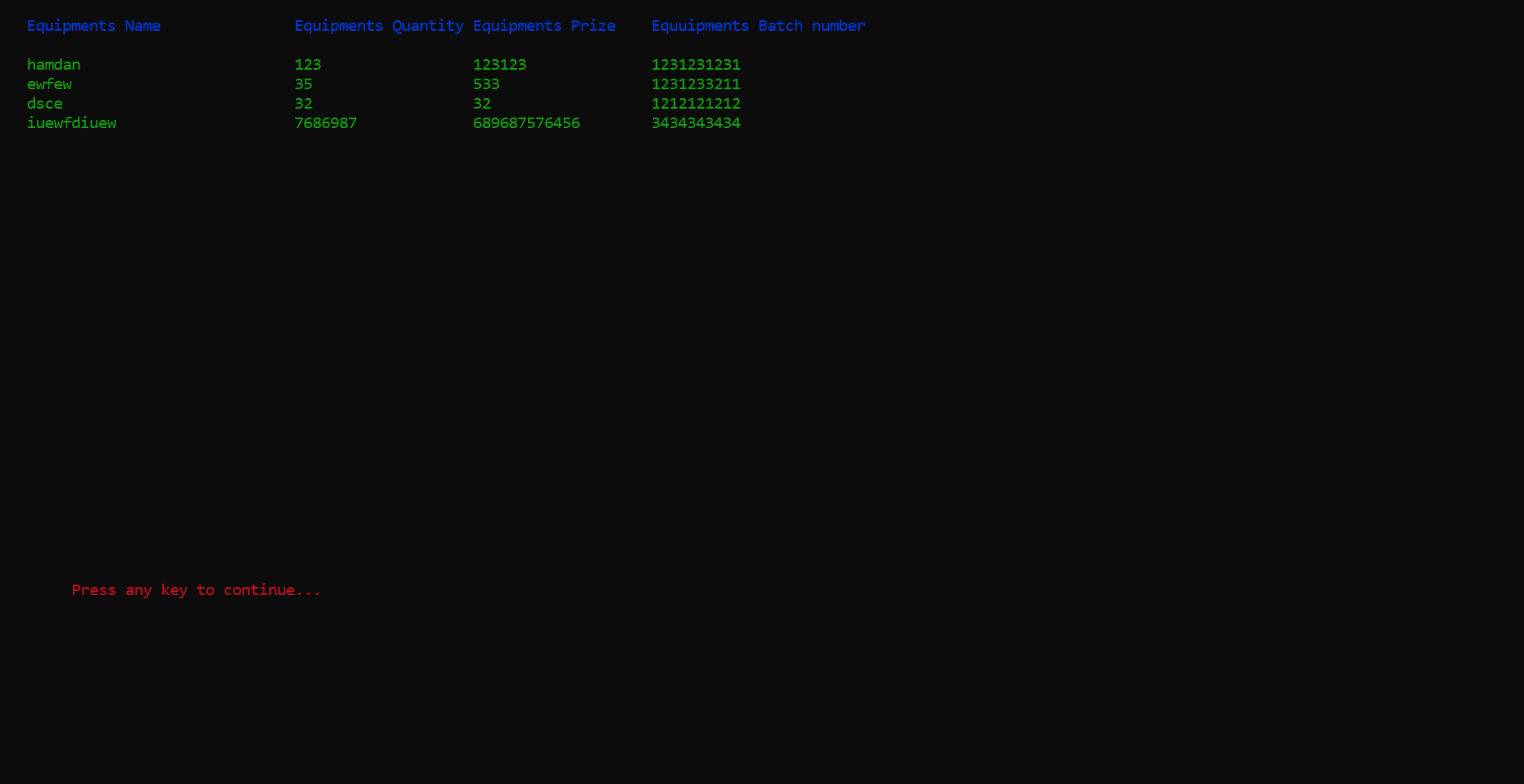


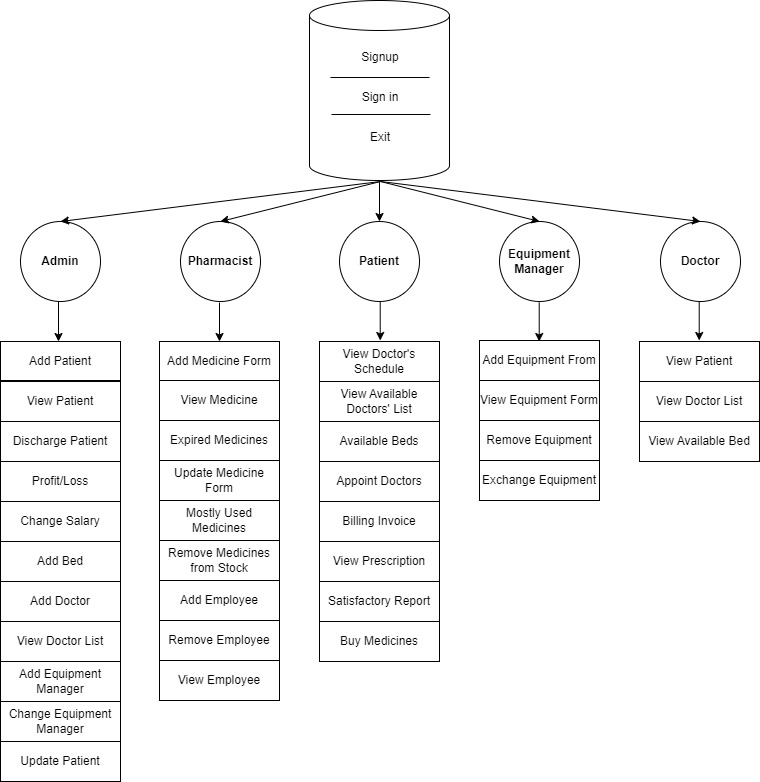
Figure -View Equipment Menu

### Remove Equipment:



Figure -Remove Equipment Menu

# ❖ Flow Chart:



# ❖ Data Structure:

Parallel 1-dimensional array data structure is implied in the current version of CrossFit Studios. Data storing is organized in an orderly fashion way, having related data stored in a single file using CSV format and different data stored in different files. Files are named in an organized way to improve readability.

* string filename;
* int aminomethane=0;
* int adminislamabad=0;
* int adminlahore=0;
* //variables declaration
* int addpatientindex=0;
* int viewpatientindex=0;
* string username[100];
* string userpassword[100];
* string userrole[100];
* string firstname[100];
* string secondname[100];
* string gender[100];
* string city[100];
* string nationality[100];
* string contactnumber[100];
* string cnic[100];
* string names;
* string passwords;
* string adminoption;
* string managername;
* string option;
* string patientname[100];
* string patientage[100];
* string patientcnic[100];
* string patientbloodgroup[100];
* string patienthistory[100];
* string visitdatepatient[100];
* string patientphonenumber[100];
* string rolereturn;
* string dischargepatientcnic;
* string patientcharges;
* string doctorsalary;
* string managersalary;
* string bedadd;
* string doctorname[100];
* string doctorspeciality[100];
* string equipmentmanagername;
* string buymedicinequantity="0";
* string purchasemedicines;
* string equipmentmanagergender;
* string equipmentmanagercnic;
* string selectdoctorname;
* string selectdoctorspeciality;
* string updatepatientcnic;
* string updatemedicinebatchnumber;
* string servicetype[100];
* string roomtype[100];
* string daysstayes[100];
* string review[100];
* string medicinename[100];
* string medicinequantity[100];
* string medicineexpirydate[100];
* string medicineprice[100];
* string medicinebatchnumber[100];
* string customername[100];
* string currentdate[100];
* string purchasemedicinename;
* string purchasemedicinequantity[100];
* string employname[100];
* string employcnic[100];
* string employphonenumber[100];
* string employgender[100];
* string updateequipmentbatchnumber;
* string removeemploycnic;
* int purchasemedicineindex=0;
* int addmedicineindex=0;
* int adddoctorindex=0;
* int addbillingindex;
* int managercheck=0;
* int addequipmentindex=0;
* string equipmentname[100];
* string equipmentquantity[100];
* string equipmentprice[100];
* string equipmentbatchnumber[100];
* int addemployindex=0;
* string hire [100];
* int size = 0;
* int addpatientindexislamabad=0;
* int viewpatientindexislamabad=0;
* string usernameislamabad[100];
* string userpasswordislamabad[100];
* string userroleislamabad[100];
* string firstnameislamabad[100];
* string secondnameislamabad[100];
* string genderislamabad[100];
* string cityislamabad[100];
* string nationalityislamabad[100];
* string contactnumberislamabad[100];
* string cnicislamabad[100];
* string namesislamabad;
* string passwordsislamabad;
* string adminoptionislamabad;
* string managernameislamabad;
* string optionislamabad;
* string patientnameislamabad[100];
* string patientageislamabad[100];
* string patientcnicislamabad[100];
* string patientbloodgroupislamabad[100];
* string patienthistoryislamabad[100];
* string visitdatepatientislamabad[100];
* string patientphonenumberislamabad[100];
* string dischargepatientcnicislamabad;
* string patientchargesislamabad;
* string doctorsalaryislamabad;
* string managersalaryislamabad;
* string bedaddislamabad;
* string doctornameislamabad[100];
* string doctorspecialityislamabad[100];
* string equipmentmanagernameislamabad;
* string buymedicinequantityislamabad="0";
* string purchasemedicinesislamabad;
* string equipmentmanagergenderislamabad;
* string equipmentmanagercnicislamabad;
* string selectdoctornameislamabad;
* string selectdoctorspecialityislamabad;
* string updatepatientcnicislamabad;
* string updatemedicinebatchnumberislamabad;
* string servicetypeislamabad[100];
* string roomtypeislamabad[100];
* string daysstayesislamabad[100];
* string reviewislamabad[100];
* string medicinenameislamabad[100];
* string medicinequantityislamabad[100];
* string medicineexpirydateislamabad[100];
* string medicinepriceislamabad[100];
* string medicinebatchnumberislamabad[100];
* string customernameislamabad[100];
* string currentdateislamabad[100];
* string purchasemedicinenameislamabad;
* string purchasemedicinequantityislamabad[100];
* string employnameislamabad[100];
* string employcnicislamabad[100];
* string employphonenumberislamabad[100];
* string employgenderislamabad[100];
* string updateequipmentbatchnumberislamabad;
* string removeemploycnicislamabad;
* int purchasemedicineindexislamabad=0;
* int addmedicineindexislamabad=0;
* int adddoctorindexislamabad=0;
* int addbillingindexislamabad;
* int managercheckislamabad=0;
* int addequipmentindexislamabad=0;
* string equipmentnameislamabad[100];
* string equipmentquantityislamabad[100];
* string equipmentpriceislamabad[100];
* string equipmentbatchnumberislamabad[100];
* int addemployindexislamabad=0;
* string hireislamabad[100];
* int sizeislamabad = 0;
* int addpatientindexlahore=0;
* int viewpatientindexlahore=0;
* string usernamelahore[100];
* string userpasswordlahore[100];
* string userrolelahore[100];
* string firstnamelahore[100];
* string secondnamelahore[100];
* string genderlahore[100];
* string citylahore[100];
* string nationalitylahore[100];
* string contactnumberlahore[100];
* string cniclahore[100];
* string nameslahore;
* string passwordslahore;
* string adminoptionlahore;
* string managernamelahore;
* string optionlahore;
* string patientnamelahore[100];
* string patientagelahore[100];
* string patientcniclahore[100];
* string patientbloodgrouplahore[100];
* string patienthistorylahore[100];
* string visitdatepatientlahore[100];
* string patientphonenumberlahore[100];
* string dischargepatientcniclahore;
* string patientchargeslahore;
* string doctorsalarylahore;
* string managersalarylahore;
* string bedaddlahore;
* string doctornamelahore[100];
* string doctorspecialitylahore[100];
* string equipmentmanagernamelahore;
* string buymedicinequantitylahore="0";
* string purchasemedicineslahore;
* string equipmentmanagergenderlahore;
* string equipmentmanagercniclahore;
* string selectdoctornamelahore;
* string selectdoctorspecialitylahore;
* string updatepatientcniclahore;
* string updatemedicinebatchnumberlahore;
* string servicetypelahore[100];
* string roomtypelahore[100];
* string daysstayeslahore[100];
* string reviewlahore[100];
* string medicinenamelahore[100];
* string medicinequantitylahore[100];
* string medicineexpirydatelahore[100];
* string medicinepricelahore[100];
* string medicinebatchnumberlahore[100];
* string customernamelahore[100];
* string currentdatelahore[100];
* string purchasemedicinenamelahore;
* string purchasemedicinequantitylahore[100];
* string employnamelahore[100];
* string employcniclahore[100];
* string employphonenumberlahore[100];
* string employgenderlahore[100];
* string updateequipmentbatchnumberlahore;
* string removeemploycniclahore;
* int purchasemedicineindexlahore=0;
* int addmedicineindexlahore=0;
* int adddoctorindexlahore=0;
* int addbillingindexlahore;
* int managerchecklahore=0;
* int addequipmentindexlahore=0;
* string equipmentnamelahore[100];
* string equipmentquantitylahore[100];
* string equipmentpricelahore[100];
* string equipmentbatchnumberlahore[100];
* int addemployindexlahore=0;
* string hirelahore[100];
* int sizelahore = 0;
* int addpatientindexmultan=0;
* int viewpatientindexmultan=0;
* string usernamemultan[100];
* string userpasswordmultan[100];
* string userrolemultan[100];
* string firstnamemultan[100];
* string secondnamemultan[100];
* string gendermultan[100];
* string citymultan[100];
* string nationalitymultan[100];
* string contactnumbermultan[100];
* string cnicmultan[100];
* string namesmultan;
* string passwordsmultan;
* string adminoptionmultan;
* string managernamemultan;
* string optionmultan;
* string patientnamemultan[100];
* string patientagemultan[100];
* string patientcnicmultan[100];
* string patientbloodgroupmultan[100];
* string patienthistorymultan[100];
* string visitdatepatientmultan[100];
* string patientphonenumbermultan[100];
* string dischargepatientcnicmultan;
* string patientchargesmultan;
* string doctorsalarymultan;
* string managersalarymultan;
* string bedaddmultan;
* string doctornamemultan[100];
* string doctorspecialitymultan[100];
* string equipmentmanagernamemultan;
* string buymedicinequantitymultan="0";
* string purchasemedicinesmultan;
* string equipmentmanagergendermultan;
* string equipmentmanagercnicmultan;
* string selectdoctornamemultan;
* string selectdoctorspecialitymultan;
* string updatepatientcnicmultan;
* string updatemedicinebatchnumbermultan;
* string servicetypemultan[100];
* string roomtypemultan[100];
* string daysstayesmultan[100];
* string reviewmultan[100];
* string medicinenamemultan[100];
* string medicinequantitymultan[100];
* string medicineexpirydatemultan[100];
* string medicinepricemultan[100];
* string medicinebatchnumbermultan[100];
* string customernamemultan[100];
* string currentdatemultan[100];
* string purchasemedicinenamemultan;
* string purchasemedicinequantitymultan[100];
* string employnamemultan[100];
* string employcnicmultan[100];
* string employphonenumbermultan[100];
* string employgendermultan[100];
* string updateequipmentbatchnumbermultan;
* string removeemploycnicmultan;
* int purchasemedicineindexmultan=0;
* int addmedicineindexmultan=0;
* int adddoctorindexmultan=0;
* int addbillingindexmultan;
* int managercheckmultan=0;
* int addequipmentindexmultan=0;
* string equipmentnamemultan[100];
* string equipmentquantitymultan[100];
* string equipmentpricemultan[100];
* string equipmentbatchnumbermultan[100];
* int addemployindexmultan=0;
* string hiremultan[100];
* int sizemultan = 0;

# ❖ Function Prototypes:

Medi Harmony is built by keeping Single Responsibility of Functions

in view. Each function is tried best to be independent of other function. Following are the Function Prototypes Used:

* string login();
* void thankyou();
* void clear();
* void hideCursor();
* string adminmenu();
* int check\_integer(string num);
* void prescription();
* void addpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient);
* void dischargepatient(string dischargepatientcnic, int addpatientindex, string patientcnic[], string patientname[], string patientage[], string patientbloodgroup[], string patienthistory[], string patientphonenumber[], string visitdatepatient[], string filenameaddpatient);
* void viewpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visibledatepatient[], string patientphonenumber[]);
* void changingsalary(string doctorsalary, string managersalary, string filenamecahngingsalary);
* void profitloss(string patientcharges, int addpatientindex, string filenameprofitloss);
* void addbed(string bedadd, string filenameaddbed);
* void adddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor);
* void viewdoctor(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex);
* void changeequipmentmanager(int &managercheck, string &equipmentmanagername, string &equipmentmanagergender, string &equipmentmanagercnic, string filenameaddequipmentmanager);
* void addequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameaddequipmentmanager);
* void updatepatient(string updatepatientcnic, string patientcnic[], int &addpatientindex, string patientname[], string patientage[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient);
* string pateintmenu();
* void doctorschedule(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex);
* void viewdoctorslist(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex);
* void selectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[]);
* void availablebeds(string bedadd, int &addpatientindex);
* void billinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[]);
* void reviewhearth();
* void bilslip(string servicetype[], string daysstayes[], string roomtype[], int &addbillingindex);
* string viewprescription();
* void patientreview(string review[], int &addpatientindex);
* string pharmacymenu();
* void addmedicineform(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine);
* void viewmedicineform(string buymedicinequantity, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenameaddmedicine);
* void updatemedicineform(string updatemedicinebatchnumber, string medicinebatchnumber[], int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string filenameaddmedicine);
* string doctormenu();
* void buymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines);
* bool charactermCheck(string characterCheck);
* void expiredmedicine(int addmedicineindex, string currentdate[], string medicineexpirydate[], string medicinename[]);
* bool recursionCheck(string check, string recursion[], int size);
* bool checkenter(string check);
* bool gendercheck(string check);
* bool digitscheck(string digits);
* bool checkshiftin(string check[]);
* void addmanager(string managername, string managercnic);
* void medicineindemand();
* int convertDatetodays(string date);
* bool isExpired(string currentDate, string expiryDate);
* void removemedicine(string updatemedicinebatchnumber, int &addmedicineindex, string medicinebatchnumber[], string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string filenameaddmedicine);
* void addemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[]);
* void removeemploy(int &addemployindex, string removeemploycnic, string employcnic[], string employname[], string employphonenumber[], string employgender[]);
* void viewemploy(int &addemployindex, string employname[], string employcnic[], string employphonenumber[], string employgender[]);
* string equipmentmenu();
* void updateequipment(string updateequipmentbatchnumber, int &addequipmentindex, string equipmentbatchnumber[], string equipmentname[], string equipmentquantity[], string equipmentprize[], string filenameaddequipment);
* void addequipmentform(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment);
* void removeequipmentform(int &addequipmentindex, string updateequipmentbatchnumber, string equipmentbatchnumber[], string equipmentname[], string equipmentprize[], string equipmentquantity[], string filenameaddequipment);
* string branchmenu();

# ❖ Weakness:

There’s a weakness in the system in viewing the expired medicines. As they are unable to check the expiry of the medicines. Sometimes, it crashes.

# ❖ Future Directions:

I want to implement Navigation system for choosing option and

making the application more user-friendly by reducing the number of typed inputs required. Also, I Aim to use GUI to make Medi Harmony market available as current demand of market is GUI based applications.

# ❖ Code:

#include <iostream>

#include<fstream>

#include <string>

#include <conio.h>

#include <windows.h>

#include <ctime>

#include<cctype>

#include<limits>

using namespace std;

//functions prototype starts

void topheading();

void header();

void delay(int milliseconds);

void showLoadingAnimation();

void gotoxy(int x, int y);

void multaninterface();

void islamabadinterface();

void lahoreinterface();

string login();

void thankyou();

void clear();

void hideCursor();

string adminmenu();

int check\_integer(string num);

void prescription();

void addpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient);

void dischargepatient(string dischargepatientcnic, int addpatientindex, string patientcnic[], string patientname[], string patientage[], string patientbloodgroup[], string patienthistory[], string patientphonenumber[], string visitdatepatient[], string filenameaddpatient);

void viewpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visibledatepatient[], string patientphonenumber[]);

void changingsalary(string doctorsalary, string managersalary, string filenamecahngingsalary);

void profitloss(string patientcharges, int addpatientindex, string filenameprofitloss);

void addbed(string bedadd, string filenameaddbed);

void adddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor);

void viewdoctor(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex);

void changeequipmentmanager(int &managercheck, string &equipmentmanagername, string &equipmentmanagergender, string &equipmentmanagercnic, string filenameaddequipmentmanager);

void addequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameaddequipmentmanager);

void updatepatient(string updatepatientcnic, string patientcnic[], int &addpatientindex, string patientname[], string patientage[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient);

string pateintmenu();

void doctorschedule(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex);

void viewdoctorslist(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex);

void selectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[]);

void availablebeds(string bedadd, int &addpatientindex);

void billinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[]);

void reviewhearth();

void bilslip(string servicetype[], string daysstayes[], string roomtype[], int &addbillingindex);

string viewprescription();

void patientreview(string review[], int &addpatientindex);

string pharmacymenu();

void addmedicineform(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine);

void viewmedicineform(string buymedicinequantity, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenameaddmedicine);

void updatemedicineform(string updatemedicinebatchnumber, string medicinebatchnumber[], int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string filenameaddmedicine);

string doctormenu();

void buymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines);

bool charactermCheck(string characterCheck);

void expiredmedicine(int addmedicineindex, string currentdate[], string medicineexpirydate[], string medicinename[]);

bool recursionCheck(string check, string recursion[], int size);

bool checkenter(string check);

bool gendercheck(string check);

bool digitscheck(string digits);

bool checkshiftin(string check[]);

void addmanager(string managername, string managercnic);

void medicineindemand();

int convertDatetodays(string date);

bool isExpired(string currentDate, string expiryDate);

void removemedicine(string updatemedicinebatchnumber, int &addmedicineindex, string medicinebatchnumber[], string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string filenameaddmedicine);

void addemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[]);

void removeemploy(int &addemployindex, string removeemploycnic, string employcnic[], string employname[], string employphonenumber[], string employgender[]);

void viewemploy(int &addemployindex, string employname[], string employcnic[], string employphonenumber[], string employgender[]);

string equipmentmenu();

void updateequipment(string updateequipmentbatchnumber, int &addequipmentindex, string equipmentbatchnumber[], string equipmentname[], string equipmentquantity[], string equipmentprize[], string filenameaddequipment);

void addequipmentform(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment);

void removeequipmentform(int &addequipmentindex, string updateequipmentbatchnumber, string equipmentbatchnumber[], string equipmentname[], string equipmentprize[], string equipmentquantity[], string filenameaddequipment);

string branchmenu();

void viewequipmentform(int &addequipmentindex, string equipmentname[], string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[]);

int count=0;

int x = 8;

int y = 10;

int x1 = 3;

int y\_1 = 3;

void signup(int &size, int &admin, string firstname[], string secondname[], string gender[], string city[], string nationality[], string contactnumber[], string cnic[], string username[], string userpassword[], string userrole[], string filename);

void signin(int size, string names, string passwords, string contactnumber[], string userrole[], string username[], string userpassword[], string &rolereturn, string cnic[], string nationality[], string city[], string gender[], string firstname[], string secondname[]);

void changemanager(string managername);

void hideblinking();

void filehandlingsignup(int &size, string firstname[], string secondname[], string gender[], string city[], string nationality[], string contactnumber[], string cnic[], string username[], string userpassword[], string userrole[], string filename);

void filehandlingaddpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient);

void filehandlingprofitloss(string patientcharges, int addpatientindex, string filenameprofitloss);

void filehandlingchangingsalary(string doctorsalary, string managersalary, string filenamechangingsalary);

void filehandlingaddbed(string bedadd, string filenameaddbed);

void filehandlingadddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor);

void filehandlingaddequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameaddequipmentmanager);

void filehandlingselectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[]);

void filehandlingbillinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[]);

void filehandlingpatientreview(string review[], int &addpatientindex);

void filehandlingbuymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines);

void filehandlingaddmedicine(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine);

void filehandlingaddemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[]);

void filehandlingaddequipments(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment);

string getField(string record, int field);

void view();

void readsignup(int &size, int &admin, string firstname[], string secondname[], string gender[], string city[], string nationality[], string contactnumber[], string cnic[], string username[], string userpassword[], string userrole[], string filename);

void readaddpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient);

void readprofitloss(string patientcharges, int addpatientindex, string filenameprofitloss);

void readchangingsalary(string doctorsalary, string managersalary, string filenamechangingsalary);

void readaddbed(string bedadd, string filenameaddbed);

// void readadddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex);

void readadddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor);

void readaddequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameequipmentmanager);

void readselectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[]);

void readbilinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[]);

void readpatientreview(string review[], int &addpatientindex);

void readbuymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines);

void readaddmedicine(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine);

void readaddemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[]);

void readaddequipments(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment);

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

//functions prototype ends

//main starts

main()

{

hideCursor();

hideblinking();

clear();

string filename;

int adminmultan=0;

int adminislamabad=0;

int adminlahore=0;

//variables declaration

int addpatientindex=0;

int viewpatientindex=0;

string username[100];

string userpassword[100];

string userrole[100];

string firstname[100];

string secondname[100];

string gender[100];

string city[100];

string nationality[100];

string contactnumber[100];

string cnic[100];

string names;

string passwords;

string adminoption;

string managername;

string option;

string patientname[100];

string patientage[100];

string patientcnic[100];

string patientbloodgroup[100];

string patienthistory[100];

string visitdatepatient[100];

string patientphonenumber[100];

string rolereturn;

string dischargepatientcnic;

string patientcharges;

string doctorsalary;

string managersalary;

string bedadd;

string doctorname[100];

string doctorspeciality[100];

string equipmentmanagername;

string buymedicinequantity="0";

string purchasemedicines;

string equipmentmanagergender;

string equipmentmanagercnic;

string selectdoctorname;

string selectdoctorspeciality;

string updatepatientcnic;

string updatemedicinebatchnumber;

string servicetype[100];

string roomtype[100];

string daysstayes[100];

string review[100];

string medicinename[100];

string medicinequantity[100];

string medicineexpirydate[100];

string medicineprice[100];

string medicinebatchnumber[100];

string customername[100];

string currentdate[100];

string purchasemedicinename;

string purchasemedicinequantity[100];

string employname[100];

string employcnic[100];

string employphonenumber[100];

string employgender[100];

string updateequipmentbatchnumber;

string removeemploycnic;

int purchasemedicineindex=0;

int addmedicineindex=0;

int adddoctorindex=0;

int addbillingindex;

int managercheck=0;

int addequipmentindex=0;

string equipmentname[100];

string equipmentquantity[100];

string equipmentprice[100];

string equipmentbatchnumber[100];

int addemployindex=0;

string hire[100];

int size = 0;

int addpatientindexislamabad=0;

int viewpatientindexislamabad=0;

string usernameislamabad[100];

string userpasswordislamabad[100];

string userroleislamabad[100];

string firstnameislamabad[100];

string secondnameislamabad[100];

string genderislamabad[100];

string cityislamabad[100];

string nationalityislamabad[100];

string contactnumberislamabad[100];

string cnicislamabad[100];

string namesislamabad;

string passwordsislamabad;

string adminoptionislamabad;

string managernameislamabad;

string optionislamabad;

string patientnameislamabad[100];

string patientageislamabad[100];

string patientcnicislamabad[100];

string patientbloodgroupislamabad[100];

string patienthistoryislamabad[100];

string visitdatepatientislamabad[100];

string patientphonenumberislamabad[100];

string dischargepatientcnicislamabad;

string patientchargesislamabad;

string doctorsalaryislamabad;

string managersalaryislamabad;

string bedaddislamabad;

string doctornameislamabad[100];

string doctorspecialityislamabad[100];

string equipmentmanagernameislamabad;

string buymedicinequantityislamabad="0";

string purchasemedicinesislamabad;

string equipmentmanagergenderislamabad;

string equipmentmanagercnicislamabad;

string selectdoctornameislamabad;

string selectdoctorspecialityislamabad;

string updatepatientcnicislamabad;

string updatemedicinebatchnumberislamabad;

string servicetypeislamabad[100];

string roomtypeislamabad[100];

string daysstayesislamabad[100];

string reviewislamabad[100];

string medicinenameislamabad[100];

string medicinequantityislamabad[100];

string medicineexpirydateislamabad[100];

string medicinepriceislamabad[100];

string medicinebatchnumberislamabad[100];

string customernameislamabad[100];

string currentdateislamabad[100];

string purchasemedicinenameislamabad;

string purchasemedicinequantityislamabad[100];

string employnameislamabad[100];

string employcnicislamabad[100];

string employphonenumberislamabad[100];

string employgenderislamabad[100];

string updateequipmentbatchnumberislamabad;

string removeemploycnicislamabad;

int purchasemedicineindexislamabad=0;

int addmedicineindexislamabad=0;

int adddoctorindexislamabad=0;

int addbillingindexislamabad;

int managercheckislamabad=0;

int addequipmentindexislamabad=0;

string equipmentnameislamabad[100];

string equipmentquantityislamabad[100];

string equipmentpriceislamabad[100];

string equipmentbatchnumberislamabad[100];

int addemployindexislamabad=0;

string hireislamabad[100];

int sizeislamabad = 0;

int addpatientindexlahore=0;

int viewpatientindexlahore=0;

string usernamelahore[100];

string userpasswordlahore[100];

string userrolelahore[100];

string firstnamelahore[100];

string secondnamelahore[100];

string genderlahore[100];

string citylahore[100];

string nationalitylahore[100];

string contactnumberlahore[100];

string cniclahore[100];

string nameslahore;

string passwordslahore;

string adminoptionlahore;

string managernamelahore;

string optionlahore;

string patientnamelahore[100];

string patientagelahore[100];

string patientcniclahore[100];

string patientbloodgrouplahore[100];

string patienthistorylahore[100];

string visitdatepatientlahore[100];

string patientphonenumberlahore[100];

string dischargepatientcniclahore;

string patientchargeslahore;

string doctorsalarylahore;

string managersalarylahore;

string bedaddlahore;

string doctornamelahore[100];

string doctorspecialitylahore[100];

string equipmentmanagernamelahore;

string buymedicinequantitylahore="0";

string purchasemedicineslahore;

string equipmentmanagergenderlahore;

string equipmentmanagercniclahore;

string selectdoctornamelahore;

string selectdoctorspecialitylahore;

string updatepatientcniclahore;

string updatemedicinebatchnumberlahore;

string servicetypelahore[100];

string roomtypelahore[100];

string daysstayeslahore[100];

string reviewlahore[100];

string medicinenamelahore[100];

string medicinequantitylahore[100];

string medicineexpirydatelahore[100];

string medicinepricelahore[100];

string medicinebatchnumberlahore[100];

string customernamelahore[100];

string currentdatelahore[100];

string purchasemedicinenamelahore;

string purchasemedicinequantitylahore[100];

string employnamelahore[100];

string employcniclahore[100];

string employphonenumberlahore[100];

string employgenderlahore[100];

string updateequipmentbatchnumberlahore;

string removeemploycniclahore;

int purchasemedicineindexlahore=0;

int addmedicineindexlahore=0;

int adddoctorindexlahore=0;

int addbillingindexlahore;

int managerchecklahore=0;

int addequipmentindexlahore=0;

string equipmentnamelahore[100];

string equipmentquantitylahore[100];

string equipmentpricelahore[100];

string equipmentbatchnumberlahore[100];

int addemployindexlahore=0;

string hirelahore[100];

int sizelahore = 0;

int addpatientindexmultan=0;

int viewpatientindexmultan=0;

string usernamemultan[100];

string userpasswordmultan[100];

string userrolemultan[100];

string firstnamemultan[100];

string secondnamemultan[100];

string gendermultan[100];

string citymultan[100];

string nationalitymultan[100];

string contactnumbermultan[100];

string cnicmultan[100];

string namesmultan;

string passwordsmultan;

string adminoptionmultan;

string managernamemultan;

string optionmultan;

string patientnamemultan[100];

string patientagemultan[100];

string patientcnicmultan[100];

string patientbloodgroupmultan[100];

string patienthistorymultan[100];

string visitdatepatientmultan[100];

string patientphonenumbermultan[100];

string dischargepatientcnicmultan;

string patientchargesmultan;

string doctorsalarymultan;

string managersalarymultan;

string bedaddmultan;

string doctornamemultan[100];

string doctorspecialitymultan[100];

string equipmentmanagernamemultan;

string buymedicinequantitymultan="0";

string purchasemedicinesmultan;

string equipmentmanagergendermultan;

string equipmentmanagercnicmultan;

string selectdoctornamemultan;

string selectdoctorspecialitymultan;

string updatepatientcnicmultan;

string updatemedicinebatchnumbermultan;

string servicetypemultan[100];

string roomtypemultan[100];

string daysstayesmultan[100];

string reviewmultan[100];

string medicinenamemultan[100];

string medicinequantitymultan[100];

string medicineexpirydatemultan[100];

string medicinepricemultan[100];

string medicinebatchnumbermultan[100];

string customernamemultan[100];

string currentdatemultan[100];

string purchasemedicinenamemultan;

string purchasemedicinequantitymultan[100];

string employnamemultan[100];

string employcnicmultan[100];

string employphonenumbermultan[100];

string employgendermultan[100];

string updateequipmentbatchnumbermultan;

string removeemploycnicmultan;

int purchasemedicineindexmultan=0;

int addmedicineindexmultan=0;

int adddoctorindexmultan=0;

int addbillingindexmultan;

int managercheckmultan=0;

int addequipmentindexmultan=0;

string equipmentnamemultan[100];

string equipmentquantitymultan[100];

string equipmentpricemultan[100];

string equipmentbatchnumbermultan[100];

int addemployindexmultan=0;

string hiremultan[100];

int sizemultan = 0;

readsignup(sizemultan, adminmultan, firstnamemultan, secondnamemultan, gendermultan, citymultan, nationalitymultan, contactnumbermultan, cnicmultan, usernamemultan, userpasswordmultan, userrolemultan, "multansignup.txt");

readsignup(sizeislamabad, adminislamabad, firstnameislamabad, secondnameislamabad, genderislamabad, cityislamabad, nationalityislamabad, contactnumberislamabad, cnicislamabad, usernameislamabad, userpasswordislamabad, userroleislamabad, "islamabadsignup.txt");

readsignup(sizelahore, adminlahore, firstnamelahore, secondnamelahore, genderlahore, citylahore, nationalitylahore, contactnumberlahore, cniclahore, usernamelahore, userpasswordlahore, userrolelahore, "lahoresignup.txt");

readaddpatient(addpatientindexmultan, patientnamemultan, patientagemultan, patientcnicmultan, patientbloodgroupmultan, patienthistorymultan, visitdatepatientmultan, patientphonenumbermultan, "multanaddpatient.txt");

readaddpatient(addpatientindexislamabad, patientnameislamabad, patientageislamabad, patientcnicislamabad, patientbloodgroupislamabad, patienthistoryislamabad, visitdatepatientislamabad, patientphonenumberislamabad, "islamabadaddpatient.txt");

readaddpatient(addpatientindexlahore, patientnamelahore, patientagelahore, patientcniclahore, patientbloodgrouplahore, patienthistorylahore, visitdatepatientlahore, patientphonenumberlahore, "lahoreaddpatient.txt");

readprofitloss(patientchargesmultan, addpatientindexmultan, "multanprofitloss.txt");

readprofitloss(patientchargeslahore, addpatientindexlahore, "lahoreprofitloss.txt");

readprofitloss(patientchargesislamabad, addpatientindexislamabad, "islamabadprofitloss.txt");

readchangingsalary(doctorsalaryislamabad, managersalaryislamabad, "islamabadchangingsalary.txt");

readchangingsalary(doctorsalarylahore, managersalarylahore, "lahorechangingsalary.txt");

readchangingsalary(doctorsalarymultan, managersalarymultan, "multanchangingsalary.txt");

readaddbed(bedaddlahore, "lahorebedadd.txt");

readaddbed(bedaddislamabad, "islamabadaddbed.txt");

readaddbed(bedaddmultan, "multanaddbed.txt");

readadddoctor(doctornamelahore, doctorspecialitylahore, adddoctorindexlahore, "lahoreadddoctor.txt");

readadddoctor(doctornamemultan, doctorspecialitymultan, adddoctorindexmultan, "multanadddoctor.txt");

readadddoctor(doctornameislamabad, doctorspecialityislamabad, adddoctorindexislamabad, "islamabadadddoctor.txt");

readaddequipmentmanager(equipmentmanagernameislamabad, equipmentmanagergenderislamabad, equipmentmanagercnicislamabad, managercheckislamabad, "islamabadaddequipmentmanager.txt");

readaddequipmentmanager(equipmentmanagernamelahore, equipmentmanagergenderlahore, equipmentmanagercniclahore, managerchecklahore, "lahoreaddequipmentmanager.txt");

readaddequipmentmanager(equipmentmanagernamemultan, equipmentmanagergendermultan, equipmentmanagercnicmultan, managercheckmultan, "multanaddequipmentmanager.txt");

readselectdoctor(selectdoctorname, selectdoctorspeciality, adddoctorindex, doctorname, doctorspeciality, hire);

readbilinginvoice(servicetype, addbillingindex, daysstayes, roomtype);

readpatientreview(review, addpatientindex);

readbuymedicines(buymedicinequantitymultan, purchasemedicinesmultan, addmedicineindexmultan, medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, medicinebatchnumbermultan, "multanbuymedicines.txt");

readbuymedicines(buymedicinequantityislamabad, purchasemedicinesislamabad, addmedicineindexislamabad, medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, medicinebatchnumberislamabad, "islamabadbuymedicines.txt");

readbuymedicines(buymedicinequantitylahore, purchasemedicineslahore, addmedicineindexlahore, medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, medicinebatchnumberlahore, "lahorebuymedicines.txt");

readaddmedicine(medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, medicinebatchnumbermultan, addmedicineindexmultan, "multanaddmedicine.txt");

readaddmedicine(medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, medicinebatchnumberlahore, addmedicineindexlahore, "lahoreaddmedicine.txt");

readaddmedicine(medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, medicinebatchnumberislamabad, addmedicineindexislamabad, "islamabadaddmedicine.txt");

readaddemploy(employname, addemployindex, employcnic, employphonenumber, employgender);

readaddequipments(equipmentnameislamabad, addequipmentindexislamabad, equipmentquantityislamabad, equipmentpriceislamabad, equipmentbatchnumberislamabad, "islamabadaddequipments.txt");

readaddequipments(equipmentnamelahore, addequipmentindexlahore, equipmentquantitylahore, equipmentpricelahore, equipmentbatchnumberlahore, "lahoreaddequipments.txt");

readaddequipments(equipmentnamemultan, addequipmentindexmultan, equipmentquantitymultan, equipmentpricemultan, equipmentbatchnumbermultan, "multanaddequipments.txt");

system("cls");

topheading();

// I call the heading function to start the system

system("cls");

//view();

//getch();

gotoxy(x1+5, y\_1);

// I call the this as loading function is in this function

while (true)

{

system("cls");

string branch=branchmenu();

//Here i check if the user enter no number then I show him this line

if(branch=="1")

{

system("cls");

islamabadinterface();

getch();

while (true)

{

header();

string chose = login();

//Here i check if the user enter no number then I show him this line

if(checkenter(chose)==0)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

//If enter 1 then signin function open

else if (chose == "1")

{

signin(sizeislamabad, namesislamabad, passwordsislamabad, contactnumberislamabad, userroleislamabad, usernameislamabad, userpasswordislamabad, rolereturn, cnicislamabad, nationalityislamabad, cityislamabad, genderislamabad, firstnameislamabad, secondnameislamabad);

//if the role is admin then admin menu open

if(rolereturn=="admin")

{

while(true)

{

string admin\_1=adminmenu();

//add patient menu open

if(admin\_1=="1")

{

addpatient(addpatientindexislamabad, patientnameislamabad, patientageislamabad, patientcnicislamabad, patientbloodgroupislamabad, patienthistoryislamabad, visitdatepatientislamabad, patientphonenumberislamabad, "islamabadaddpatient.txt");

filehandlingaddpatient(addpatientindexislamabad, patientnameislamabad, patientageislamabad, patientcnicislamabad, patientbloodgroupislamabad, patienthistoryislamabad, visitdatepatientislamabad, patientphonenumberislamabad, "islamabadaddpatient.txt");

}

//view patient menu open

else if(admin\_1=="2")

{

viewpatient(addpatientindexislamabad, patientnameislamabad, patientageislamabad, patientcnicislamabad, patientbloodgroupislamabad, patienthistoryislamabad, visitdatepatientislamabad, patientphonenumberislamabad);

getch();

}

//discharge patient occurs

else if(admin\_1=="3")

{

dischargepatient(dischargepatientcnicislamabad, addpatientindexislamabad, patientcnicislamabad, patientnameislamabad, patientageislamabad, patientbloodgroupislamabad, patienthistoryislamabad, patientphonenumberislamabad, visitdatepatientislamabad, "islamabadaddpatient.txt");

getch();

}

//to see profit

else if(admin\_1=="4")

{

profitloss(patientchargesislamabad, addpatientindexislamabad, "islamabadprofitloss.txt");

getch();

}

//to change the salary

else if(admin\_1=="5")

{

changingsalary(doctorsalaryislamabad, managersalaryislamabad, "islamabadchangingsalary.txt");

getch();

}

//to add bed in the hospital

else if(admin\_1=="6")

{

addbed(bedaddislamabad, "islamabadaddbed.txt");

getch();

}

//to add new doctors

else if(admin\_1=="7")

{

adddoctor(doctornameislamabad, doctorspecialityislamabad, adddoctorindexislamabad, "islamabadadddoctor.txt");

getch();

}

//to view doctors list

else if(admin\_1=="8")

{

viewdoctor(doctornameislamabad, doctorspecialityislamabad, hireislamabad, adddoctorindexislamabad);

getch();

}

//to add equipments for surgery

else if(admin\_1=="9")

{

addequipmentmanager(equipmentmanagernameislamabad, equipmentmanagergenderislamabad, equipmentmanagercnicislamabad, managercheckislamabad, "islamabadaddequipmentmanager.txt");

getch();

}

//to chenge\exchange the equipment

else if(admin\_1=="10")

{

changeequipmentmanager(managercheckislamabad, equipmentmanagernameislamabad, equipmentmanagergenderislamabad, equipmentmanagercnicislamabad, "islamabadaddequipmentmanager.txt");

getch();

}

//to update patient

else if(admin\_1=="11")

{

updatepatient(updatepatientcnicislamabad, patientcnicislamabad, addpatientindexislamabad, patientnameislamabad, patientageislamabad, patientbloodgroupislamabad, patienthistoryislamabad, visitdatepatientislamabad, patientphonenumberislamabad, "islamabadaddpatient.txt");

getch();

}

//if anyother then loop break and shifted to back page

else

{

break;

}

}

}

//if role is patient the patient menu opens

else if(rolereturn=="patient")

{

while(true)

{

string patient\_1=pateintmenu();

//to view doctors schedule

if(patient\_1=="1")

{

doctorschedule(doctornameislamabad, doctorspecialityislamabad, hireislamabad, adddoctorindexislamabad);

getch();

}

//to view doctors list

else if(patient\_1=="2")

{

viewdoctorslist(doctornameislamabad, doctorspecialityislamabad,hireislamabad, adddoctorindexislamabad);

getch();

}

//to view available beds

else if(patient\_1=="3")

{

availablebeds(bedaddislamabad, addpatientindexislamabad);

getch();

}

//to select doctors for surgery

else if(patient\_1=="4")

{

selectdoctor(selectdoctornameislamabad, selectdoctorspecialityislamabad, adddoctorindexislamabad, doctornameislamabad, doctorspecialityislamabad, hireislamabad);

getch();

}

//to pays bills

else if(patient\_1=="5")

{

billinginvoice(servicetypeislamabad, addbillingindexislamabad, daysstayesislamabad, roomtypeislamabad);

getch();

}

//to view prescription verified by google

else if(patient\_1=="6")

{

prescription();

}

//to give review

else if(patient\_1=="7")

{

patientreview(reviewislamabad, addpatientindexislamabad);

}

//to buy respected medicines

else if(patient\_1=="8")

{

buymedicines(buymedicinequantityislamabad, purchasemedicinesislamabad, addmedicineindexislamabad, medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, medicinebatchnumberislamabad, "islamabadbuymedicines.txt");

}

//to shift back

else

{

break;

}

}

}

//if the role is pharmacist then the pharmacist menu opens

else if(rolereturn=="pharmacist")

{

while(true)

{

string pharmacist\_1=pharmacymenu();

//add medicine form

if(pharmacist\_1=="1")

{

filehandlingaddmedicine(medicinename, medicinequantity, medicineexpirydate, medicineprice, medicinebatchnumber, addmedicineindex, "islamabadaddmedicine.txt");

addmedicineform(medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, medicinebatchnumberislamabad, addmedicineindexislamabad, "islamabadaddmedicine.txt");

}

//to view medicine form

else if(pharmacist\_1=="2")

{

viewmedicineform(buymedicinequantityislamabad, addmedicineindexislamabad, medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, medicinebatchnumberislamabad, "islamabadadaddmedicine.txt");

getch();

}

//to update medicine form

else if(pharmacist\_1=="3")

{

updatemedicineform(updatemedicinebatchnumberislamabad, medicinebatchnumberislamabad, addmedicineindexislamabad, medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, "islamabadaddmedicine.txt");

getch();

}

//to view expired medicines in a stock

else if(pharmacist\_1=="4")

{

expiredmedicine(addmedicineindexislamabad, currentdateislamabad, medicineexpirydateislamabad, medicinenameislamabad);

getch();

}

//to view the medicines which is mostly in demand

else if(pharmacist\_1=="5")

{

medicineindemand();

getch();

}

//to remove medicines

else if(pharmacist\_1=="6")

{

removemedicine(updatemedicinebatchnumberislamabad, addmedicineindexislamabad, medicinebatchnumberislamabad, medicinenameislamabad, medicinequantityislamabad, medicineexpirydateislamabad, medicinepriceislamabad, "islamabadadmedicine.txt");

getch();

}

//to add employ in the pharmacy

else if(pharmacist\_1=="7")

{

addemploy(employnameislamabad, addemployindexislamabad, employcnicislamabad, employphonenumberislamabad, employgenderislamabad);

getch();

}

//to remove employ from pharmacy

else if(pharmacist\_1=="8")

{

removeemploy(addemployindexislamabad, removeemploycnicislamabad, employcnicislamabad, employnameislamabad, employphonenumberislamabad, employgenderislamabad);

getch();

}

//to view the list of employs

else if(pharmacist\_1=="9")

{

viewemploy(addemployindexislamabad, employnameislamabad, employcnicislamabad, employphonenumberislamabad, employgenderislamabad);

getch();

}

//to shift back to the page

else

{

break;

}

}

}

//if the role is manager then the manager menu opens

else if(rolereturn=="equipments-manager")

{

while(true)

{

string equipmentmanager=equipmentmenu();

if(equipmentmanager=="1")

{

//open add equipment menu

addequipmentform(equipmentnameislamabad, addequipmentindexislamabad, equipmentquantityislamabad, equipmentpriceislamabad, equipmentbatchnumberislamabad, "islamabadaddequipments.txt");

}

else if(equipmentmanager=="2")

{

//show add equipments

viewequipmentform(addequipmentindexislamabad, equipmentnameislamabad, equipmentquantityislamabad, equipmentpriceislamabad, equipmentbatchnumberislamabad);

getch();

}

else if(equipmentmanager=="3")

{

//to remove equipment

removeequipmentform(addequipmentindexislamabad, updateequipmentbatchnumberislamabad, equipmentbatchnumberislamabad, equipmentnameislamabad, equipmentpriceislamabad, equipmentquantityislamabad, "islamabadaddequipments.txt");

}

else if(equipmentmanager=="4")

{

//to update equipment

updateequipment(updateequipmentbatchnumberislamabad, addequipmentindexislamabad, equipmentbatchnumberislamabad, equipmentnameislamabad, equipmentquantityislamabad, equipmentpriceislamabad, "islamabadaddequipments.txt");

}

else

{

break;

}

}

}

else if(rolereturn=="doctor")

{

while(true)

{

string doctoroption=doctormenu();

if(doctoroption=="1")

{

viewpatient(addpatientindexislamabad, patientnameislamabad, patientageislamabad, patientcnicislamabad, patientbloodgroupislamabad, patienthistoryislamabad, visitdatepatientislamabad, patientphonenumberislamabad);

}

else if(doctoroption=="2")

{

viewdoctorslist(doctornameislamabad, doctorspecialityislamabad, hireislamabad, adddoctorindexislamabad);

}

else if(doctoroption=="3")

{

availablebeds(bedaddislamabad, addpatientindexislamabad);

}

else

{

break;

}

}

}

}

// to opens the signup page

else if (chose == "2")

{

signup(sizeislamabad, adminislamabad, firstnameislamabad, secondnameislamabad, genderislamabad, cityislamabad, nationalityislamabad, contactnumberislamabad, cnicislamabad, usernameislamabad, userpasswordislamabad, userroleislamabad, "islamabadsignup.txt");

}

//to exit the programm

else if (chose == "3")

{

break;

}

//if user enter wrong input then we will take input again

else

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout << "Please engage in the progression by selecting a key to proceed further..." << endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

system("cls");

}

}

else if(branch=="2")

{

system("cls");

lahoreinterface();

getch();

while (true)

{

header();

string chose = login();

//Here i check if the user enter no number then I show him this line

if(checkenter(chose)==0)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

//If enter 1 then signin function open

else if (chose == "1")

{

signin(sizelahore, nameslahore, passwordslahore, contactnumberlahore, userrolelahore, usernamelahore, userpasswordlahore, rolereturn, cniclahore, nationalitylahore, citylahore, genderlahore, firstnamelahore, secondnamelahore); //if the role is admin then admin menu open

if(rolereturn=="admin")

{

while(true)

{

string admin\_1=adminmenu();

//add patient menu open

if(admin\_1=="1")

{

addpatient(addpatientindexlahore, patientnamelahore, patientagelahore, patientcniclahore, patientbloodgrouplahore, patienthistorylahore, visitdatepatientlahore, patientphonenumberlahore, "lahoreaddpatient.txt");

}

//view patient menu open

else if(admin\_1=="2")

{

viewpatient(addpatientindexlahore, patientnamelahore, patientagelahore, patientcniclahore, patientbloodgrouplahore, patienthistorylahore, visitdatepatientlahore, patientphonenumberlahore);

getch();

}

//discharge patient occurs

else if(admin\_1=="3")

{

dischargepatient(dischargepatientcniclahore, addpatientindexlahore, patientcniclahore, patientnamelahore, patientagelahore, patientbloodgrouplahore, patienthistorylahore, patientphonenumberlahore, visitdatepatientlahore, "lahoreaddpatient.txt");

getch();

}

//to see profit

else if(admin\_1=="4")

{

profitloss(patientchargeslahore, addpatientindexlahore, "lahoreprofitloss.txt");

getch();

}

//to change the salary

else if(admin\_1=="5")

{

changingsalary(doctorsalarylahore, managersalarylahore, "lahorecahngingsalary.txt");

getch();

}

//to add bed in the hospital

else if(admin\_1=="6")

{

addbed(bedaddlahore, "lahoreaddbed.txt");

getch();

}

//to add new doctors

else if(admin\_1=="7")

{

adddoctor(doctornamelahore, doctorspecialitylahore, adddoctorindexlahore, "lahoreadddoctor.txt");

getch();

}

//to view doctors list

else if(admin\_1=="8")

{

viewdoctor(doctornamelahore, doctorspecialitylahore, hirelahore, adddoctorindexlahore);

getch();

}

//to add equipments for surgery

else if(admin\_1=="9")

{

addequipmentmanager(equipmentmanagernamelahore, equipmentmanagergenderlahore, equipmentmanagercniclahore, managerchecklahore, "lahoreaddequipmentmanager.txt");

getch();

}

//to chenge\exchange the equipment

else if(admin\_1=="10")

{

changeequipmentmanager(managerchecklahore, equipmentmanagernamelahore, equipmentmanagergenderlahore, equipmentmanagercniclahore, "lahoreaddequipmentmanager.txt");

getch();

}

//to update patient

else if(admin\_1=="11")

{

updatepatient(updatepatientcniclahore, patientcniclahore, addpatientindexlahore, patientnamelahore, patientagelahore, patientbloodgrouplahore, patienthistorylahore, visitdatepatientlahore, patientphonenumberlahore, "lahoreaddpatient.txt");

getch();

}

//if anyother then loop break and shifted to back page

else

{

break;

}

}

}

//if role is patient the patient menu opens

else if(rolereturn=="patient")

{

while(true)

{

string patient\_1=pateintmenu();

//to view doctors schedule

if(patient\_1=="1")

{

doctorschedule(doctornamelahore, doctorspecialitylahore, hirelahore, adddoctorindexlahore);

getch();

}

//to view doctors list

else if(patient\_1=="2")

{

viewdoctorslist(doctornamelahore, doctorspecialitylahore,hirelahore, adddoctorindexlahore);

getch();

}

//to view available beds

else if(patient\_1=="3")

{

availablebeds(bedaddlahore, addpatientindexlahore);

getch();

}

//to select doctors for surgery

else if(patient\_1=="4")

{

selectdoctor(selectdoctornamelahore, selectdoctorspecialitylahore, adddoctorindexlahore, doctornamelahore, doctorspecialitylahore, hirelahore);

getch();

}

//to pays bills

else if(patient\_1=="5")

{

billinginvoice(servicetypelahore, addbillingindexlahore, daysstayeslahore, roomtypelahore);

getch();

}

//to view prescription verified by google

else if(patient\_1=="6")

{

prescription();

}

//to give review

else if(patient\_1=="7")

{

patientreview(reviewlahore, addpatientindexlahore);

}

//to buy respected medicines

else if(patient\_1=="8")

{

buymedicines(buymedicinequantitylahore, purchasemedicineslahore, addmedicineindexlahore, medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, medicinebatchnumberlahore, "lahorebuymedicines.txt");

}

//to shift back

else

{

break;

}

}

}

//if the role is pharmacist then the pharmacist menu opens

else if(rolereturn=="pharmacist")

{

while(true)

{

string pharmacist\_1=pharmacymenu();

//add medicine form

if(pharmacist\_1=="1")

{

addmedicineform(medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, medicinebatchnumberlahore, addmedicineindexlahore, "lahoreaddmedicine.txt");

getch();

}

//to view medicine form

else if(pharmacist\_1=="2")

{

viewmedicineform(buymedicinequantitylahore, addmedicineindexlahore, medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, medicinebatchnumberlahore, "lahoreaddmedicine.txt");

getch();

}

//to update medicine form

else if(pharmacist\_1=="3")

{

updatemedicineform(updatemedicinebatchnumberlahore, medicinebatchnumberlahore, addmedicineindexlahore, medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, "lahoreaddmedicine.txt");

getch();

}

//to view expired medicines in a stock

else if(pharmacist\_1=="4")

{

expiredmedicine(addmedicineindexlahore, currentdatelahore, medicineexpirydatelahore, medicinenamelahore);

getch();

}

//to view the medicines which is mostly in demand

else if(pharmacist\_1=="5")

{

medicineindemand();

getch();

}

//to remove medicines

else if(pharmacist\_1=="6")

{

removemedicine(updatemedicinebatchnumberlahore, addmedicineindexlahore, medicinebatchnumberlahore, medicinenamelahore, medicinequantitylahore, medicineexpirydatelahore, medicinepricelahore, "lahoreaddmedicine.txt");

getch();

}

//to add employ in the pharmacy

else if(pharmacist\_1=="7")

{

addemploy(employnamelahore, addemployindexlahore, employcniclahore, employphonenumberlahore, employgenderlahore);

getch();

}

//to remove employ from pharmacy

else if(pharmacist\_1=="8")

{

removeemploy(addemployindexlahore, removeemploycniclahore, employcniclahore, employnamelahore, employphonenumberlahore, employgenderlahore);

getch();

}

//to view the list of employs

else if(pharmacist\_1=="9")

{

viewemploy(addemployindexlahore, employnamelahore, employcniclahore, employphonenumberlahore, employgenderlahore);

getch();

}

//to shift back to the page

else

{

break;

}

}

}

//if the role is manager then the manager menu opens

else if(rolereturn=="equipments-manager")

{

while(true)

{

string equipmentmanager=equipmentmenu();

if(equipmentmanager=="1")

{

//open add equipment menu

addequipmentform(equipmentnamelahore, addequipmentindexlahore, equipmentquantitylahore, equipmentpricelahore, equipmentbatchnumberlahore, "lahoreaddequipments.txt");

}

else if(equipmentmanager=="2")

{

//show add equipments

viewequipmentform(addequipmentindexlahore, equipmentnamelahore, equipmentquantitylahore, equipmentpricelahore, equipmentbatchnumberlahore);

getch();

}

else if(equipmentmanager=="3")

{

//to remove equipment

removeequipmentform(addequipmentindexlahore, updateequipmentbatchnumberlahore, equipmentbatchnumberlahore, equipmentnamelahore, equipmentpricelahore, equipmentquantitylahore, "lahoreaddequipments.txt");

}

else if(equipmentmanager=="4")

{

//to update equipment

updateequipment(updateequipmentbatchnumberlahore, addequipmentindexlahore, equipmentbatchnumberlahore, equipmentnamelahore, equipmentquantitylahore, equipmentpricelahore, "lahoreaddequipments.txt");

}

else

{

break;

}

}

}

else if(rolereturn=="doctor")

{

while(true)

{

string doctoroption=doctormenu();

if(doctoroption=="1")

{

viewpatient(addpatientindexlahore, patientnamelahore, patientagelahore, patientcniclahore, patientbloodgrouplahore, patienthistorylahore, visitdatepatientlahore, patientphonenumberlahore);

}

else if(doctoroption=="2")

{

viewdoctorslist(doctornamelahore, doctorspecialitylahore, hirelahore, adddoctorindexlahore);

}

else if(doctoroption=="3")

{

availablebeds(bedaddlahore, addpatientindexlahore);

}

else

{

break;

}

}

}

}

// to opens the signup page

else if (chose == "2")

{

signup(sizelahore, adminlahore, firstnamelahore, secondnamelahore, genderlahore, citylahore, nationalitylahore, contactnumberlahore, cniclahore, usernamelahore, userpasswordlahore, userrolelahore, "lahoresignup.txt");

}

//to exit the programm

else if (chose == "3")

{

break;

}

//if user enter wrong input then we will take input again

else

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout << "Please engage in the progression by selecting a key to proceed further..." << endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

system("cls");

}

}

if(branch=="3")

{

system("cls");

multaninterface();

getch();

while (true)

{

header();

string chose = login();

//Here i check if the user enter no number then I show him this line

if(checkenter(chose)==0)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

//If enter 1 then signin function open

else if (chose == "1")

{

signin(sizemultan, namesmultan, passwordsmultan, contactnumbermultan, userrolemultan, usernamemultan, userpasswordmultan, rolereturn, cnicmultan, nationalitymultan, citymultan, gendermultan, firstnamemultan, secondnamemultan);

//if the role is admin then admin menu open

if(rolereturn=="admin")

{

while(true)

{

string admin\_1=adminmenu();

//add patient menu open

if(admin\_1=="1")

{

addpatient(addpatientindexmultan, patientnamemultan, patientagemultan, patientcnicmultan, patientbloodgroupmultan, patienthistorymultan, visitdatepatientmultan, patientphonenumbermultan, "multanaddpatient.txt");

}

//view patient menu open

else if(admin\_1=="2")

{

viewpatient(addpatientindexmultan, patientnamemultan, patientagemultan, patientcnicmultan, patientbloodgroupmultan, patienthistorymultan, visitdatepatientmultan, patientphonenumbermultan);

getch();

}

//discharge patient occurs

else if(admin\_1=="3")

{

dischargepatient(dischargepatientcnicmultan, addpatientindexmultan, patientcnicmultan, patientnamemultan, patientagemultan, patientbloodgroupmultan, patienthistorymultan, patientphonenumbermultan, visitdatepatientmultan, "multanaddpatient.txt");

getch();

}

//to see profit

else if(admin\_1=="4")

{

profitloss(patientchargesmultan, addpatientindexmultan, "multanprofitloss.txt");

getch();

}

//to change the salary

else if(admin\_1=="5")

{

changingsalary(doctorsalarymultan, managersalarymultan, "multanchangingsalary.txt");

getch();

}

//to add bed in the hospital

else if(admin\_1=="6")

{

addbed(bedaddmultan, "multanaddbed.txt");

getch();

}

//to add new doctors

else if(admin\_1=="7")

{

adddoctor(doctornamemultan, doctorspecialitymultan, adddoctorindexmultan, "multanadddoctor.txt");

getch();

}

//to view doctors list

else if(admin\_1=="8")

{

viewdoctor(doctornamemultan, doctorspecialitymultan, hiremultan, adddoctorindexmultan);

getch();

}

//to add equipments for surgery

else if(admin\_1=="9")

{

addequipmentmanager(equipmentmanagernamemultan, equipmentmanagergendermultan, equipmentmanagercnicmultan, managercheckmultan, "multanaddequipmentmanager.txt");

getch();

}

//to chenge\exchange the equipment

else if(admin\_1=="10")

{

changeequipmentmanager(managercheckmultan, equipmentmanagernamemultan, equipmentmanagergendermultan, equipmentmanagercnicmultan, "multanaddequipmentmanager.txt");

getch();

}

//to update patient

else if(admin\_1=="11")

{

updatepatient(updatepatientcnicmultan, patientcnicmultan, addpatientindexmultan, patientnamemultan, patientagemultan, patientbloodgroupmultan, patienthistorymultan, visitdatepatientmultan, patientphonenumbermultan, "multanaddpatient.txt");

getch();

}

//if anyother then loop break and shifted to back page

else

{

break;

}

}

}

//if role is patient the patient menu opens

else if(rolereturn=="patient")

{

while(true)

{

string patient\_1=pateintmenu();

//to view doctors schedule

if(patient\_1=="1")

{

doctorschedule(doctornamemultan, doctorspecialitymultan, hiremultan, adddoctorindexmultan);

getch();

}

//to view doctors list

else if(patient\_1=="2")

{

viewdoctorslist(doctornamemultan, doctorspecialitymultan,hiremultan, adddoctorindexmultan);

getch();

}

//to view available beds

else if(patient\_1=="3")

{

availablebeds(bedaddmultan, addpatientindexmultan);

getch();

}

//to select doctors for surgery

else if(patient\_1=="4")

{

selectdoctor(selectdoctornamemultan, selectdoctorspecialitymultan, adddoctorindexmultan, doctornamemultan, doctorspecialitymultan, hiremultan);

getch();

}

//to pays bills

else if(patient\_1=="5")

{

billinginvoice(servicetypemultan, addbillingindexmultan, daysstayesmultan, roomtypemultan);

getch();

}

//to view prescription verified by google

else if(patient\_1=="6")

{

prescription();

}

//to give review

else if(patient\_1=="7")

{

patientreview(reviewmultan, addpatientindexmultan);

}

//to buy respected medicines

else if(patient\_1=="8")

{

buymedicines(buymedicinequantitymultan, purchasemedicinesmultan, addmedicineindexmultan, medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, medicinebatchnumbermultan, "multanbuymedicines.txt");

}

//to shift back

else

{

break;

}

}

}

//if the role is pharmacist then the pharmacist menu opens

else if(rolereturn=="pharmacist")

{

while(true)

{

string pharmacist\_1=pharmacymenu();

//add medicine form

if(pharmacist\_1=="1")

{

addmedicineform(medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, medicinebatchnumbermultan, addmedicineindexmultan, "multanaddmedicine.txt");

getch();

}

//to view medicine form

else if(pharmacist\_1=="2")

{

viewmedicineform(buymedicinequantitymultan, addmedicineindexmultan, medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, medicinebatchnumbermultan, "multanaddmedicine.txt");

getch();

}

//to update medicine form

else if(pharmacist\_1=="3")

{

updatemedicineform(updatemedicinebatchnumbermultan, medicinebatchnumbermultan, addmedicineindexmultan, medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, "multanaddmedicine.txt");

getch();

}

//to view expired medicines in a stock

else if(pharmacist\_1=="4")

{

expiredmedicine(addmedicineindexmultan, currentdatemultan, medicineexpirydatemultan, medicinenamemultan);

getch();

}

//to view the medicines which is mostly in demand

else if(pharmacist\_1=="5")

{

medicineindemand();

getch();

}

//to remove medicines

else if(pharmacist\_1=="6")

{

removemedicine(updatemedicinebatchnumbermultan, addmedicineindexmultan, medicinebatchnumbermultan, medicinenamemultan, medicinequantitymultan, medicineexpirydatemultan, medicinepricemultan, "multanaddmedicine.txt");

getch();

}

//to add employ in the pharmacy

else if(pharmacist\_1=="7")

{

addemploy(employnamemultan, addemployindexmultan, employcnicmultan, employphonenumbermultan, employgendermultan);

getch();

}

//to remove employ from pharmacy

else if(pharmacist\_1=="8")

{

removeemploy(addemployindexmultan, removeemploycnicmultan, employcnicmultan, employnamemultan, employphonenumbermultan, employgendermultan);

getch();

}

//to view the list of employs

else if(pharmacist\_1=="9")

{

viewemploy(addemployindexmultan, employnamemultan, employcnicmultan, employphonenumbermultan, employgendermultan);

getch();

}

//to shift back to the page

else

{

break;

}

}

}

//if the role is manager then the manager menu opens

else if(rolereturn=="equipments-manager")

{

while(true)

{

string equipmentmanager=equipmentmenu();

if(equipmentmanager=="1")

{

//open add equipment menu

addequipmentform(equipmentnamemultan, addequipmentindexmultan, equipmentquantitymultan, equipmentpricemultan, equipmentbatchnumbermultan, "multanaddequipments.txt");

}

else if(equipmentmanager=="2")

{

//show add equipments

viewequipmentform(addequipmentindexmultan, equipmentnamemultan, equipmentquantitymultan, equipmentpricemultan, equipmentbatchnumbermultan);

getch();

}

else if(equipmentmanager=="3")

{

//to remove equipment

removeequipmentform(addequipmentindexmultan, updateequipmentbatchnumbermultan, equipmentbatchnumbermultan, equipmentnamemultan, equipmentpricemultan, equipmentquantitymultan, "multanaddequipments.txt");

}

else if(equipmentmanager=="4")

{

//to update equipment

updateequipment(updateequipmentbatchnumbermultan, addequipmentindexmultan, equipmentbatchnumbermultan, equipmentnamemultan, equipmentquantitymultan, equipmentpricemultan, "multanaddequipments.txt");

}

else

{

break;

}

}

}

else if(rolereturn=="doctor")

{

while(true)

{

string doctoroption=doctormenu();

if(doctoroption=="1")

{

viewpatient(addpatientindexmultan, patientnamemultan, patientagemultan, patientcnicmultan, patientbloodgroupmultan, patienthistorymultan, visitdatepatientmultan, patientphonenumbermultan);

}

else if(doctoroption=="2")

{

viewdoctorslist(doctornamemultan, doctorspecialitymultan, hiremultan, adddoctorindexmultan);

}

else if(doctoroption=="3")

{

availablebeds(bedaddmultan, addpatientindexmultan);

}

else

{

break;

}

}

}

}

// to opens the signup page

else if (chose == "2")

{

signup(sizemultan, adminmultan, firstnamemultan, secondnamemultan, gendermultan, citymultan, nationalitymultan, contactnumbermultan, cnicmultan, usernamemultan, userpasswordmultan, userrolemultan, "multansignup.txt");

}

//to exit the programm

else if (chose == "3")

{

break;

}

//if user enter wrong input then we will take input again

else

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout << "Please engage in the progression by selecting a key to proceed further..." << endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

system("cls");

}

}

else if(branch=="4")

{

thankyou();

return 0;

}

else

{

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again. "<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

}

}

void hideblinking()

{

COORD cursorPos;

cursorPos.X = 0;

cursorPos.Y = 0;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), cursorPos);

return;

}

int check\_integer(string num)

{

//the function is used to check integer in the input from the user

int check = 0;

for (int i = 0; num[i] != '\0'; i++)

{

if ((num[i] != '0') || (num[i] != '1') || (num[i] != '2') || (num[i] != '3') || (num[i] != '4') || (num[i] != '5') || (num[i] != '6') || (num[i] != '7') || (num[i] != '8') || (num[i] != '9'))

{

check++;

}

}

return check;

}

string branchmenu()

{

//this is the branch menu which we decide to go to a branch

string branchname;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"In which branch you want to go."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout << "1. Green City Branch." << endl;

gotoxy(x, y + 2);

SetConsoleTextAttribute(h, 1);

cout << "2. Punjab Pride Branch." << endl;

gotoxy(x, y + 3);

SetConsoleTextAttribute(h, 1);

cout << "3. Shrine City Branch." << endl;

gotoxy(x, y + 4);

SetConsoleTextAttribute(h, 8);

cout << "4. EXIT."<<endl;

gotoxy(x, y + 5);

SetConsoleTextAttribute(h, 1);

cout << "Which distinguished branch are you contemplating visiting...";

cin >> branchname;

return branchname;

}

void hideCursor()

{

HANDLE consoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cursorInfo;

GetConsoleCursorInfo(consoleHandle, &cursorInfo);

cursorInfo.bVisible = false;

SetConsoleCursorInfo(consoleHandle, &cursorInfo);

}

void clear()

{

COORD cursorPos;

cursorPos.X = 0;

cursorPos.Y = 0;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), cursorPos);

}

void topheading()

{

cout<<" ";

SetConsoleTextAttribute(h, 255);

cout << " .+@@#######@@\*. ";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 255);

cout << " .-%@##########\*@@-. ";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 255);

cout << ".:@###############@-.";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 255);

cout << ".=@######";

SetConsoleTextAttribute(h, 68);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "#####%@+.";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 255);

cout << ".=#####";

SetConsoleTextAttribute(h, 68);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "#####+.";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 255);

cout << ".-@%#####";

SetConsoleTextAttribute(h, 68);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "######@+.";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 170);

cout << "............";

SetConsoleTextAttribute(h, 255);

cout << ".:\*@#############@#:.";

SetConsoleTextAttribute(h, 170);

cout << "............";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 170);

cout << ".@%%%%%%%%%%";

SetConsoleTextAttribute(h, 255);

cout << "%%%%@%##########@%%%%";

SetConsoleTextAttribute(h, 170);

cout << "%%%%%%%%%@@:";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 170);

cout << ".@\*+++++++++";

SetConsoleTextAttribute(h, 255);

cout << "++++\*%@@%###%@@%\*++++";

SetConsoleTextAttribute(h, 170);

cout << "+++++++++%@:";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 170);

cout << ".@%#################%%%%%#################@@:";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 170);

cout << ".@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@:";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 17);

cout << ".@:";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 17);

cout << "#@:";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 17);

cout << ".@:";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@@@@@@@@@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@@@@@@@@@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@@@@@@@@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 17);

cout << "#@:";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 17);

cout << ".@:";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << " @";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "..";

SetConsoleTextAttribute(h, 255);

cout << ":+#########+:.. ";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 17);

cout << ".@:";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << " @";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "@#";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "%=";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 8);

cout << ".-#%%%%%%%%%@@-.";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << " @";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "+";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "%=";

SetConsoleTextAttribute(h, 8);

cout << " "<<endl;

cout<<" .#";

SetConsoleTextAttribute(h, 8);

cout << "@+:............:\*";

SetConsoleTextAttribute(h, 8);

cout << "@@ ";

SetConsoleTextAttribute(h, 119);

cout << " @";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ":@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "@:";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << " + ";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "+%"<<endl;

SetConsoleTextAttribute(h, 8);

cout<<" .+";

SetConsoleTextAttribute(h, 8);

cout << "%-.================:";

SetConsoleTextAttribute(h, 8);

cout << "@@";

SetConsoleTextAttribute(h, 119);

cout << "@@@@@@@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 119);

cout << "@@@@@@@";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 119);

cout << "@@@@@@";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "@:";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << " ";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 255);

cout << "=%";

SetConsoleTextAttribute(h, 8);

cout << " "<<endl;

cout<<" .+\*:=====================\*= ";

SetConsoleTextAttribute(h, 14);

cout <<" -@+%%@\*==\*#@@@@%\*+=+#%%%.";

SetConsoleTextAttribute(h, 8);

cout << " "<<endl;

cout<<" .-%=======================+#- ";

SetConsoleTextAttribute(h, 14);

cout <<".@%%%%%%%%%%%%%%%%%%%%%%%%@+";

SetConsoleTextAttribute(h, 8);

cout << " "<<endl;

cout<<" .+#=========++++++=========#";

SetConsoleTextAttribute(h, 119);

cout << "@@@@#";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 119);

cout << "@@@@@@@";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 14);

cout << " :#@@@%%%%%%%%%%%%%%%%%%%%%%%%%#: "<<endl;

SetConsoleTextAttribute(h, 8);

cout<<" .+#=+\*#%@@%#\*\*++\*\*#%@@%#\*++# ";

SetConsoleTextAttribute(h, 119);

cout << " #";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 14);

cout << " :##%@%%%%%%%%%@@@@@@%%%%%%%%%%#: "<<endl;

SetConsoleTextAttribute(h, 8);

cout<<" .+@@#+-";

SetConsoleTextAttribute(h, 15);

cout << "::::::::::::::::";

SetConsoleTextAttribute(h, 8);

cout << "-+@@@ ";

SetConsoleTextAttribute(h, 119);

cout << " #";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 14);

cout << " :#\*%@%%%%%%@@@%=::=#@@@@%%%%%%%- "<<endl;

SetConsoleTextAttribute(h, 8);

cout<<" .\*@%\*-";

SetConsoleTextAttribute(h, 15);

cout << ":::-=-::::::-=-:::";

SetConsoleTextAttribute(h, 8);

cout << "-%%@ ";

SetConsoleTextAttribute(h, 119);

cout << " #";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << "@.";

SetConsoleTextAttribute(h, 14);

cout << " :##%@@@@@@%\*-";

SetConsoleTextAttribute(h, 15);

cout << "::::::::";

SetConsoleTextAttribute(h, 14);

cout << "-\*%@@@@@@%-. "<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" .%+=+-::-";

SetConsoleTextAttribute(h, 1);

cout << "@";

SetConsoleTextAttribute(h, 15);

cout << ":--::::--";

SetConsoleTextAttribute(h, 1);

cout << "@";

SetConsoleTextAttribute(h, 15);

cout << "-::-\*:\*@";

SetConsoleTextAttribute(h, 119);

cout << " %";

SetConsoleTextAttribute(h, 119);

cout << "%%";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout << ".@@@@@@@@@.";

SetConsoleTextAttribute(h, 14);

cout << " :\*@##%%=";

SetConsoleTextAttribute(h, 15);

cout << "-::#@+=";

SetConsoleTextAttribute(h, 15);

cout << "::::-+#%=:-";

SetConsoleTextAttribute(h, 14);

cout << "=\*@#\*#: "<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" .-#%\*-:::+##=::::-\*#\*-::-%%#:. ";

SetConsoleTextAttribute(h, 14);

cout << ".=%@@%\*";

SetConsoleTextAttribute(h, 15);

cout << "::::\*";

SetConsoleTextAttribute(h, 1);

cout << "@";

SetConsoleTextAttribute(h, 15);

cout << "\*:::::-:";

SetConsoleTextAttribute(h, 1);

cout << "@";

SetConsoleTextAttribute(h, 15);

cout << ":+:::-";

SetConsoleTextAttribute(h, 14);

cout <<"%\*@@#:."<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" -#-::::::::--::::::::=#: ";

SetConsoleTextAttribute(h, 119);

cout <<".@@@@@@@@@@@@@@.";

SetConsoleTextAttribute(h, 14);

cout << "=@%%%%%";

SetConsoleTextAttribute(h, 15);

cout << ":::-:-::::-:::-:-:::";

SetConsoleTextAttribute(h, 14);

cout << "=@%%%@#."<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" .\*#-:::::=";

SetConsoleTextAttribute(h, 4);

cout << "\*-\_-\*";

SetConsoleTextAttribute(h, 15);

cout << ":::-:-%= ";

SetConsoleTextAttribute(h, 119);

cout <<".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout <<"@.";

SetConsoleTextAttribute(h, 14);

cout << "=@@%%%%";

SetConsoleTextAttribute(h, 15);

cout << "+::-:::-====-:::-::-";

SetConsoleTextAttribute(h, 14);

cout << "%%%%%@@:"<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" .+%+-:::-++++-:::-\*@= ";

SetConsoleTextAttribute(h, 119);

cout <<".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout <<"@.";

SetConsoleTextAttribute(h, 14);

cout << ":#@@%%%%";

SetConsoleTextAttribute(h, 15);

cout << "\*-::::-";

SetConsoleTextAttribute(h, 4);

cout << "#%#%\*";

SetConsoleTextAttribute(h, 15);

cout << ":::::+";

SetConsoleTextAttribute(h, 14);

cout << "%%%%%@@=."<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" .:\*@#\*+======+\*%%+. ";

SetConsoleTextAttribute(h, 119);

cout <<".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout <<"@.";

SetConsoleTextAttribute(h, 14);

cout << " .=#@@@#\*%#";

SetConsoleTextAttribute(h, 15);

cout << "+=---------=\*%\*+";

SetConsoleTextAttribute(h, 14);

cout << "%@@%\*:. "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" .\*%##\*\*\*%%\*\*###%+ ";

SetConsoleTextAttribute(h, 119);

cout <<".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout <<"@.";

SetConsoleTextAttribute(h, 6);

cout << " .-+:.+";

SetConsoleTextAttribute(h, 13);

cout << "@#%####%####";

SetConsoleTextAttribute(h, 6);

cout << "@%+. "<<endl;

SetConsoleTextAttribute(h, 6);

cout<<" .+@+";

SetConsoleTextAttribute(h, 2);

cout << "+%+++====+\*\*#";

SetConsoleTextAttribute(h, 6);

cout << "+\*%\*###-. ";

SetConsoleTextAttribute(h, 119);

cout <<".@";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 119);

cout <<"@.";

SetConsoleTextAttribute(h, 6);

cout << " .\*\*##%#%";

SetConsoleTextAttribute(h, 13);

cout << "\*%+========+";

SetConsoleTextAttribute(h, 6);

cout << "####-. "<<endl;

cout<<" .\*+ ";

SetConsoleTextAttribute(h, 2);

cout << "+%==========\*";

SetConsoleTextAttribute(h, 6);

cout << "%+##\*##%- ";

SetConsoleTextAttribute(h, 119);

cout <<".@@@@@@@@@@@@@@.";

SetConsoleTextAttribute(h, 6);

cout << " +## ";

SetConsoleTextAttribute(h, 13);

cout << " %@==========";

SetConsoleTextAttribute(h, 6);

cout << "\*%=\*\*. "<<endl;

cout<<" .\*%%@@%#";

SetConsoleTextAttribute(h, 2);

cout << "\*\*\*\*\*\*\*\*\*\*";

SetConsoleTextAttribute(h, 17);

cout << "@%\*\*\*\*#%@@@@@@@@@@@@@@@@@@@@@@@@@\*\*\*#%";

SetConsoleTextAttribute(h, 13);

cout << "@@%+======+#%# "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 68);

cout << "##===+\*\*+===%";

SetConsoleTextAttribute(h, 51);

cout << "\*+@##%#\*\*\*\*###################\*\*\*#%%#+";

SetConsoleTextAttribute(h, 68);

cout << "+@=++\*\*\*\*\*++#=";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 68);

cout << "+%=\*";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "%++#";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "%+=#";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "%=+#";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 68);

cout << "-@\*%";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "\*###";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "#\*\*#";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "\*%+#";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 68);

cout << ".\*%%";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "%%%+";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "-=++";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 68);

cout << "++=-";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

gotoxy(x1, y\_1+36);

SetConsoleTextAttribute(h, 2);

cout<<"Made By: ";

gotoxy(x1+9, y\_1+36);

SetConsoleTextAttribute(h, 3);

cout<<"MUHAMMAD\_OWAIS\_KHUBAISI";

SetConsoleTextAttribute(h, 2);

showLoadingAnimation();

}

void gotoxy(int x, int y)

{

//this function is used for using the gotoxy in the system

COORD coordinates;

coordinates.X = x;

coordinates.Y = y;

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), coordinates);

}

void delay(int milliseconds)

{

//this is for the loading time delay

clock\_t start\_time = clock();

while ((clock() - start\_time) / CLOCKS\_PER\_SEC \* 100000 < milliseconds)

{

}

}

void showLoadingAnimation()

{

//this function is actually the loading function

const char animation[4] = {'|', '/', '-', '\\'};

gotoxy(x1+80, y\_1+36);

cout << "Loading ";

for (int i = 0; i < 10; ++i)

{

cout << animation[i % 4] << ' ';

delay(5);

cout << "\b\b";

}

gotoxy(x1+50, y\_1+20);

cout << "Complete!" << endl;

}

void header()

{

system("cls");

SetConsoleTextAttribute(h, 6);

//this is the main header of the application where inside this block all the things occurs and proceed

cout<<" ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\* ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*"<<endl;

SetConsoleTextAttribute(h, 6);

cout<<" .::-:::. .:::::.. .::-:::. .::::::. .::: ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "\_\_\_\_ \_ \_ \_\_\_ \_\_ \_\_ \_ \_\_\_\_ \_\_\_ ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "::. ..:::::. .:::-::. ..::-::. ..::-::. "<<endl;

cout<<".:-===--=::.:-==-===::.::=-=--=::.-:==-===:-.::=-= ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "( \_ \\( \\/ )/ \_) / \_\\ ( )( \_\_ )/ \_\_) ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "-=::.::===-==:-.::=--=-=::.::=-=-==-:.::=--===-:."<<endl;

cout<<"--====--==---===--===---====--==---===--===---==-= ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << ") \_\_/ ) / \\\_ \\/ \\/ ) / )( \\\_\_ \\ ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "-==---===--===---==--====---====-===-:-==---=-=--"<<endl;

cout<<"-====-.::::::::::::::::::::::::::::::::::::::::::: ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "(\_\_) (\_\_/ (\_\_\_/\\\_/\\\_/\\\_)\_\_) (\_\_) (\_\_\_/ ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << "::::::::::::::::::::::::::::::::::::::::::.-====-"<<endl;

cout<<".=-=-=.. ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*";

SetConsoleTextAttribute(h, 6);

cout << " Welcome to ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*\*\*";

SetConsoleTextAttribute(h, 6);

cout << " ..=-=-=."<<endl;

cout<<" .-:==-. ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\*";

SetConsoleTextAttribute(h, 6);

cout << " Medi-Harmony ";

SetConsoleTextAttribute(h, 11);

cout << "\*\*\* ";

SetConsoleTextAttribute(h, 6);

cout << ".-==:-. "<<endl;

cout<<" ::-:. ";

SetConsoleTextAttribute(h, 11);

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

SetConsoleTextAttribute(h, 6);

cout << " .:-:: "<<endl;

cout<<" .-----: :--=--. "<<endl;

cout<<".:==-=-: :-=-==:."<<endl;

cout<<".::-==-: :-==-::."<<endl;

cout<<"..:=---: :---=-.."<<endl;

cout<<" .:--:: ::--:. "<<endl;

cout<<" :--:: ::--:. "<<endl;

cout<<".:==-=-: :-=-==:."<<endl;

cout<<".::-==-: :-==-::."<<endl;

cout<<"..:=---: :---=-.."<<endl;

cout<<" .:--:: ::--:. "<<endl;

cout<<" :--:: ::--:. "<<endl;

cout<<".:==-=-: :-=-==:."<<endl;

cout<<".::-==-: :-==-::."<<endl;

cout<<"..:=---: :---=-.."<<endl;

cout<<" .:--:: ::--:. "<<endl;

cout<<" :--:: ::--:. "<<endl;

cout<<"..-=---: :---=-.."<<endl;

cout<<".:-==--: :--==-:."<<endl;

cout<<".:-==--: :--==-:."<<endl;

cout<<"..:=---: :---=-.."<<endl;

cout<<" :--:: ::--:. "<<endl;

cout<<" .:--:: ::--:. "<<endl;

cout<<"..:=---: :---=-.."<<endl;

cout<<".::-==-: :-==--:."<<endl;

cout<<".:==-=-: :-=-==:."<<endl;

cout<<" .-----: :--=--. "<<endl;

cout<<" ::--. .:-:: "<<endl;

cout<<" .-:==-. .-==:-. "<<endl;

cout<<".=-=-=:. ..=-=-=."<<endl;

cout<<"-====-.::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.-====-"<<endl;

cout<<"--====--==---===--===---====--==---===--===---==-==-==---===--===---==-==-==---===--===---==-==-==---===--===---==-==-==---===--===---==--====---===--===---==-=-=-=--"<<endl;

}

void multaninterface()

{

system("cls");

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<" + "<<endl;

cout<<" +++ "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << ":::::::";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << "--:::::::::::::--";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << "-...................-";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << ":.........................:";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << ":...............................:";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" -";

SetConsoleTextAttribute(h, 15);

cout << ":...................................:";

SetConsoleTextAttribute(h, 15);

cout << "- "<<endl;

cout<<" -";

SetConsoleTextAttribute(h, 15);

cout << ".......................................";

SetConsoleTextAttribute(h, 15);

cout << "- "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << ":.........................................:";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << ":............................................";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" -";

SetConsoleTextAttribute(h, 15);

cout << ":.............................................:";

SetConsoleTextAttribute(h, 15);

cout << "- "<<endl;

cout<<" =";

SetConsoleTextAttribute(h, 15);

cout << ":...............................................:";

SetConsoleTextAttribute(h, 15);

cout << "= "<<endl;

cout<<" \*\* -";

SetConsoleTextAttribute(h, 15);

cout << ":...............................................:";

SetConsoleTextAttribute(h, 15);

cout << "- \*\* "<<endl;

cout<<" + -";

SetConsoleTextAttribute(h, 15);

cout << "-:-::-::-::-::-::-::-::-::-::-::-:-::--::-::-::-::-";

SetConsoleTextAttribute(h, 15);

cout << "- + "<<endl;

SetConsoleTextAttribute(h, 10);

cout<<" =========================================================== "<<endl;

cout<<" +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ "<<endl;

cout<<" =================================================================== "<<endl;

SetConsoleTextAttribute(h, 6);

cout<<" +\* +--";

SetConsoleTextAttribute(h, 11);

cout << "=-==\*\*++++++\*";

SetConsoleTextAttribute(h, 6);

cout << "\*+--+--=+-+=";

SetConsoleTextAttribute(h, 11);

cout << "\*\*+++++++++++\*\*";

SetConsoleTextAttribute(h, 6);

cout << "=++--++--";

SetConsoleTextAttribute(h, 11);

cout << "+\*\*\*+++++\*\*==-=";

SetConsoleTextAttribute(h, 6);

cout << "--+ \*+ "<<endl;

cout<<" -- +--";

SetConsoleTextAttribute(h, 11);

cout << "=-==\* \*";

SetConsoleTextAttribute(h, 6);

cout << "+--:--=----=";

SetConsoleTextAttribute(h, 11);

cout << "\*\*+++++++++++\*\*";

SetConsoleTextAttribute(h, 6);

cout << "-----:--+";

SetConsoleTextAttribute(h, 11);

cout << "\* \*==-=";

SetConsoleTextAttribute(h, 6);

cout << "--+ -- "<<endl;

cout<<" =:..:= +--";

SetConsoleTextAttribute(h, 11);

cout << "=-==\* \*";

SetConsoleTextAttribute(h, 6);

cout << "+--:--=----=";

SetConsoleTextAttribute(h, 11);

cout << "\*\*++++ ++++\*\*";

SetConsoleTextAttribute(h, 6);

cout << "--=--:--+";

SetConsoleTextAttribute(h, 11);

cout << "\* \*==-=";

SetConsoleTextAttribute(h, 6);

cout << "--+ =:..:= "<<endl;

cout<<" -......:+--";

SetConsoleTextAttribute(h, 11);

cout << "=-==\* \*";

SetConsoleTextAttribute(h, 6);

cout << "+--:=-=----=";

SetConsoleTextAttribute(h, 11);

cout << "\*\*++++ ++++\*\*";

SetConsoleTextAttribute(h, 6);

cout << "--=-=:--+";

SetConsoleTextAttribute(h, 11);

cout << "\* \*==-=";

SetConsoleTextAttribute(h,6);

cout << "--+::.....- "<<endl;

cout<<" +=--------";

SetConsoleTextAttribute(h, 11);

cout << "===============";

SetConsoleTextAttribute(h, 6);

cout << "-------------";

SetConsoleTextAttribute(h, 11);

cout << "=+++++++++++++=";

SetConsoleTextAttribute(h, 6);

cout << "-----------";

SetConsoleTextAttribute(h, 11);

cout << "===============";

SetConsoleTextAttribute(h, 6);

cout << "--------=+ "<<endl;

cout<<" ==-------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "------------";

SetConsoleTextAttribute(h, 11);

cout << "++++++++++++++++";

SetConsoleTextAttribute(h, 6);

cout << "---------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "-------== "<<endl;

cout<<" =--------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "------------";

SetConsoleTextAttribute(h, 11);

cout << "=\*\*\*\*\*++++\*\*\*\*\*=";

SetConsoleTextAttribute(h, 6);

cout << "---------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "--------= "<<endl;

cout<<" +=-------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "------------";

SetConsoleTextAttribute(h, 11);

cout << "=\*============\*=";

SetConsoleTextAttribute(h, 6);

cout << "---------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "-------=+ "<<endl;

cout<<" +=-------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "------------";

SetConsoleTextAttribute(h, 11);

cout << "=#\*\*\*- -\*\*\*#=";

SetConsoleTextAttribute(h, 11);

cout << "---------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "---------------";

SetConsoleTextAttribute(h, 11);

cout << "=";

SetConsoleTextAttribute(h, 6);

cout << "-------=+ "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 11);

cout << "==========";

SetConsoleTextAttribute(h, 6);

cout << "-";

SetConsoleTextAttribute(h, 11);

cout << "===============";

SetConsoleTextAttribute(h, 6);

cout << "-";

SetConsoleTextAttribute(h, 11);

cout << "=============-+#\* \*#+-==========";

SetConsoleTextAttribute(h, 6);

cout << "-";

SetConsoleTextAttribute(h, 11);

cout << "===============";

SetConsoleTextAttribute(h, 6);

cout << "-";

SetConsoleTextAttribute(h, 11);

cout << "==========";

SetConsoleTextAttribute(h, 6);

cout << ""<<endl;

cout<<" =---------=---------------=--------------";

SetConsoleTextAttribute(h, 11);

cout << "+#\* \*#+";

SetConsoleTextAttribute(h, 6);

cout << "-----------=---------------=---------="<<endl;

cout<<" ###########################################################################################"<<endl;

cout<<" SHAH RUKHN-e-ALAM Multan"<<endl;

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void lahoreinterface()

{

system("cls");

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

SetConsoleTextAttribute(h, 8);

cout<<" \* "<<endl;

cout<<" \* "<<endl;

cout<<" \* "<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" :#: "<<endl;

cout<<" .:::::::. "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 8);

cout << "+";

SetConsoleTextAttribute(h, 15);

cout << " .:::::::::::. ";

SetConsoleTextAttribute(h, 8);

cout << "+";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 8);

cout << "+";

SetConsoleTextAttribute(h, 15);

cout << " .:::::::::::::. ";

SetConsoleTextAttribute(h, 8);

cout << "+";

SetConsoleTextAttribute(h, 15);

cout << " "<<endl;

cout<<" .::. --- ";

SetConsoleTextAttribute(h, 12);

cout << "#";

SetConsoleTextAttribute(h, 15);

cout << "...:::::::::::::::...";

SetConsoleTextAttribute(h, 12);

cout << "#";

SetConsoleTextAttribute(h, 15);

cout << " --- .::. "<<endl;

cout<<" \*\*\*\*\* .:::::. ";

SetConsoleTextAttribute(h, 12);

cout << "#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#";

SetConsoleTextAttribute(h, 15);

cout << " .:::::. \*\*\*\*\* "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 12);

cout << ".##%%##";

SetConsoleTextAttribute(h, 15);

cout << " .:::::::::. ";

SetConsoleTextAttribute(h, 12);

cout << "#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#";

SetConsoleTextAttribute(h, 15);

cout << " .:::::::::. ";

SetConsoleTextAttribute(h, 12);

cout << "##%%##. "<<endl;

cout<<" ++######";

SetConsoleTextAttribute(h, 15);

cout << " .::. .:::::::::::::. ";

SetConsoleTextAttribute(h, 12);

cout << "#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#";

SetConsoleTextAttribute(h, 15);

cout << " .:::::::::::::. .::. ";

SetConsoleTextAttribute(h, 12);

cout << "######++ "<<endl;

cout<<" ++######";

SetConsoleTextAttribute(h, 15);

cout << " #### ::::::::::::::::: ";

SetConsoleTextAttribute(h, 12);

cout << "#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#";

SetConsoleTextAttribute(h, 15);

cout << " ::::::::::::::::: \*####\* ";

SetConsoleTextAttribute(h, 12);

cout << "######++ "<<endl;

cout<<" ++##### +##+";

SetConsoleTextAttribute(h, 15);

cout << " +++++++++++++++++ ";

SetConsoleTextAttribute(h, 12);

cout << "#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#";

SetConsoleTextAttribute(h, 15);

cout<< " +++++++++++++++++ ";

SetConsoleTextAttribute(h, 12);

cout << "+##+ #####++ "<<endl;

cout<<" ++##### ######";

SetConsoleTextAttribute(h, 15);

cout << " ================= ";

SetConsoleTextAttribute(h, 12);

cout << "#\*\*\*\*\*\*\*\*#####\*\*\*\*\*\*\*\*#";

SetConsoleTextAttribute(h, 15);

cout << " ================= ";

SetConsoleTextAttribute(h, 12);

cout << "###### #####++ "<<endl;

cout<<" ++\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*####### #######\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*++ "<<endl;

cout<<" ++++++++++++++++++++++++++++++++++++++++++# #++++++++++++++++++++++++++++++++++++++++++ "<<endl;

SetConsoleTextAttribute(h, 4);

cout<<" ++++##################################++++# ";

SetConsoleTextAttribute(h, 11);

cout << "Lahore";

SetConsoleTextAttribute(h, 4);

cout << " #++++#################################+++++ "<<endl;

SetConsoleTextAttribute(h, 12);

cout<<" ++\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*++# #++\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*++ "<<endl;

cout<<" ++\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*++# #++\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*++ "<<endl;

cout<<" ++\*+\*\*\*\*\*\*+\*\*\*\*\*\*+\*\*\*\*\*\*+\*\*\*\*\*\*+\*\*\*\*\*\*\*\*++# #++\*\*\*\*\*\*\*\*+\*\*\*\*\*\*+\*\*\*\*\*\*\*\*\*\*\*\*\*+\*\*\*\*\*\*+\*++ "<<endl;

cout<<" ";

SetConsoleTextAttribute(h, 4);

cout << "###";

SetConsoleTextAttribute(h, 15);

cout << "\*\*\*: :\*\*\*: :\*\*\*: :\*\*\*: :\*\*\*: :\*\*\*++";

SetConsoleTextAttribute(h, 4);

cout <<"#";

SetConsoleTextAttribute(h, 15);

cout << " @@ ";

SetConsoleTextAttribute(h, 4);

cout << "#";

SetConsoleTextAttribute(h, 15);

cout << "++\*\*\*: :\*\*\*: :\*\*\*: :\*\*\*: :\*\*\*: :\*\*\*";

SetConsoleTextAttribute(h, 4);

cout << "###"<<endl;

cout<<" ###";

SetConsoleTextAttribute(h, 15);

cout << "\*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*++";

SetConsoleTextAttribute(h, 4);

cout <<"#";

SetConsoleTextAttribute(h, 15);

cout << " @@@@ ";

SetConsoleTextAttribute(h, 4);

cout << "#";

SetConsoleTextAttribute(h, 15);

cout << "++\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\*";

SetConsoleTextAttribute(h, 4);

cout << "###"<<endl;

cout<<" ###";

SetConsoleTextAttribute(h, 15);

cout << "\*\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*++";

SetConsoleTextAttribute(h, 4);

cout <<"#";

SetConsoleTextAttribute(h, 15);

cout << " @@@@ ";

SetConsoleTextAttribute(h, 4);

cout << "#";

SetConsoleTextAttribute(h, 15);

cout << "++\*\* \*\*\* \*\*\* \*\*\* \*\*\* \*\*\*";

SetConsoleTextAttribute(h, 4);

cout << "###"<<endl;

cout<<" ++++++++++++++++++++++++++++++++++++++++++++%%%%%%%%%%%%%++++++++++++++++++++++++++++++++++++++++++++"<<endl;

cout<<" BADSHAHI MOSQUE LAHORE"<<endl;

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void islamabadinterface()

{

system("cls");

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

cout<<""<<endl;

SetConsoleTextAttribute(h, 15);

cout<<" @@@ @@@ "<<endl;

cout<<" @@@ @@@ "<<endl;

cout<<" @@@@@ @@@@@ "<<endl;

cout<<" @@@@@ @@@@@ "<<endl;

cout<<" @@@@@ @@@@@ "<<endl;

cout<<" @@@@@@@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << " @@ ";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 8);

cout << " @@ ";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << " @@ ";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 8);

cout << " @@ ";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << " @@ ";

SetConsoleTextAttribute(h, 15);

cout << " ";

SetConsoleTextAttribute(h, 8);

cout << " @@ ";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@@@@@@@@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@@@@ @@@ @@@@@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@ @@@ @@@@@@@ @@@ @@@@@ ";

SetConsoleTextAttribute(h, 8);

cout << "@@@@";

SetConsoleTextAttribute(h, 15);

cout << " @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@@@@@@@ @@@@ @@@@ @@@@ @@@@ @@@@@@@@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@@ @@@ @@@ @ @ @@@ @@@ @@@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@ @@@@ @@@@ @ @ @@@@ @@@@ @@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@ @@@@ @@@@ @ @ @ @ @@@@ @@@@ @@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@ @@@ @@@@@ @ @ @ @ @@@@@ @@@@ @@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@ @@@@ @@@@ @@ @ @ @ @ @@ @@@@ @@@@ @@ @@@@@@@ "<<endl;

cout<<" @@@@@@@ @@ @@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@ @@ @@@@@@@ "<<endl;

cout<<" @@@ @@@@ @@ @@@@ @@@@@@@@ @@@@@@@@@@@@@@@ @@@@@ @@@@@@@@@@@@@@@@@@@@@@@ @@ @@@@ @@@ "<<endl;

cout<<" @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ @@@@ @@@@ @@@ @@@ @@@ @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"<<endl;

cout<<" @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"<<endl;

cout<<" FAISAL MOSQUE ISLAMABAD"<<endl;

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

string login()

{

//this is the login menu as the loading of the application completed this menu opens as the user have to sign up or sign in first

string log;

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Login Menu";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "1. Sign-In(Already have an account): " << endl;

gotoxy(x, y + 1);

SetConsoleTextAttribute(h, 1);

cout << "2. Sign-Up: " << endl;

gotoxy(x, y + 2);

SetConsoleTextAttribute(h, 8);

cout << "3. Exit: " << endl;

gotoxy(x, y + 3);

SetConsoleTextAttribute(h, 1);

cout << "Which function would you like to perform....";

cin >> log;

return log;

}

void signup(int &size, int &admin, string firstname[], string secondname[], string gender[], string city[], string nationality[], string contactnumber[], string cnic[], string username[], string userpassword[], string userrole[], string filename)

{

//this is the signup menu in this all the things are inquired bout the user and all information should be accurate

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"SignUp";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while (true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "Enter your First-Name: ";

getline(cin, firstname[size]);

if(checkenter(firstname[size])==0)

{

//this is used to check if the user enter empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck(firstname[size])==0)

{

//this is used to check the digits if the user enter digits then it shows false

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

//if the user enter same name according to our demand then the loop breaks

break;

}

}

while (true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout << "Enter your Last-Name: ";

getline(cin, secondname[size]);

if(checkenter(secondname[size])==0)

{

//this is used to check wheather the user enter empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(digitscheck(secondname[size])==0)

{

//this is used to check if the user enter digits in the name

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<" Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

//if the user enter the name according to our validations then the loop breaks

break;

}

}

while (true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout << "Gender(in small letters): ";

getline(cin, gender[size]);

if(checkenter(gender[size])==0)

{

//this is used to check the wheather the user enter the empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(gendercheck(gender[size])==0)

{

//this is used to check the gender if gender is wrong then the input is taken again

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Wrong ones. Such type of gender doesn't exist."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

//the loop breaksif the user enter the same inut according to our requirements

break;

}

}

while (true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout << "City: ";

//cin.ignore();

getline(cin, city[size]);

if(checkenter(city[size])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(digitscheck(city[size])==0)

{

//this is used to check the digits in the input

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

//the loop breaks if the user enter the same input according to our requirments

break;

}

}

while (true)

{

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout << "Nationality: ";

getline(cin, nationality[size]);

if(checkenter(nationality[size])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else if(digitscheck(nationality[size])==0)

{

//this is to check wheather the user input digits

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else

{

break;

}

}

while (true)

{

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout << "Enter your contact number: ";

getline(cin, contactnumber[size]);

string contactnumberCheck = contactnumber[size];

if(checkenter(contactnumber[size])==0)

{

//this is used to check wheather input empty

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else if (contactnumberCheck.length() != 11)

{

//this is used to check the length of the number

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else if(charactermCheck(contactnumber[size])==0)

{

//this is used to check wheather the user input the characters

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else if(recursionCheck(contactnumberCheck,contactnumber, size)==0)

{

//this is used to check wheather the same number is input already

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"The contact number has been already entered. Try another one."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else

{

break;

}

}

while (true)

{

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout << "Enter your CNIC: ";

getline(cin, cnic[size]);

string cnicCheck = cnic[size];

if ((checkenter(cnic[size])==0))

{

//this is used to check wheather the user input empty

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else if (cnicCheck.length() != 13)

{

//thisi is used to check the length of the number

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else if(charactermCheck(cnic[size])==0)

{

//this is used to check wheather the user input character

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else if(recursionCheck(cnic[size], cnic, size)==0)

{

//this is used wheather if the input is already entred by someone

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"The CNIC has been already entered. Try another one."<<endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else

{

break;

}

}

while (true)

{

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 1);

cout << "Choose your user-name: ";

getline(cin, username[size]);

if(checkenter(username[size])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+7);

cout<<" "<<endl;

}

else if(digitscheck(username[size])==0)

{

//this is used to check wheather the input digits

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+7);

cout<<" "<<endl;

}

else if(recursionCheck(username[size], username, size)==0)

{

//this is used to check wheather the input is already entered

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 4);

cout<<"The username has been already taken. Try another one."<<endl;

getch();

gotoxy(x, y+7);

cout<<" "<<endl;

}

else

{

break;

}

}

while (true)

{

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 1);

cout << "Enter Password: ";

getline(cin, userpassword[size]);

if(checkenter(userpassword[size])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+8);

cout<<" "<<endl;

}

else if(recursionCheck(userpassword[size], userpassword, size)==0)

{

//this is used to ceck wheather the input is already entered

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 4);

cout<<"The password has been already taken. Try again."<<endl;

getch();

gotoxy(x, y+8);

cout<<" "<<endl;

}

else

{

break;

}

}

while (true)

{

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 1);

cout << "Enter your role(admin, doctor, patient, pharmacist, equipments-manager): ";

getline(cin, userrole[size]);

string userroleCheck = userrole[size];

if(checkenter(userrole[size])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+9);

cout<<" "<<endl;

}

else if (userroleCheck != "admin" && userroleCheck != "doctor" && userroleCheck != "patient" && userroleCheck != "pharmacist" && userroleCheck != "equipments-manager")

{

//this is used to check wheather the role match accoring to the given format

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 4);

cout << "Role you entered is unavailable. Please select from from the given ones."<<endl;

getch();

gotoxy(x, y+9);

cout<<" "<<endl;

}

else if(userrole[size]=="admin" && admin==1)

{

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 4);

cout<<"The admin has already signed-up. You are not an admin."<<endl;

getch();

gotoxy(x, y+10);

cout<<" "<<endl;

}

else if(digitscheck(userrole[size])==0 && userrole[10]=="-")

{

//this is used to check wheather the user input digits

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+9);

cout<<" "<<endl;

}

else

{

break;

}

}

bool temp = true;

for (int i = 0; i < size; i++)

{

//check the user name password

if (username[size] == username[i] && userpassword[size] == userpassword[i])

{

SetConsoleTextAttribute(h, 4);

cout << "Invalid user name and password.Please correct another one." << endl;

SetConsoleTextAttribute(h, 4);

cout << "Please engage in the progression by selecting a key to proceed further..." << endl;

temp = false;

break;

}

}

if (temp == true)

{

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 2);

cout << "Welcome to Medi Harmony platform as a " << userrole[size] << endl;

gotoxy(x, y+11);

SetConsoleTextAttribute(h, 2);

cout<<"Now to see menu you first have to sign in."<<endl;

gotoxy(x, y+12);

SetConsoleTextAttribute(h, 2);

cout<<"Press any key to go back to the login menu."<<endl;

size++;

count++;

filehandlingsignup(size, firstname, secondname, gender, city, nationality, contactnumber, cnic, username, userpassword, userrole, filename);

}

getch();

}

bool recursionCheck(string check,string recursion[], int size)

{

//this function is used to match the input with the already entered inputs if matches then return false

for (int i = 0; i<size; i++)

{

if (recursion[i] == check)

{

return false;

}

}

return true;

}

bool digitscheck(string digits)

{

//this is used to check character is the user input character then return false

for(int i=0; i<digits.length(); i++)

{

int a=digits[i];

if(isalpha(a))

{

continue;

}

else

{

return false;

}

}

return true;

}

bool charactermCheck(string characterCheck)

{

//this is used to check the digits if the user input character then return false

for (int i = 0; i < characterCheck.length(); i++)

{

char c = characterCheck[i];

if (isdigit(c))

{

continue;

}

else

{

return false;

}

}

return true;

}

bool checkenter(string check)

{

//this function return false if the user input empty

if(check=="" || check==" " || check==" " || check==" " || check==" " || check==" " || check==" " || check==" ")

{

return false;

}

else

{

return true;

}

}

bool gendercheck(string check)

{

//this function return false if the user entered wrong input

if(check!="male" && check!="female" && check!="custom")

{

return false;

}

else

{

return true;

}

}

bool checkshiftin(string check[])

{

//this function return false if the user not entered input according to th egiven format

if(check[2]!="/" && check[5]!="/")

{

return false;

}

else

{

return true;

}

}

void signin(int size, string names, string passwords, string contactnumber[], string userrole[], string username[], string userpassword[], string &rolereturn, string cnic[], string nationality[], string city[], string gender[], string firstname[], string secondname[])

{

while(true){

//this the sign in function

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Sign-In";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

gotoxy(x, y);

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "Enter your user-name: ";

getline(cin, names);

if (checkenter(names)==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout << "Please provide valid input; empty entries are not accepted. Please try again." << endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout << "Enter Password: ";

getline(cin, passwords);

if (checkenter(passwords)==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout << "Please provide valid input; empty entries are not accepted. Please try again." << endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

int check=0;

for (int i = 0; i < size; i++)

{

if (passwords == userpassword[i] && names == username[i])

{

//this function will return true if the password and username matches

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

cout << "Welcome to our " << userrole[i] << " menu. Press any key to continue....";

rolereturn=userrole[i];

getch();

check=1;

break;

}

}

if (check==1)

{

break;

}

else

{

gotoxy(x,y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Password or Username incorrect...";

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+4);

getch();

break;

}

}

}

string adminmenu()

{

//this the admin menu

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Admin Menu";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

string adminoption;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "Select one of the following option..." << endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout << "1. Add Patient." << endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout << "2. View Patient." << endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout << "3. Discharge Patient." << endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout << "4. Analyzing financial returns or evaluating the profitability aspect." << endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout << "5. Changing Salary." << endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout << "6. Add Bed." << endl;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 1);

cout << "7. Add Doctor." << endl;

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 1);

cout << "8. View Doctor List." << endl;

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 1);

cout << "9. Add Equipments-Manager." << endl;

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 1);

cout << "10. Change Equipments-Manager." << endl;

gotoxy(x, y+11);

SetConsoleTextAttribute(h, 1);

cout << "11. Update patient." << endl;

gotoxy(x, y+12);

SetConsoleTextAttribute(h, 4);

cout << "Note: Enter your option from the above menu otherwise you will be shifted back to login page.";

gotoxy(x, y+13);

SetConsoleTextAttribute(h, 1);

cout<<"Enter your option....";

cin >> adminoption;

return adminoption;

}

void addpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient)

{

//this is the add patient menu

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Patient";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter Patient name: ";

getline(cin, patientname[addpatientindex]);

if(checkenter(patientname[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck(patientname[addpatientindex])==0)

{

//this is used to check wheather the user input digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Age: ";

getline(cin, patientage[addpatientindex]);

string patientageCheck=patientage[addpatientindex];

if(checkenter(patientage[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(patientage[addpatientindex])==0)

{

//this is used to check wheather the user input character

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter CNIC: ";

getline(cin, patientcnic[addpatientindex]);

string patientcnicCheck=patientcnic[addpatientindex];

if(checkenter(patientcnic[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if (patientcnicCheck.length() != 13)

{

//this is used to check the length of the input

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(patientcnic[addpatientindex])==0)

{

//this is used to check wheather the user input character

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(recursionCheck(patientcnicCheck, patientcnic, addpatientindex)==0)

{

//this is used to check wheather the user input is already entered by someone

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The CNIC have been already entered. Try another one."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"Blood Group(): ";

cin>>patientbloodgroup[addpatientindex];

if(checkenter(patientbloodgroup[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(patientbloodgroup[addpatientindex]!="O-" && patientbloodgroup[addpatientindex]!="A+" && patientbloodgroup[addpatientindex]!="O+" && patientbloodgroup[addpatientindex]!="A-" && patientbloodgroup[addpatientindex]!="B+" && patientbloodgroup[addpatientindex]!="B-" && patientbloodgroup[addpatientindex]!="AB+" && patientbloodgroup[addpatientindex]!="AB-")

{

//this is used to check wheather the user input is invalid

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly sure that the enter blood group is valid."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"Medicle History(yes or no): ";

cin>>patienthistory[addpatientindex];

if(checkenter(patienthistory[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else if(patienthistory[addpatientindex]!="yes" && patienthistory[addpatientindex]!="no")

{

//this is used to check wheather the user input invalid

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Respected user, we kindly request your input as per the provided guidance. Please try again."<<endl;

getch();

gotoxy(x, y+4);

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"Date of Visit(dd/mm/yyyy): ";

cin>>visitdatepatient[addpatientindex];

if(checkenter(visitdatepatient[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else if(checkshiftin==0)

{

//this is used to check wheather the user input in a given format

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Respected user, we kindly request your input as per the provided guidance. Please try again."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

bool temp=false;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout<<"Enter patient's phone number: ";

cin>>patientphonenumber[addpatientindex];

string patientphonenumberCheck=patientphonenumber[addpatientindex];

if(checkenter(patientphonenumber[addpatientindex])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

temp = true;

// getch();

// gotoxy(x, y+6);

// cout<<" "<<endl;

}

else if (patientphonenumberCheck.length() != 11)

{

//this is used to check wheather the user input length

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

temp=true;

// getch();

// gotoxy(x, y+6);

// cout<<" "<<endl;

}

else if(charactermCheck(patientphonenumber[addpatientindex])==0)

{

//this is used to check wheather the user input characters

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

temp=true;

// getch();

// gotoxy(x, y+6);

// cout<<" "<<endl;

}

else if(recursionCheck(patientphonenumberCheck, patientphonenumber, addpatientindex)==0)

{

//this is used to check wheather the user input is alreday entered

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"The phone number have been already entered. Try again."<<endl;

temp=true;

// getch();

// cout<<" "<<endl;

}

// else

// {

// break;

// }

if(temp==true){

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else

{

filehandlingaddpatient(addpatientindex, patientname, patientage, patientcnic, patientbloodgroup, patienthistory, visitdatepatient, patientphonenumber, filenameaddpatient);

break;

}

}

addpatientindex++;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 2);

cout<<"The addition of the patient has been completed successfully."<<endl;

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+8);

cout<<"Press any key to continue...";

getch();

}

void viewpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visibledatepatient[], string patientphonenumber[])

{

//this is used to view the list of the patient

system("cls");

if(addpatientindex==0)

{

//if the no patient is entered

cout<<"There are currently no records matching the patient criteria."<<endl;

}

else

{

gotoxy(x1, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Name";

gotoxy(x1+20, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Age";

gotoxy(x1+30, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"CNIC";

gotoxy(x1+48, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Blood Group";

gotoxy(x1+62, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"History";

gotoxy(x+75, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Date";

gotoxy(x1+90, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Phone No.";

for(int c=0; c<addpatientindex; c++)

{

gotoxy(x1, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<patientname[c];

gotoxy(x1+20, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<patientage[c];

gotoxy(x1+30, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<patientcnic[c];

gotoxy(x1+48, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<patientbloodgroup[c];

SetConsoleTextAttribute(h, 2);

gotoxy(x1+62, y\_1+c);

cout<<patienthistory[c];

gotoxy(x1+75, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<visibledatepatient[c];

gotoxy(x1+90, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<patientphonenumber[c];

}

}

gotoxy(x1+20, y\_1+20);

cout<<"Press any key to contniue...";

getch();

}

void dischargepatient(string dischargepatientcnic, int addpatientindex, string patientcnic[], string patientname[], string patientage[], string patientbloodgroup[], string patienthistory[], string patientphonenumber[], string visitdatepatient[], string filenameaddpatient)

{

//this function will discharge the patient if the user cnic matches

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Discharge Patient";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int a;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the CNIC of the patient you want to discharge: ";

getline(cin, dischargepatientcnic);

if(checkenter(dischargepatientcnic)==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if (dischargepatientcnic.length() != 13)

{

//this is used to check wheather the user input is not according to the given format

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(dischargepatientcnic)==0)

{

//this is used to check wheather the user input character

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

for (int x = 0; patientcnic[x] != ""; x++)

{

if (dischargepatientcnic==patientcnic[x])

{

for (int i = x; patientcnic[i] != ""; i++)

{

patientname[i] =patientname[i + 1];

patientage[i]=patientage[i + 1];

patientcnic[i]=patientcnic[i + 1];

patientbloodgroup[i]=patientbloodgroup[i + 1];

patienthistory[i]=patienthistory[i+1];

visitdatepatient[i]=visitdatepatient[i+1];

patientphonenumber[i]=patientphonenumber[i+1];

}

addpatientindex--;

break;

}

}

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

filehandlingaddpatient(addpatientindex, patientname, patientage, patientcnic, patientbloodgroup, patienthistory, visitdatepatient, patientphonenumber, filenameaddpatient);

cout<<"The discharge process for the patient has been completed successfully."<<endl;

gotoxy(60, 22);

SetConsoleTextAttribute(h, 4);

cout << "Press any key to continue...";

getch();

}

void updatepatient(string updatepatientcnic, string patientcnic[], int &addpatientindex, string patientname[], string patientage[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient)

{

//this is used to update the patient

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Update Patient";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int c=0;

int a=-1;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the CNIC of the patient you want to update: ";

getline(cin, updatepatientcnic);

if(checkenter(updatepatientcnic)==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if (updatepatientcnic.length()!=13)

{

//this is used to check wheather the user input length

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(updatepatientcnic)==0)

{

//this is used to check wheather the user input characters

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

cout<<" "<<endl;

}

else

{

break;

}

}

for(int i=0; i<addpatientindex; i++)

{

if(patientcnic[i]==updatepatientcnic)

{

a=0;

c=i;

break;

}

}

if(a==0)

{

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter Patient name: ";

getline(cin, patientname[c]);

if(checkenter(patientname[c])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

if(digitscheck(patientname[c])==0)

{

//this is used to check wheather the user input digits

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"Age: ";

getline(cin, patientage[c]);

string patientageCheck=patientage[c];

if(checkenter(patientage[c])==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(charactermCheck(patientage[c])==0)

{

//this is used to check wheather the user input alphabets

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

filehandlingaddpatient(addpatientindex, patientname, patientage, patientcnic, patientbloodgroup, patienthistory, visitdatepatient, patientphonenumber, filenameaddpatient);

break;

}

}

}

else if(a==-1)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"The patient with this CNIC were not added. You have to check the patient list."<<endl;

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+4);

cout<<"Press any key to continus...";

getch();

}

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 2);

cout<<"The requested patient has been updated successfully"<<endl;

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+5);

cout<<"Press any key to continue...";

}

void profitloss(string patientcharges, int addpatientindex, string filenameprofitloss)

{

//this is used t view the profit

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

if(addpatientindex==0)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

cout<<"No money earned."<<endl;

}

else

{

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Profit Loss";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Charges per patient: ";

getline(cin, patientcharges);

if(checkenter(patientcharges)==0)

{

//this is used to check wheather the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck==0)

{

//this is used to check wheather the user input alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

filehandlingprofitloss(patientcharges, addpatientindex, filenameprofitloss);

break;

}

}

}

int b=stoi(patientcharges);

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

cout<<"Money Earned from patient: "<<b\*addpatientindex;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void changingsalary(string doctorsalary, string managersalary, string filenamechangingsalary)

{

//this is used to change the salary or update saalary of doctors

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Salary Change";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the Salary of the doctors: ";

getline(cin, doctorsalary);

if(checkenter(doctorsalary)==0)

{

//this is used to check if the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck==0)

{

//this is used to check if the user input alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the Salary of the manager: ";

getline(cin, managersalary);

if(checkenter(managersalary)==0)

{

//this is used to check if the user input empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck==0)

{

//this is used to check if the user input alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure you have entered only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

filehandlingchangingsalary(doctorsalary, managersalary, filenamechangingsalary);

break;

}

}

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+2);

cout<<"Press any key to continue...";

}

void addbed(string bedadd, string filenameaddbed)

{

//this is used to add bed

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Bed";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the number of bed you want to enter: ";

getline(cin, bedadd);

if(checkenter(bedadd)==0)

{

//this is used to check if the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please amke sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

filehandlingaddbed(bedadd, filenameaddbed);

break;

}

}

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

cout<<bedadd<<" beds added successfully."<<endl;

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void adddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor)

{

//this is used to add doctors in the hospital

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Doctor";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Doctor's Name: ";

getline(cin, doctorname[adddoctorindex]);

if(checkenter(doctorname[adddoctorindex])==0)

{

//this is used to check if the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Doctor Specialiaty: ";

getline(cin, doctorspeciality[adddoctorindex]);

if(checkenter(doctorspeciality[adddoctorindex])==0)

{

//this is used to check if the user input empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input digits

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

cout<<" "<<endl;

}

else

{

adddoctorindex++;

filehandlingadddoctor(doctorname, doctorspeciality, adddoctorindex, filenameadddoctor);

break;

}

}

SetConsoleTextAttribute(h, 2);

gotoxy(x, y+2);

cout<<"The doctor has been added successfully."<<endl;

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+3);

cout<<"Press ant key to continue...";

getch();

}

void viewdoctor(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex)

{

//this function will view the doctors in the system

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

if(adddoctorindex==0)

{

SetConsoleTextAttribute(h, 2);

cout<<"At present, the records indicate a absence of registered doctors."<<endl;

}

else

{

gotoxy(x1, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Doctor Name";

gotoxy(x1+20, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Doctor's Speciality";

gotoxy(x1+45, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Hired/Free";

for(int c=0; c<adddoctorindex; c++)

{

gotoxy(x1, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<doctorname[c];

gotoxy(x1+20, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<doctorspeciality[c];

gotoxy(x1+45, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<hire[c];

}

}

gotoxy(x, y+20);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void addequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameaddequipmentmanager)

{

//this is use to add the equipment manager in th esystem

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Equipment manager";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

if(managercheck==0)

{

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the name of the equipment-manager: ";

getline(cin, equipmentmanagername);

if(checkenter(equipmentmanagername)==0)

{

//this is used to check if the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter Gender: ";

getline(cin, equipmentmanagergender);

if(checkenter(equipmentmanagergender)==0)

{

//this is used to check if the user input empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(gendercheck(equipmentmanagergender)==0)

{

//this is used to check if the user input is not according to the format

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Wrong ones. Such type of gender does not exist."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter CNIC: ";

getline(cin, equipmentmanagercnic);

if ((checkenter(equipmentmanagercnic)==0))

{

//this is used to check if the user input empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if (equipmentmanagercnic.length() != 13)

{

//this is used to check if the user input length

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentmanagercnic)==0)

{

//this is used to check if the user input alphabets

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

managercheck++;

filehandlingaddequipmentmanager(equipmentmanagername, equipmentmanagergender, equipmentmanagercnic, managercheck, filenameaddequipmentmanager);

break;

}

}

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 2);

cout<<"The manager is added successfully."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+5);

cout<<"Press any key to continue...";

getch();

}

else

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 2);

cout<<"Manager is already entered."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

}

void changeequipmentmanager(int &managercheck, string &equipmentmanagername, string &equipmentmanagergender, string &equipmentmanagercnic, string filenameaddequipmentmanager)

{

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

if(managercheck==1)

{

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter new manager name: ";

getline(cin, equipmentmanagername);

if(checkenter(equipmentmanagername)==0)

{

//this is used to check if the user input empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Gender: ";

getline(cin, equipmentmanagergender);

if(checkenter(equipmentmanagergender)==0)

{

//this is used to check if the user input empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(gendercheck(equipmentmanagergender)==0)

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Wrong ones. Such type of gender does not exist."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"CNIC: ";

getline(cin, equipmentmanagercnic);

if ((checkenter(equipmentmanagercnic)==0))

{

//this is used to check if the user input empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if (equipmentmanagercnic.length() != 13)

{

//this is used to check if the user input length

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentmanagercnic)==0)

{

//this is used to check if the user input alphabets

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

filehandlingaddequipmentmanager(equipmentmanagername, equipmentmanagergender, equipmentmanagercnic, managercheck, filenameaddequipmentmanager);

break;

}

}

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 2);

cout<<"The manager has been updated successfully."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

else

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

cout<<"Prior to assuming the managerial position, it is necessary to first include a manager in the system."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

}

string pateintmenu()

{

//thsi function will give patient menu

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Patient Menu";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

string pateintoption;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "Select one of the following option..." << endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"1. View Doctor Schedule." <<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"2. View Available Doctor List."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"3. Available Beds."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"4. Select Doctor."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"5. Billing and invoice."<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout<<"6. View Prescription."<<endl;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 1);

cout<<"7. Satisfactory Report."<<endl;

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 1);

cout<<"8. Buy Medicines."<<endl;

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 4);

cout << "Note: Enter your option from the above menu otherwise you will be shifted back to login page.";

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 1);

cout<<"Enter your option...";

cin>>pateintoption;

return pateintoption;

}

void doctorschedule(string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex)

{

//this function will show the doctors

system("cls");

viewdoctor(doctorname, doctorspeciality,hire, adddoctorindex);

gotoxy(x1, y\_1+21);

SetConsoleTextAttribute(h, 2);

cout<<"NOTE: All the doctors are available from 9pm to 10am. Press any key to continue...";

}

void viewdoctorslist (string doctorname[], string doctorspeciality[],string hire[], int &adddoctorindex)

{

//this function will view doctors list to patient

system("cls");

viewdoctor(doctorname, doctorspeciality,hire, adddoctorindex);

gotoxy(x1+50, y\_1+20);

cout<<"Press any key to continue...";

getch();

}

void availablebeds(string bedadd, int &addpatientindex)

{

//to show available bed in the beds in the hospital

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Available Beds";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

gotoxy(x, y);

SetConsoleTextAttribute(h, 2);

cout<<"The number of beds available are "<<bedadd<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<" %%%% %%%% "<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%% %%%%% "<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%% %%%% %%%%%% "<<endl;

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%% %%%% %%%%%% "<<endl;

gotoxy(x, y+11);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+12);

SetConsoleTextAttribute(h, 1);

cout<<" @%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%@ "<<endl;

gotoxy(x, y+13);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+14);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+15);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+16);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+17);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

gotoxy(x, y+18);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"<<endl;

gotoxy(x, y+19);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"<<endl;

gotoxy(x, y+20);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"<<endl;

gotoxy(x, y+21);

SetConsoleTextAttribute(h, 1);

cout<<" @%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"<<endl;

gotoxy(x, y+22);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%% %%%%%# "<<endl;

gotoxy(x, y+23);

SetConsoleTextAttribute(h, 1);

cout<<" %%%%%% %%%%%# "<<endl;

gotoxy(x, y+24);

SetConsoleTextAttribute(h, 1);

cout<<" %%%% %%%% "<<endl;

gotoxy(x, y+25);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

void selectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[])

{

//this function will display the doctors selection

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

int b;

if(adddoctorindex==0)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"We are sorry to say you that at present no doctor is on his duty."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

else if(adddoctorindex!=0)

{

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Doctor Appointment";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the name of the doctor you want to hire: ";

getline(cin, selectdoctorname);

if(checkenter(selectdoctorname)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the issue are you seeking to address:";

getline(cin, selectdoctorspeciality);

if(checkenter(selectdoctorspeciality)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(digitscheck==0)

{

//this is used to check if the user input is digits

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you entered only alphabets."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

filehandlingselectdoctor(selectdoctorname, selectdoctorspeciality, adddoctorindex, doctorname, doctorspeciality, hire);

break;

}

}

int g;

//will show the hired in the list

for(int f=0; f<adddoctorindex; f++)

{

if(selectdoctorname==doctorname[f] && selectdoctorspeciality==doctorspeciality[f])

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

cout<<"Successfully! "<<selectdoctorname<<", an "<<selectdoctorspeciality<<" specialist has been hired by you."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

g=f;

hire[g]="hire";

break;

}

else

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The doctor is not added successfully."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

}

}

}

void billinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[])

{

//this function will calculate the bills

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Bill Pay";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the service type(emergency or normal): ";

getline(cin, servicetype[addbillingindex]);

if(checkenter(servicetype[addbillingindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(servicetype[addbillingindex]!="emergency" && servicetype[addbillingindex]!="normal")

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly adhere to the provided instructions and provide the necessary input."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the number of days you stayed: ";

getline(cin, daysstayes[addbillingindex]);

if(checkenter(daysstayes[addbillingindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(daysstayes[addbillingindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter rooom type(VIP or AC or normal): ";

getline(cin, roomtype[addbillingindex]);

if(checkenter(roomtype[addbillingindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(roomtype[addbillingindex]!="VIP" && roomtype[addbillingindex]!="AC" && roomtype[addbillingindex]!="normal")

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly adhere to the provided instructions and provide the necessary input."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

filehandlingbillinginvoice(servicetype, addbillingindex, daysstayes, roomtype);

break;

}

}

//here call the bills function to show with cls

bilslip(servicetype, daysstayes, roomtype, addbillingindex);

}

void bilslip(string servicetype[], string daysstayes[], string roomtype[], int &addbillingindex)

{

//this function will show you the bill check

system("cls");

gotoxy(x1+7, y\_1-2);

SetConsoleTextAttribute(h, 2);

cout<<"Check Bill"<<endl;

int i;

if(servicetype[addbillingindex]=="emergency")

{

i=10000;

}

else if(servicetype[addbillingindex]=="normal")

{

i=5000;

}

gotoxy(x1, y\_1);

SetConsoleTextAttribute(h, 1);

cout<<"Service Type Charges: "<<i;

int c=stoi(daysstayes[addbillingindex])\*1000;

gotoxy(x1, y\_1+1);

SetConsoleTextAttribute(h, 1);

cout<<"Stayed Charges: "<<c;

int b;

if(roomtype[addbillingindex]=="VIP")

{

b=5000;

}

else if(roomtype[addbillingindex]=="AC")

{

b=3000;

}

else if(roomtype[addbillingindex]=="normal")

{

b=1000;

}

gotoxy(x1, y\_1+2);

SetConsoleTextAttribute(h, 1);

cout<<"Room Type Charges per day: "<<b;

int total=i+(b\*stoi(daysstayes[addbillingindex]))+c;

gotoxy(x1, y\_1+5);

SetConsoleTextAttribute(h, 1);

cout<<"Total Charges: "<<total;

int doctorfee=1000;

gotoxy(x1, y\_1+3);

SetConsoleTextAttribute(h, 1);

cout<<"Doctor Fee: "<<doctorfee;

gotoxy(x1, y\_1+7);

SetConsoleTextAttribute(h, 2);

cout<<"Thank you for selecting Pysantas Medi Harmony."<<endl;

gotoxy(x1, y\_1+8);

SetConsoleTextAttribute(h, 2);

cout<<"We are dedicated to prioritizing your health and well-being as our primary mission."<<endl;

gotoxy(x1, y\_1+10);

SetConsoleTextAttribute(h, 11);

cout<<" \*\*+-:+\* \*+:-+\*\* "<<endl;

gotoxy(x1, y\_1+11);

SetConsoleTextAttribute(h, 11);

cout<<" #\*+::=-.::+# ##=:::==::+\*# "<<endl;

gotoxy(x1, y\_1+12);

SetConsoleTextAttribute(h, 11);

cout<<" #\*=::-:...:=-::=-::=-:...:-::=\*# "<<endl;

gotoxy(x1, y\_1+13);

SetConsoleTextAttribute(h, 11);

cout<<" \*-:.:...:-# \*:--:+ #\*-....:.:-\* "<<endl;

gotoxy(x1, y\_1+14);

SetConsoleTextAttribute(h, 11);

cout<<" \*-......::+# +==+ #=::......-\* "<<endl;

gotoxy(x1, y\_1+15);

SetConsoleTextAttribute(h, 11);

cout<<" \*-......::--\* +--:.......=\* "<<endl;

gotoxy(x1, y\_1+16);

SetConsoleTextAttribute(h, 11);

cout<<" \*-.......::-:--\* \*--::::.......-\* "<<endl;

gotoxy(x1, y\_1+17);

SetConsoleTextAttribute(h, 11);

cout<<" \*-.........::::::-=\* \*=-::::::.........=# "<<endl;

gotoxy(x1, y\_1+18);

SetConsoleTextAttribute(h, 11);

cout<<" #+:........::::::::::::-+# \*+-::::::::::::.........+ "<<endl;

gotoxy(x1, y\_1+19);

SetConsoleTextAttribute(h, 11);

cout<<" =.....:::----==+++++++=:::\*+::-==+++++===----:::.....+ "<<endl;

gotoxy(x1, y\_1+20);

SetConsoleTextAttribute(h, 11);

cout<<" =.....:--+# \*+-=+# \*+-::.....+ "<<endl;

gotoxy(x1, y\_1+21);

SetConsoleTextAttribute(h, 11);

cout<<" +=.....::-\* ## \*+--:.....= "<<endl;

gotoxy(x1, y\_1+22);

SetConsoleTextAttribute(h, 11);

cout<<" =.....:--\* +-::.....= "<<endl;

gotoxy(x1, y\_1+23);

SetConsoleTextAttribute(h, 11);

cout<<" -.....::-+ #+-::.....= "<<endl;

gotoxy(x1, y\_1+24);

SetConsoleTextAttribute(h, 11);

cout<<" -:....::::=+ +=::::....: "<<endl;

gotoxy(x1, y\_1+25);

SetConsoleTextAttribute(h, 11);

cout<<" ..........: :.......... "<<endl;

gotoxy(x1, y\_1+26);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

string viewprescription()

{

//this function will give you the prescripion of th evarious diseases

system("cls");

header();

string disease;

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"View Prescription";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Catalog of Ailments."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"1. Diabetes."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"2. Hypertension."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"3. Asthma."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"4. Hearth Diseases."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"5. Stroke."<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout<<"6. Influenza."<<endl;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 1);

cout<<"7. Alzheimer's Disease."<<endl;

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 1);

cout<<"8. AIDS."<<endl;

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 1);

cout<<"9. Azma."<<endl;

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 1);

cout<<"10. Epilepsy Disease."<<endl;

gotoxy(x, y+11);

SetConsoleTextAttribute(h, 1);

cout<<"11. Chronic Kidney Disease."<<endl;

gotoxy(x, y+12);

SetConsoleTextAttribute(h, 1);

cout<<"12. Anemia."<<endl;

gotoxy(x, y+13);

SetConsoleTextAttribute(h, 1);

cout<<"13. Gout."<<endl;

gotoxy(x, y+14);

SetConsoleTextAttribute(h, 1);

cout<<"14. Hepatitis."<<endl;

gotoxy(x, y+15);

SetConsoleTextAttribute(h, 1);

cout<<"15. Tuberculosis."<<endl;

gotoxy(x, y+16);

SetConsoleTextAttribute(h, 1);

cout<<"16. Diarrhea."<<endl;

gotoxy(x, y+17);

SetConsoleTextAttribute(h, 1);

cout<<"17. Pneumonia."<<endl;

gotoxy(x, y+18);

SetConsoleTextAttribute(h, 1);

cout<<"18. Goiter."<<endl;

gotoxy(x, y+19);

SetConsoleTextAttribute(h, 1);

cout<<"19. Apendicitis"<<endl;

gotoxy(x, y+20);

SetConsoleTextAttribute(h, 4);

cout<<"Note: Enter your option from the above menu otherwise you will be shifted back to login page."<<endl;

gotoxy(x, y+21);

SetConsoleTextAttribute(h, 1);

cout<<"Please articulate the symptoms manifesting within you...";

cin>>disease;

return disease;

}

void buymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines)

{

//in this function youwill buy medicines

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Buy MEdicine";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int a=0;

int meri\_marzi;

if(addmedicineindex==0)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 2);

cout<<"We are sorry to say thet at present, no medicines in the store."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

else

{

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the name of medicines you want to purchase: ";

getline(cin, purchasemedicines);

if(checkenter(purchasemedicines)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the quantity of the medicine: ";

getline(cin, buymedicinequantity);

int buy=stoi(buymedicinequantity);

check\_integer(buymedicinequantity);

for(int i=0; i<addmedicineindex; i++)

{

if(purchasemedicines==medicinename[i])

{

a=i;

break;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

cout<<"Thank you for your purchase of the medication. We greatly value your choice and trust in our product."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

else

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The medication is presently inaccessible or not currently in stock."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

}

if(buy<stoi(medicinequantity[a]))

{

meri\_marzi=stoi(medicinequantity[a])-buy;

medicinequantity[a]=to\_string(meri\_marzi);

break;

}

else if(checkenter(buymedicinequantity)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(buymedicinequantity)==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

filehandlingbuymedicines(buymedicinequantity, purchasemedicines, addmedicineindex, medicinename, medicinequantity, medicineexpirydate, medicineprice, medicinebatchnumber, filenamebuymedicines);

break;

}

}

getch();

}

}

void patientreview(string review[], int &addpatientindex)

{

//this function will take the patient review

system("cls");

header();

while(true)

{

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Patient Review";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"How much you rate the hospitality (out of 5): ";

cin>>review[addpatientindex];

if(checkenter(review[addpatientindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(review[addpatientindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(review[addpatientindex]!="1" && review[addpatientindex]!="2" && review[addpatientindex]!="3" && review[addpatientindex]!="4" && review[addpatientindex]!="5")

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Sir, ratings can be provided exclusively on a scale of 5."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

filehandlingpatientreview(review, addpatientindex);

break;

}

}

if(review[addpatientindex]=="1" || review[addpatientindex]=="2")

{

reviewhearth();

}

else if(review[addpatientindex]=="3" || review[addpatientindex]=="4")

{

reviewhearth();

}

else

{

reviewhearth();

}

addpatientindex;

}

void addmanager(string managername, string managercnic)

{

while(true)

{

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add manager";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

//this function will add manager

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "Enter manager name: ";

getline(cin, managername);

if(checkenter(managername)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck(managername)==0)

{

//this is used to check if the user input is digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}while (true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout << "Enter manager CNIC: ";

getline(cin, managercnic);

string cnicCheck=managercnic;

if ((checkenter(managercnic)==0))

{

//this is used to check wheather the user input empty

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else if (cnicCheck.length() != 13)

{

//thisi is used to check the length of the number

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else if(charactermCheck(managercnic)==0)

{

//this is used to check wheather the user input character

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+6);

cout<<" "<<endl;

}

else

{

break;

}

}

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 2);

cout<<"The manager has been added successfully"<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

void changemanager(string managername)

{

while(true)

{

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Change manager";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

//this is used to change the manager

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout << "Enter new manager name: ";

getline(cin, managername);

if(checkenter(managername)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck(managername)==0)

{

//this is used to check if the user input is digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

}

}

string doctormenu()

{

//this function will show you doctormenu

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Doctor Menu";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

string doctoroption;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Select one of the following option..."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"1. View Patient."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"2. View Doctors List."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"3. View availble beds."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout << "Note: Enter your option from the above menu otherwise you will be shifted back to login page.";

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"Enter your option...";

cin>>doctoroption;

return doctoroption;

}

string pharmacymenu()

{

//this function will show you pharmacy menu

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Pharmacist Menu";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

string parmacyoption;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Select one of the following option..."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"1. Add Medicine Form."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"2. View Medicine."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"3. Update Medicine Form."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"4. Expired Medicine."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"5. Mostly Used Medicines(In Demand)."<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout<<"6. Remove Medicine From Stock."<<endl;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 1);

cout<<"7. Add Employ."<<endl;

gotoxy(x, y+8);

SetConsoleTextAttribute(h, 1);

cout<<"8. Remove Employ."<<endl;

gotoxy(x, y+9);

SetConsoleTextAttribute(h, 1);

cout<<"9. View Employ."<<endl;

gotoxy(x, y+10);

SetConsoleTextAttribute(h, 4);

cout << "Note: Enter your option from the above menu otherwise you will be shifted back to login page.";

gotoxy(x, y+11);

SetConsoleTextAttribute(h, 1);

cout<<"Enter your option...";

cin>>parmacyoption;

return parmacyoption;

}

void addmedicineform(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine)

{

//this function will show you medicine form

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Medicine";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the name of the medicine: ";

getline(cin, medicinename[addmedicineindex]);

if(checkenter(medicinename[addmedicineindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck(medicinename[addmedicineindex])==0)

{

//this is used to check if the user input is digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the quantity of the medicine: ";

getline(cin, medicinequantity[addmedicineindex]);

if(checkenter(medicinequantity[addmedicineindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(medicinequantity[addmedicineindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Date of Expiry(dd/mm/yyyy): ";

getline(cin, medicineexpirydate[addmedicineindex]);

string check=medicineexpirydate[addmedicineindex];

if(checkenter(medicineexpirydate[addmedicineindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(medicineexpirydate[addmedicineindex])==0 && check[2]!='/' && check[5]!='/')

{

//this is used to check if the user input is alphabets

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the prize: ";

getline(cin, medicineprice[addmedicineindex]);

if(checkenter(medicineprice[addmedicineindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(charactermCheck(medicineprice[addmedicineindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"Batch Number (put 10 characters): ";

getline(cin, medicinebatchnumber[addmedicineindex]);

string medicinebatchnumbercheck=medicinebatchnumber[addmedicineindex];

if(checkenter(medicinebatchnumber[addmedicineindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else if(medicinebatchnumbercheck.length()<10)

{

//this is used to check if the user input lenght

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly provide a numeric entry consisting of at least ten digits."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else if(charactermCheck(medicinebatchnumber[addmedicineindex])==0)

{

//this is used to check if the user input is alphabest

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else

{

addmedicineindex++;

filehandlingaddmedicine(medicinename, medicinequantity, medicineexpirydate, medicineprice, medicinebatchnumber, addmedicineindex, filenameaddmedicine);

break;

}

}

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 2);

cout<<"The medicines has been added successfully."<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key continue...";

getch();

}

void viewmedicineform(string buymedicinequantity, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenameaddmedicine)

{

//this function will show the medicine form

system("cls");

if(addmedicineindex==0)

{

SetConsoleTextAttribute(h, 2);

cout<<"No medicines are present."<<endl;

}

else

{

gotoxy(x1, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Medicine Name";

gotoxy(x1+25, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Quantity";

gotoxy(x1+35, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Expiry Date";

gotoxy(x1+52, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Price";

gotoxy(x1+60, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Batch Number";

}

for(int c=0; c<addmedicineindex; c++)

{

gotoxy(x1, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<medicinename[c];

gotoxy(x1+25, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<(medicinequantity[c]);

gotoxy(x1+35, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<medicineexpirydate[c];

gotoxy(x1+52, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<medicineprice[c];

gotoxy(x1+60, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<medicinebatchnumber[c];

}

gotoxy(x, y+20);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void updatemedicineform(string updatemedicinebatchnumber, string medicinebatchnumber[], int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string filenameaddmedicine)

{

//this function will update medicines form

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Update Medicine";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int c=0;

int a=-1;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Batch Number (put 10 characters): ";

getline(cin, updatemedicinebatchnumber);

if(checkenter(updatemedicinebatchnumber)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(updatemedicinebatchnumber.length()<10)

{

//this is used to check if the user input is not according to th egiven format

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly provide a numeric entry consisting of at least ten digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(updatemedicinebatchnumber)==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

for(int i=0; i<addmedicineindex; i++)

{

if(medicinebatchnumber[i]==updatemedicinebatchnumber)

{

a=0;

c=i;

break;

}

}

if(a==0)

{

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the name of the medicine: ";

getline(cin, medicinename[c]);

if(checkenter(medicinename[c])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the quantity of the medicine: ";

getline(cin, medicinequantity[c]);

if(checkenter(medicinequantity[c])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(charactermCheck(medicinequantity[c])==0)

{

//this is used to check if the user input is digits

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only alphabets."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"Date of Expiry(dd/mm/yyyy): ";

getline(cin, medicineexpirydate[c]);

string check=medicineexpirydate[c];

if(checkenter(medicineexpirydate[c])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else if(charactermCheck(medicineexpirydate[c])==0 && check[2]!='/' && check[5]!='/')

{

//this is used to check if the user input is alphabets

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+4);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the prize: ";

getline(cin, medicineprice[c]);

if(checkenter(medicineprice[c])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else if(charactermCheck(medicineprice[c])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"PLease make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else

{

addmedicineindex++;

filehandlingaddmedicine(medicinename, medicinequantity, medicineexpirydate, medicineprice, medicinebatchnumber, addmedicineindex, filenameaddmedicine);

break;

}

}

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 2);

cout<<"The medicine has been updated."<<endl;

gotoxy(x, y+7);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

else if(a==-1)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"The medicine you want to update/exchange is not available in the store. Please make sure that you have view medicines."<<endl;

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+1);

cout<<"Press any key to continue...";

getch();

}

}

void medicineindemand()

{

//this function will show you th emedicines in demand

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Demanding Medicines";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

gotoxy(x, y);

SetConsoleTextAttribute(h, 2);

cout<<"For headache, usually the panadol is taken."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

cout<<"For fever, usually the ibuprofen and asparin is taken."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

cout<<"For infections, usually the Amoxicillin is taken."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 2);

cout<<"For hypothyroidism, usually the Levothyroxine is taken."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 2);

cout<<"For allergy, Cetirizine (Zyrtec) is taken."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 2);

cout<<"For reducing inflammation, usually the Prednisone is taken"<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"All these medicines are taken as prescribed by the physicians."<<endl;

}

void addemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[])

{

//tis function will add employ in the system

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Employ";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter Employ name: ";

getline(cin, employname[addemployindex]);

if(checkenter(employname[addemployindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

if(digitscheck(employname[addemployindex])==0)

{

//this is used to check if the user input is digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter employ CNIC: ";

getline(cin, employcnic[addemployindex]);

string employcniccheck=employcnic[addemployindex];

if ((checkenter(employcnic[addemployindex])==0))

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if (employcniccheck.length() != 13)

{

//this is used to check if the user input length

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(employcnic[addemployindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(recursionCheck(employcnic[addemployindex], employcnic, addemployindex)==0)

{

//this is used to check if the user input is already entered

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"The CNIC (Computerized National Identity Card) information has been successfully entered previously."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter Employ's phone number: ";

getline(cin, employphonenumber[addemployindex]);

string employphonenumbercheck=employphonenumber[addemployindex];

if(checkenter(employphonenumber[addemployindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if (employphonenumbercheck.length() != 11)

{

//this is used to check if the user input length

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(employphonenumber[addemployindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(recursionCheck(employphonenumbercheck, employphonenumber, addemployindex)==0)

{

//this is used to check if the user input is alredy entered

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The contact number has been successfully entered previously."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout << "Enter Employ's Gender(in small letters): ";

getline(cin, employgender[addemployindex]);

if(checkenter(employgender[addemployindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(gendercheck(employgender[addemployindex])==0)

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Wrong ones. Such type of gender doesn't exist."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

filehandlingaddemploy(employname, addemployindex, employcnic, employphonenumber, employgender);

break;

}

}

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 2);

cout<<"The employ has been added successfully."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

addemployindex++;

}

void expiredmedicine(int addmedicineindex, string currentdate[], string medicineexpirydate[], string medicinename[])

{

//this function will show you the expired medicines

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

string input;

int check = 0;

int expiredcount = 0;

bool expiredmedicine;

string coutexpiredmedicine[1000];

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Expired Medicine";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

if(addmedicineindex==0)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 2);

cout<<"No medicine is present."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue..."<<endl;

getch();

}

else

{

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Kindly provide the current date (dd/mm/yyyy): ";

cin>>input;

if(checkenter(input)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(input)==0 && input[2]!='/' && input[5]!='/')

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

for (int i=0; i<addmedicineindex; i++)

{

medicineexpirydate[i];

expiredmedicine=isExpired(input, medicineexpirydate[i]);

if (expiredmedicine == true)

{

check=1;

coutexpiredmedicine[expiredcount] = medicinename[i];

expiredcount++;

}

}

if (check == 1)

{

for (int i=0; i<expiredcount; i++)

{

gotoxy(x, y+3+i);

cout<<coutexpiredmedicine[i]<<endl;

}

}

else

{

SetConsoleTextAttribute(h, 2);

cout << "No medicine is expired" << endl;

}

}

}

int convertDatetodays(string date)

{

string dateDays="0";

string dateMonths="0";

string dateYears="0";

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

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

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

return stoi(dateDays) + stoi(dateMonths) \* 30 + (stoi(dateYears) - 2000) \* 365;

}

bool isExpired(string currentDate, string expiryDate)

{

//this function will return false if the medicines is not expired

int currentDays = convertDatetodays(currentDate);

int expiryDays = convertDatetodays(expiryDate);

if(currentDays>expiryDays)

{

return 1;

}

else{

return 0;

}

}

void removemedicine(string updatemedicinebatchnumber, int &addmedicineindex, string medicinebatchnumber[], string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string filenameaddmedicine)

{

//thius is used to remove the medicines

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Exchange Medicine";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int c=0;

int a=-1;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Batch Number of the medicine you want to remove (put 10 characters): ";

getline(cin, updatemedicinebatchnumber);

if(checkenter(updatemedicinebatchnumber)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(updatemedicinebatchnumber.length()<10)

{

//this is used to check if the user input length

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly provide a numeric entry consisting of atleast ten digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(updatemedicinebatchnumber)==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Plesae make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

for (int x = 0; medicinebatchnumber[x] != ""; x++)

{

if (updatemedicinebatchnumber==medicinebatchnumber[x])

{

a=0;

for (int i = x; medicinebatchnumber[i] != ""; i++)

{

medicinename[i]=medicinename[i + 1];

medicinequantity[i]=medicinequantity[i + 1];

medicineexpirydate[i] =medicineexpirydate[i + 1];

medicineprice[i]=medicineprice[i + 1];

medicinebatchnumber[i]=medicinebatchnumber[i+1];

}

addmedicineindex--;

break;

}

}

if(a==0)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

cout<<"The medicine associated with this specific batch number has been removed from records."<<endl;

filehandlingaddmedicine(medicinename, medicinequantity, medicineexpirydate, medicineprice, medicinebatchnumber, addmedicineindex, filenameaddmedicine);

getch();

}

else if(a==-1)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"This medicine is not available.";

getch();

}

gotoxy(60, 22);

SetConsoleTextAttribute(h, 4);

cout << "Press any key to continue...";

getch();

}

void removeemploy(int &addemployindex, string removeemploycnic, string employcnic[], string employname[], string employphonenumber[], string employgender[])

{

//this function will remove employ from the list

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Remove Employ";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int c=0;

int a=-1;

if(addemployindex==0)

{

//if the employ is not entered

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

cout<<"To change employ to first have to enter employ."<<endl;

}

else

{

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the cnic of the employ you want to remove: ";

getline(cin, removeemploycnic);

if ((checkenter(removeemploycnic)==0))

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"The medicine associated with this specific batch number has been removed from records."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if (removeemploycnic.length() != 13)

{

//this is used to check if the user input length

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout << "The length of the provided number appears to be incorrect." << endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(removeemploycnic)==0)

{

//this is used to check if the user input is not according to the given format

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

for (int x = 0; employcnic[x] != ""; x++)

{

if (removeemploycnic==employcnic[x])

{

a=0;

for (int i = x; employcnic[i] != ""; i++)

{

employname[i] = employname[i + 1];

employcnic[i] = employcnic[i + 1];

employphonenumber[i] = employphonenumber[i + 1];

employgender[i] = employgender[i + 1];

}

addemployindex--;

break;

}

}

if(a==0)

{

SetConsoleTextAttribute(h, 2);

gotoxy(x, y+3);

filehandlingaddemploy(employname, addemployindex, employcnic, employphonenumber, employgender);

cout<<"The individual linked to this CNIC has been subject to separation from employment."<<endl;

gotoxy(60, 22);

SetConsoleTextAttribute(h, 4);

cout << "Press any key to continue...";

getch();

}

else if(a==-1)

{

SetConsoleTextAttribute(h, 4);

gotoxy(x, y+5);

cout<<"The employ with this CNIC is not added."<<endl;

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

}

}

void viewemploy(int &addemployindex, string employname[], string employcnic[], string employphonenumber[], string employgender[])

{

//this function will view the employ that was added in the system

system("cls");

if(addemployindex==0)

{

//if the employ is not added

SetConsoleTextAttribute(h, 2);

cout<<"No employee in record."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

else

{

gotoxy(x1, y\_1);

SetConsoleTextAttribute(h, 1);

cout<<"Employee Name";

gotoxy(x1+30, y\_1);

SetConsoleTextAttribute(h, 1);

cout<<"Employee CNIC";

gotoxy(x1+45, y\_1);

SetConsoleTextAttribute(h, 1);

cout<<"Employee Phone no.";

gotoxy(x1+65, y\_1);

SetConsoleTextAttribute(h, 1);

cout<<"Employee's Gender";

filehandlingaddemploy(employname, addemployindex, employcnic, employphonenumber, employgender);

}

for(int c=0; c<addemployindex; c++)

{

gotoxy(x1, y\_1+c+2);

SetConsoleTextAttribute(h, 2);

cout<<employname[c];

gotoxy(x1+30, y\_1+c+2);

SetConsoleTextAttribute(h, 2);

cout<<employcnic[c];

gotoxy(x1+45, y\_1+c+2);

SetConsoleTextAttribute(h, 2);

cout<<employphonenumber[c];

gotoxy(x1+65, y\_1+c+2);

SetConsoleTextAttribute(h, 2);

cout<<employgender[c];

gotoxy(x, y+20);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

}

string equipmentmenu()

{

//this function will use to see the equipment manager

system("cls");

header();

string equipmentoption;

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Select one of the following option..."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"1. Add Equipments Form."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"2. View Equipments."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"3. Remove Equipments Form."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 1);

cout<<"4. Exchnage Equipments. "<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout << "Note: Enter your option from the above menu otherwise you will be shifted back to login page.";

gotoxy(x, y+6);

SetConsoleTextAttribute(h, 1);

cout<<"Enter your option...";

cin>>equipmentoption;

return equipmentoption;

}

void addequipmentform(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment)

{

//this function will add equipments form

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Add Equipments";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Enter name of the Equipment: ";

getline(cin, equipmentname[addequipmentindex]);

if(checkenter(equipmentname[addequipmentindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(digitscheck(equipmentname[addequipmentindex])==0)

{

//this is used to check if the user input is digits

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the quantity of the Equipment: ";

getline(cin, equipmentquantity[addequipmentindex]);

if(checkenter(equipmentquantity[addequipmentindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentquantity[addequipmentindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the prize: ";

getline(cin, equipmentprize[addequipmentindex]);

if(checkenter(equipmentprize[addequipmentindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentprize[addequipmentindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 1);

cout<<"Traceability Number(put 10 characters): ";

getline(cin, equipmentbatchnumber[addequipmentindex]);

string equipmentbatchnumberCheck=equipmentbatchnumber[addequipmentindex];

if(checkenter(equipmentbatchnumber[addequipmentindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(equipmentbatchnumberCheck.length()<10)

{

//this is used to check if the user input lenght

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly provide a numeric entry consisting of at least ten digits."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(recursionCheck(equipmentbatchnumberCheck,equipmentbatchnumber, addequipmentindex)==0)

{

//this is used to check wheather the same number is input already

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"The contact number has been already entered. Try another one."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentbatchnumber[addequipmentindex])==0)

{

//this is used to check if the user input is alphabest

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y+3);

cout<<" "<<endl;

}

else

{

addequipmentindex++;

filehandlingaddequipments(equipmentname, addequipmentindex, equipmentquantity, equipmentprize, equipmentbatchnumber, filenameaddequipment);

break;

}

}

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 2);

cout<<"The equipment has been added to the store."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

void viewequipmentform(int &addequipmentindex, string equipmentname[], string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[])

{

//this function will show you the list of equipment added

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

if(addequipmentindex==0)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 2);

cout<<"There is no equipment added at that time."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

else

{

gotoxy(x1, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Equipments Name";

gotoxy(x1+30, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Equipments Quantity";

gotoxy(x1+50, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Equipments Prize";

gotoxy(x1+70, y\_1-2);

SetConsoleTextAttribute(h, 1);

cout<<"Equuipments Batch number";

for(int c=0; c<addequipmentindex; c++)

{

gotoxy(x1, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<equipmentname[c];

gotoxy(x1+30, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<equipmentquantity[c];

gotoxy(x1+50, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<equipmentprize[c];

gotoxy(x1+70, y\_1+c);

SetConsoleTextAttribute(h, 2);

cout<<equipmentbatchnumber[c];

}

}

gotoxy(x, y+20);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

void removeequipmentform(int &addequipmentindex, string updateequipmentbatchnumber, string equipmentbatchnumber[], string equipmentname[], string equipmentprize[], string equipmentquantity[], string filenameaddequipment)

{

//thius is used to remove the medicines

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Remove Equipments";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int c=0;

int a=-1;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Traceability Number of the equipment you want to remove (put 10 characters): ";

getline(cin, updateequipmentbatchnumber);

if(checkenter(updateequipmentbatchnumber)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(updateequipmentbatchnumber.length()<10)

{

//this is used to check if the user input length

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly provide a numeric entry consisting of atleast ten digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(updateequipmentbatchnumber)==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Plesae make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

for (int x = 0; equipmentbatchnumber[x] != ""; x++)

{

if (updateequipmentbatchnumber== equipmentbatchnumber[x])

{

a=0;

for (int i = x; equipmentbatchnumber[i] != ""; i++)

{

equipmentname[i] = equipmentname[i + 1];

equipmentprize[i] = equipmentprize[i + 1];

equipmentbatchnumber[i] = equipmentbatchnumber[i + 1];

equipmentquantity[i] = equipmentquantity[i + 1];

}

addequipmentindex--;

break;

}

}

if(a==0)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 2);

filehandlingaddequipments(equipmentname, addequipmentindex, equipmentquantity, equipmentprize, equipmentbatchnumber, filenameaddequipment);

cout<<"The medicine associated with this specific batch number has been removed from records."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout << "Press any key to continue...";

getch();

}

else if(a==-1)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The Tracebility number of the equipment were not added."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

}

}

void updateequipment(string updateequipmentbatchnumber, int &addequipmentindex, string equipmentbatchnumber[], string equipmentname[], string equipmentquantity[], string equipmentprize[], string filenameaddequipment)

{

//this function will update medicines form

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

system("cls");

header();

gotoxy(x, y-1);

SetConsoleTextAttribute(h, 6);

cout<<" \*\*\*\*\*\*";

SetConsoleTextAttribute(h, 4);

cout<<"Exchange Equipments";

SetConsoleTextAttribute(h, 6);

cout<<"\*\*\*\*\*\*"<<endl;

int c=0;

int a=-1;

while(true)

{

gotoxy(x, y);

SetConsoleTextAttribute(h, 1);

cout<<"Traceability Number(put 10 characters): ";

getline(cin, updateequipmentbatchnumber);

if(checkenter(updateequipmentbatchnumber)==0)

{

//this is used to check if the user input is empty

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(updateequipmentbatchnumber.length()<10)

{

//this is used to check if the user input is not according to th egiven format

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Kindly provide a numeric entry consisting of at least ten digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else if(charactermCheck(updateequipmentbatchnumber)==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have enter only digits."<<endl;

getch();

gotoxy(x, y);

cout<<" "<<endl;

}

else

{

break;

}

}

for(int i=0; i<addequipmentindex; i++)

{

if(equipmentbatchnumber[i]==updateequipmentbatchnumber)

{

a=0;

c=i;

break;

}

}

if(a==0)

{

while(true)

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the new name of the equipment: ";

getline(cin, equipmentname[c]);

if(checkenter(equipmentname[addequipmentindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else if(digitscheck(equipmentname[addequipmentindex])==0)

{

//this is used to check if the user input is digits

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only alphabets."<<endl;

getch();

gotoxy(x, y+1);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the new quantity of the equipments: ";

getline(cin, equipmentquantity[c]);

if(checkenter(equipmentquantity[addequipmentindex])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentquantity[addequipmentindex])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Please make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+2);

cout<<" "<<endl;

}

else

{

break;

}

}

while(true)

{

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 1);

cout<<"Enter the new prize: ";

getline(cin, equipmentprize[c]);

if(checkenter(equipmentprize[c])==0)

{

//this is used to check if the user input is empty

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"Please provide valid input; empty entries are not accepted. Please try again."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else if(charactermCheck(equipmentprize[c])==0)

{

//this is used to check if the user input is alphabets

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"PLease make sure that you have entered only digits."<<endl;

getch();

gotoxy(x, y+5);

cout<<" "<<endl;

}

else

{

filehandlingaddequipments(equipmentname, addequipmentindex, equipmentquantity, equipmentprize, equipmentbatchnumber, filenameaddequipment);

break;

}

}

}

else if(a==-1)

{

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The tacebility number of teh equipment is not added. Please make sure that you have entered the accurate number."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

}

void thankyou()

{

system("cls");

cout<<""<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" =======";

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" ===========";

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" ===============";

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" ++++++=============";

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@ @@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" ++++=+=-:--:====";

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@ @@@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" --==-===-:=-:-===";

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@@@@ @@@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" :::======---===";

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@@@ @@@@ @@@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" :::: --===========";

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@@ @@@ @@@@ @@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" --:: ++====++";

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ @ @@@@ @@@@ @@@@ @@@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" ::: =======+";

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@@ @@ @@@ @@@@@@ @@@@ @@@@ @@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" ::-- +===+";

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@@ @@@ @@ @@@@@@@ @@@@@ @@@ @@@@@ @@@@@@@ "<<endl;

SetConsoleTextAttribute(h, 11);

cout<<" :-";

SetConsoleTextAttribute(h, 2);

cout<<" @";

SetConsoleTextAttribute(h, 11);

cout<<" \*+ ";

SetConsoleTextAttribute(h, 2);

cout<<"@@@ @@@@ @@@@@ @@@@ @@@ @@ @@@@ @@@ @@ @@@@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ @@@@ @@@@@@@ @@@ @@ @@@ @@@@ @@@ @@ @@@@ @@@@@@@@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@@ @@@@ @@@@@@ @@@ @@@ @@@@@ @@@@@ @@@@ @@@@ @@@@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@ @@@@ @@@ @@@@@ @@ @@@@@ @@@ @@@ @@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@@@@ @@@ @@@@@@@ @@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@@@@ @@@@ @@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@@ @@@ @@@ @@ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@ @@@ @@@ ";

SetConsoleTextAttribute(h, 11);

cout<<"\*++ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@@ @@ @@@ @@@ @@@@@@@ ";

SetConsoleTextAttribute(h, 11);

cout<<" +===+==+ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@ @@@ @@ @@@@ @@@@@@ @ ";

SetConsoleTextAttribute(h, 11);

cout<<"==--====+--=+======+ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@ @@ @@ @@@ @@ @@@ @@@@ @@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<"+===--=-===--==--====:-+:: : "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @ @@@ @@@ @@@ @ @@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<" ==---======----==-=----=---: "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ @@@@ @@ @@@@@@ ";

SetConsoleTextAttribute(h, 11);

cout<<"===+====:.:=---=-:---: "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@@@@@@@@@ @ @@@ ";

SetConsoleTextAttribute(h, 11);

cout<<"=======---:---:::: "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<":-:---- \*\*++++++=+-::.:-=-:- "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@ @@@@ ";

SetConsoleTextAttribute(h, 11);

cout<<"::- -::-::-::- \*+++===++===:::::-..- "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@@@@ ";

SetConsoleTextAttribute(h, 11);

cout<<":::::::-====---- +++++++++=--==-:--:-:: "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<"::-::==+++=----:----=++ ===--=-:::-=-= "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<"+ -::::::----=---------:- =---== + "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<"++ +++ :--:----:-:-:------: +=--= "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<"+=++===== -:-:-----:--:: ===+ "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<" ++ + ::::: + "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@ @@ ";

SetConsoleTextAttribute(h, 11);

cout<<"-- "<<endl;

SetConsoleTextAttribute(h, 2);

cout<<" @@@@@ "<<endl;

SetConsoleTextAttribute(h, 13);

cout<<"Made By:";

SetConsoleTextAttribute(h, 15);

cout<<" MUHAMMAD\_OWAIS\_KHUBAISI..."<<endl;

SetConsoleTextAttribute(h, 4);

cout<<"Press any key to continue...";

getch();

}

void prescription()

{

while(true)

{

string prescription=viewprescription();

//for different diseases in the menu

//all this data is taken from google and is copy right

if(prescription=="1")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Medications used to treat diabetes include: "<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<" 'Metformin', 'Insulin releasing pills', 'Starch blockers', 'Amylin analogs'. "<<endl;

getch();

}

else if(prescription=="2")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Medications used to treat diabetes include: "<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<" 'Angiotensin-converting enzyme (ACE)' "<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"inhibitors reduce blood pressure by relaxing your blood vessels."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<" Common examples are enalapril, lisinopril, perindopril and ramipril. The most common side effect is a persistent dry cough."<<endl;

getch();

}

else if(prescription=="3")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Some inhaled asthma medication combinations contain both a corticosteroid and a bronchodilator:"<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<" 'Fluticasone and salmeterol' (Advair Diskus, AirDuo Digihaler, others)"<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<" 'Budesonide and formoterol' (Symbicort)"<<endl;

getch();

}

else if(prescription=="4")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Some commonly prescribed include: "<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Statins: Atorvastatin (Lipitor), Fluvastatin (Lescol), Lovastatin (Mevacor), Pitavastatin (Livalo), Pravastatin (Pravachol), Rosuvastatin (Crestor), Simvastatin (Zocor)"<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Nicotinic acids: Niacin."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Cholesterol absorption inhibitor: Ezetimibe (Zetia)."<<endl;

getch();

}

else if(prescription=="5")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Aspirin and other antiplatelets."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Most people will be given aspirin straight after having an ischaemic stroke."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"As well as being a painkiller, aspirin is an antiplatelet, which reduces the chances of another clot forming."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Other antiplatelet medicines may be used later, such as clopidogrel and dipyridamole."<<endl;

getch();

}

else if(prescription=="6")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"To treat flu, oseltamivir or inhaled zanamivir are usually prescribed for five days."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Oseltamivir treatment is given to hospitalized patients, and some patients might be treated for more than five days."<<endl;

getch();

}

else if(prescription=="7")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Donepezil, galantamine and rivastigmine can be prescribed for people with early- to mid-stage Alzheimer's disease."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"The latest guidelines recommend that these medicines should be continued in the later, severe, stages of the disease."<<endl;

getch();

}

else if(prescription=="8")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"The treatment for HIV is called antiretroviral therapy (ART)."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"ART involves taking a combination of HIV medicines (called an HIV treatment regimen) every day."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"ART is recommended for everyone who has HIV. ART cannot cure HIV, but HIV medicines help people with HIV live longer, healthier lives."<<endl;

getch();

}

else if(prescription=="9")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"These are the most common long-term control medications for asthma."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"These anti-inflammatory drugs include: "<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"fluticasone (Flovent HFA), budesonide (Pulmicort Flexhaler), beclomethasone (Qvar RediHaler), ciclesonide (Alvesco, Omnaris) and mometasone (Asmanex HFA)."<<endl;

getch();

}

else if(prescription=="10")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Acetazolamide."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Brivaracetam available as Briviact."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Cannabidiol available as Epidyolex."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Carbamazepine also available as Tegretol, Tegretol Prolonged Release."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"Doctors at a specialist epilepsy centre should review your epilepsy and treatment to try to get you better seizure control."<<endl;

gotoxy(x, y+5);

SetConsoleTextAttribute(h, 4);

cout<<"They might also talk to you about other treatments for epilepsy."<<endl;

getch();

}

else if(prescription=="11")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"For adults with chronic kidney disease (CKD)."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"The main treatments are:"<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"lifestyle changes - to help you stay as healthy as possible."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"medicine - to control associated problems, such as high blood pressure and high cholesterol."<<endl;

gotoxy(x, y+4);

SetConsoleTextAttribute(h, 4);

cout<<"dialysis - treatment to replicate some of the kidney's functions, which may be necessary in advanced (stage 5) CKD."<<endl;

getch();

}

else if(prescription=="12")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"'Ferrous sulfate' is the mainstay treatment for treating patients with iron deficiency anemia."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"They should be continued for about 2 months after correction of the anemia and its etiologic cause in order to replenish body stores of iron."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Ferrous sulfate is the most common and cheapest form of iron utilized."<<endl;

getch();

}

else if(prescription=="13")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Your doctor may recommend one of these medicines that you can't get over the counter:"<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Allopurinol (Aloprim, Zyloprim) reduces uric acid production."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Colchicine (Colcrys, Gloperba, Mitigare) reduces inflammation."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"Febuxostat (Uloric) reduces uric acid production."<<endl;

getch();

}

else if(prescription=="14")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Treatment for chronic hepatitis B may include:"<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Antiviral medications."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"Several antiviral medicines — including entecavir (Baraclude), tenofovir (Viread), lamivudine (Epivir), adefovir (Hepsera) and telbivudine — can help fight the virus and slow its ability to damage your liver."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"These drugs are taken by mouth."<<endl;

getch();

}

else if(prescription=="15")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"If you have an active TB disease you will probably be treated with a combination of antibacterial medications for a period of six to 12 months. "<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"The most common treatment for active TB is isoniazid INH in combination with three other drugs—rifampin, pyrazinamide and ethambutol."<<endl;

getch();

}

else if(prescription=="16")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Loperamide."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Loperamide, which people commonly refer to as its brand name, Imodium, is an anti-diarrheal agent that treats diarrhea by affecting the opiate receptors in the intestine. "<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"This drug works by slowing the contractions of the intestines and how quickly its contents pass through."<<endl;

getch();

}

else if(prescription=="17")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"In otherwise uncomplicated pneumonia, azithromycin is the initial drug of choice, as it covers most of the potential etiologic agents, including Mycoplasma species. "<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Compared with other drugs, this agent also causes less GI upset, and it has the potential for good compliance because of its reduced dosing frequency."<<endl;

getch();

}

else if(prescription=="18")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"An underactive thyroid is treated with a thyroid hormone replacement."<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"The drug levothyroxine (Levoxyl, Thyquidity, others) replaces T-4 and results in the pituitary gland releasing less TSH."<<endl;

gotoxy(x, y+2);

SetConsoleTextAttribute(h, 4);

cout<<"The drug liothyronine (Cytomel) may be prescribed as a T-3 replacement."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<"These treatments may decrease the size of the goiter."<<endl;

getch();

}

else if(prescription=="19")

{

system("cls");

header();

gotoxy(x, y);

SetConsoleTextAttribute(h, 4);

cout<<"Appendicitis Medication: "<<endl;

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 4);

cout<<"Penicillins, Cephalosporins, Aminoglycosides, Carbapenems, Fluoroquinolones, Anti-infective Agents, Analgesics.";

getch();

}

else

{

break;

}

}

}

void reviewhearth()

{

gotoxy(x, y+1);

SetConsoleTextAttribute(h, 2);

cout<<"Thank you. Next time will provide you better service."<<endl;

gotoxy(x, y+3);

SetConsoleTextAttribute(h, 4);

cout<<" =\*\*+=--=+\*\*= -+\*+====+\*\*=: "<<endl;

gotoxy(x, y+4);

cout<<" :++: +\*- ++ =\*: =: "<<endl;

gotoxy(x, y+5);

cout<<" -+: =\*: == +\* :+: "<<endl;

gotoxy(x, y+6);

cout<<" =+. -\* \*- =\* -\*= "<<endl;

gotoxy(x, y+7);

cout<<" -\*. -\*+\*- \*+ -\*\* "<<endl;

gotoxy(x, y+8);

cout<<" ++. -\*- :\* =+\*: "<<endl;

gotoxy(x, y+9);

cout<<" \*=. .+ .- :+-+- "<<endl;

gotoxy(x, y+10);

cout<<" =+. =\* =\* :\*-=\* "<<endl;

gotoxy(x, y+11);

cout<<" :\*: \*= +# -+ \* "<<endl;

gotoxy(x, y+12);

cout<<" ++. =\* :\*\*: -= \*- "<<endl;

gotoxy(x, y+13);

cout<<" +=. -\* =+\*+ += +\* "<<endl;

gotoxy(x, y+14);

cout<<" :+=. -\*+============\*--# \*- :\*===================- "<<endl;

gotoxy(x, y+15);

cout<<" :+=. \* +: "<<endl;

gotoxy(x, y+16);

cout<<" =+. \*= + "<<endl;

gotoxy(x, y+17);

cout<<" :\*=. =\*=+ "<<endl;

gotoxy(x, y+18);

cout<<" =\*-. .\*+ \*\*=+ " <<endl;

gotoxy(x, y+19);

cout<<" ++:. .+\* \*\*: "<<endl;

gotoxy(x, y+20);

cout<<" :++:. :+\*: \*+: " <<endl;

gotoxy(x, y+21);

cout<<" +\*:. :++: :+: " <<endl;

gotoxy(x, y+22);

cout<<" =\*=. -\*= =: " <<endl;

gotoxy(x, y+23);

cout<<" :++- :++- : "<<endl;

gotoxy(x, y+24);

cout<<" =\*= =\*=: "<<endl;

gotoxy(x, y+25);

cout<<" :-: "<<endl;

gotoxy(x, y+26);

cout<<"Press any key to continue...";

getch();

}

void view()

{

SetConsoleTextAttribute(h, 2);

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<" %%% ";

SetConsoleTextAttribute(h, 4);

cout<<"PYSANTAS\_MEDI\_HARMONY ";

SetConsoleTextAttribute(h, 2);

cout<<"%%% "<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<" %%% %%% "<<endl;

cout<<" %%% %%% "<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%%% %%%%%% %%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%%% %%%%%% %%%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%% "<<endl;

cout<<" %%% %%% %%%%%%%%%%%%%%% %%% %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%% %%% %%%%%%%%%%%%%%% %%% %%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% % % %%%%%%%%%%% %%%%%% %%%%%%%%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %% %%%%%% %% %%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %% %% %%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %% %% %%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %% %% %%%% % % %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%% %%%%%%%%%%% %%%%%%%%%% %%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% %%%%%%%%%%% %%%%%%%%%% %%% "<<endl;

cout<<" %%% %%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%% %%%%% %%%%% %%%%% %%%% %%%%% %%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% % % %%%%% %%%%% %%%%% %%%% %%%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %%%%% %%%%% %%%% %%%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %%%%% %%%%% %%%% %%%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %%%%% %%%%% %%%% %%%%% % % %%% "<<endl;

cout<<" %%% % % %%%%% %%%%% %%%%% %%%% %%%%% % % %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%% %%% %%% %%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% % % %%% -------- %%% % % %%% "<<endl;

cout<<" %%% % % %%% %%%%%%%% %%% % % %%% "<<endl;

cout<<" %%% % % %%% %%%%%%%%%%%%%% %%% % % %%% "<<endl;

cout<<" %%% % % %%% %% %% %%% % % %%% "<<endl;

cout<<" %%% % % %%% %% %% %%% % % %%% "<<endl;

cout<<" %%% %%%%%%%%%%%%% %%% %% %% %%% %%%%%%%%%%%%% %%% "<<endl;

cout<<" %%% %%% %% %% %%% %%% "<<endl;

cout<<" %%% %%% %% %% %%% %%% "<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"<<endl;

cout<<" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"<<endl;

SetConsoleTextAttribute(h, 3);

cout<<"Made By: ";

SetConsoleTextAttribute(h, 1);

cout<<"MUHAMMAD\_OWAIS\_KHUBAISI"<<endl;

}

string getField(string record, int field)

{

int commaCount = 1;

string item;

for (int x = 0; x < record.length(); x++)

{

if (record[x] == ',')

{

commaCount = commaCount + 1;

}

else if (commaCount == field)

{

item = item + record[x];

}

}

return item;

}

void filehandlingsignup(int &size, string firstname[], string secondname[], string gender[], string city[], string nationality[], string contactnumber[], string cnic[], string username[], string userpassword[], string userrole[], string filename)

{

fstream signupm;

signupm.open(filename, ios::out);

for(int i=0; firstname[i]!="\0"; i++)

{

signupm<<firstname[i]<<","<<secondname[i]<<","<<gender[i]<<","<<city[i]<<","<<nationality[i]<<","<<contactnumber[i]<<","<<cnic[i]<<","<<username[i]<<","<<userpassword[i]<<","<<userrole[i]<<endl;

}

signupm.close();

}

void readsignup(int &size, int &admin, string firstname[], string secondname[], string gender[], string city[], string nationality[], string contactnumber[], string cnic[], string username[], string userpassword[], string userrole[], string filename)

{

size=0;

string record;

fstream data;

data.open(filename, ios::in);

while(!(data.eof()))

{

getline(data, record);

if(record=="")

{

break;

}

firstname[size]=getField(record, 1);

secondname[size]=getField(record, 2);

gender[size]=getField(record, 3);

city[size]=getField(record, 4);

nationality[size]=getField(record, 5);

contactnumber[size]=getField(record, 6);

cnic[size]=getField(record, 7);

username[size]=getField(record, 8);

userpassword[size]=getField(record, 9);

userrole[size]=getField(record, 10);

if(userrole[size]=="admin")

{

admin=1;

}

size++;

}

data.close();

}

void filehandlingaddpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient)

{

fstream addpatientm;

addpatientm.open(filenameaddpatient, ios::out);

for(int i=0; i<addpatientindex; i++)

{

addpatientm<<patientname[i]<<","<<patientage[i]<<","<<patientcnic[i]<<","<<patientbloodgroup[i]<<","<<patienthistory[i]<<","<<visitdatepatient[i]<<","<<patientphonenumber[i]<<endl;

}

addpatientm.close();

}

void readaddpatient(int &addpatientindex, string patientname[], string patientage[], string patientcnic[], string patientbloodgroup[], string patienthistory[], string visitdatepatient[], string patientphonenumber[], string filenameaddpatient)

{

addpatientindex=0;

string record;

fstream data;

data.open(filenameaddpatient, ios::in);

while(!(data.eof()))

{

getline(data, record);

if(record=="")

{

break;

}

patientname[addpatientindex]=getField(record, 1);

patientage[addpatientindex]=getField(record, 2);

patientcnic[addpatientindex]=getField(record, 3);

patientbloodgroup[addpatientindex]=getField(record, 4);

patienthistory[addpatientindex]=getField(record, 5);

visitdatepatient[addpatientindex]=getField(record, 6);

patientphonenumber[addpatientindex]=getField(record, 7);

addpatientindex++;

}

data.close();

}

void filehandlingprofitloss(string patientcharges, int addpatientindex, string filenameprofitloss)

{

fstream profitm;

profitm.open(filenameprofitloss, ios::out);

profitm<<patientcharges<<endl;

profitm.close();

}

void readprofitloss(string patientcharges, int addpatientindex, string filenameprofitloss)

{

addpatientindex=0;

string b;

string record;

fstream data;

data.open(filenameprofitloss, ios::in);

getline(data, record);

patientcharges=getField(record, 1);

b=getField(record, 2);

if(b=="")

{

addpatientindex=0;

}

if(b!="")

{

addpatientindex=stoi(b);

}

data.close();

}

void filehandlingchangingsalary(string doctorsalary, string managersalary, string filenamechangingsalary)

{

fstream line;

line.open(filenamechangingsalary, ios::out);

line<<doctorsalary<<","<<managersalary<<endl;

line.close();

}

void readchangingsalary(string doctorsalary, string managersalary, string filenamechangingsalary)

{

string record;

fstream data;

data.open(filenamechangingsalary, ios::in);

getline(data, record);

doctorsalary=getField(record, 1);

managersalary=getField(record, 2);

data.close();

}

void filehandlingaddbed(string bedadd, string filenameaddbed)

{

fstream line;

line.open(filenameaddbed, ios::out);

line<<bedadd<<endl;

line.close();

}

void readaddbed(string bedadd, string filenameaddbed)

{

string record;

fstream data;

data.open(filenameaddbed, ios::in);

getline(data, record);

bedadd=getField(record, 1);

data.close();

}

void filehandlingadddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor)

{

fstream line;

line.open(filenameadddoctor, ios::out);

for(int i=0; i<adddoctorindex; i++)

{

line<<doctorname[i]<<","<<doctorspeciality[i]<<endl;

}

line.close();

}

void readadddoctor(string doctorname[], string doctorspeciality[], int &adddoctorindex, string filenameadddoctor)

{

adddoctorindex=0;

string record;

fstream data;

data.open(filenameadddoctor, ios::in);

while(!(data.eof()))

{

getline(data, record);

if(record=="")

{

break;

}

doctorname[adddoctorindex]=getField(record, 1);

doctorspeciality[adddoctorindex]=getField(record, 2);

adddoctorindex++;

}

data.close();

}

void filehandlingaddequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameaddequipmentmanager)

{

fstream line;

line.open(filenameaddequipmentmanager, ios::out);

line<<equipmentmanagername<<","<<equipmentmanagergender<<","<<equipmentmanagercnic<<","<<endl;

line.close();

}

void readaddequipmentmanager(string equipmentmanagername, string equipmentmanagergender, string equipmentmanagercnic, int &managercheck, string filenameequipmentmanager)

{

managercheck=0;

string b;

string record;

fstream data;

data.open(filenameequipmentmanager, ios::in);

getline(data, record);

equipmentmanagername=getField(record, 1);

equipmentmanagergender=getField(record, 2);

equipmentmanagercnic=getField(record, 3);

b=getField(record, 2);

if(record!="")

{

managercheck=1;

}

data.close();

}

void filehandlingselectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[])

{

fstream line;

line.open("selectDoctor.txt", ios::out);

line<<selectdoctorname<<","<<selectdoctorspeciality<<endl;

line.close();

}

void readselectdoctor(string selectdoctorname, string selectdoctorspeciality, int &adddoctorindex, string doctorname[], string doctorspeciality[], string hire[])

{

adddoctorindex=0;

string b;

string record;

fstream data;

data.open("selectDoctor.txt", ios::in);

getline(data, record);

selectdoctorname=getField(record, 1);

selectdoctorspeciality=getField(record, 2);

b=getField(record, 3);

if(b=="")

{

adddoctorindex=0;

}

if(b!="")

{

adddoctorindex=stoi(b);

}

data.close();

}

void filehandlingbillinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[])

{

fstream line;

line.open("BillingInvoice.txt", ios::out);

for(int i=0; servicetype[i]!="\0"; i++)

{

line<<servicetype[i]<<","<<daysstayes[i]<<","<<roomtype[i]<<endl;

}

line.close();

}

void readbilinginvoice(string servicetype[], int &addbillingindex, string daysstayes[], string roomtype[])

{

addbillingindex=0;

string record;

fstream data;

data.open("BillingInvoice.txt", ios::in);

while(!(data.eof()))

{

getline(data, record);

servicetype[addbillingindex]=getField(record, 1);

daysstayes[addbillingindex]=getField(record, 3);

roomtype[addbillingindex]=getField(record, 4);

addbillingindex++;

}

data.close();

}

void filehandlingpatientreview(string review[], int &addpatientindex)

{

fstream line;

line.open("Patient Review.txt", ios::out);

for(int i=0; review[i]!="\0"; i++)

{

line<<review[i]<<endl;

}

line.close();

}

void readpatientreview(string review[], int &addpatientindex)

{

addpatientindex=0;

string record;

fstream data;

data.open("Patient Review.txt", ios::in);

while(!(data.eof()))

{

getline(data, record);

review[addpatientindex]=getField(record, 1);

addpatientindex++;

}

data.close();

}

void filehandlingbuymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines)

{

fstream line;

line.open(filenamebuymedicines, ios::out);

line<<purchasemedicines<<","<<buymedicinequantity<<endl;

line.close();

}

void readbuymedicines(string buymedicinequantity, string purchasemedicines, int &addmedicineindex, string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], string filenamebuymedicines)

{

addmedicineindex=0;

string b;

string record;

fstream data;

data.open(filenamebuymedicines, ios::in);

getline(data, record);

buymedicinequantity=getField(record, 1);

purchasemedicines=getField(record, 2);

b=getField(record, 3);

if(b=="")

{

addmedicineindex=0;

}

if(b!="")

{

addmedicineindex=stoi(b);

}

data.close();

}

void filehandlingaddmedicine(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine)

{

fstream line;

line.open(filenameaddmedicine, ios::out);

for(int i=0; i<addmedicineindex; i++)

{

line<<medicinename[i]<<","<<medicinequantity[i]<<","<<medicineexpirydate[i]<<","<<medicineprice[i]<<","<<medicinebatchnumber[i]<<endl;

}

line.close();

}

void readaddmedicine(string medicinename[], string medicinequantity[], string medicineexpirydate[], string medicineprice[], string medicinebatchnumber[], int &addmedicineindex, string filenameaddmedicine)

{

addmedicineindex=0;

string record;

fstream data;

data.open(filenameaddmedicine, ios::in);

while(!(data.eof()))

{

getline(data, record);

medicinename[addmedicineindex]=getField(record, 1);

medicinequantity[addmedicineindex]=getField(record, 2);

medicineexpirydate[addmedicineindex]=getField(record, 3);

medicineprice[addmedicineindex]=getField(record, 4);

medicinebatchnumber[addmedicineindex]=getField(record, 5);

addmedicineindex++;

}

data.close();

}

void filehandlingaddemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[])

{

fstream line;

line.open("AddPatient.txt", ios::out);

for(int i=0; i<addemployindex; i++)

{

line<<employname[i]<<","<<employcnic[i]<<","<<employphonenumber[i]<<","<<employgender[i]<<endl;

}

line.close();

}

void readaddemploy(string employname[], int &addemployindex, string employcnic[], string employphonenumber[], string employgender[])

{

addemployindex=0;

string record;

fstream data;

data.open("AddPatient.txt", ios::in);

while(!(data.eof()))

{

getline(data, record);

employname[addemployindex]=getField(record, 1);

employcnic[addemployindex]=getField(record, 3);

employphonenumber[addemployindex]=getField(record, 4);

employgender[addemployindex]=getField(record, 5);

addemployindex++;

}

data.close();

}

void filehandlingaddequipments(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment)

{

fstream line;

line.open(filenameaddequipment, ios::out);

for(int i=0; i<addequipmentindex; i++)

{

line<<equipmentname[i]<<","<<equipmentquantity[i]<<","<<equipmentprize[i]<<","<<equipmentbatchnumber[i]<<endl;

}

line.close();

}

void readaddequipments(string equipmentname[], int &addequipmentindex, string equipmentquantity[], string equipmentprize[], string equipmentbatchnumber[], string filenameaddequipment)

{

addequipmentindex=0;

string record;

fstream data;

data.open(filenameaddequipment, ios::in);

while(!(data.eof()))

{

getline(data, record);

if(record=="")

{

break;

}

equipmentname[addequipmentindex]=getField(record, 1);

equipmentquantity[addequipmentindex]=getField(record, 2);

equipmentprize[addequipmentindex]=getField(record, 3);

equipmentbatchnumber[addequipmentindex]=getField(record, 4);

addequipmentindex++;

}

data.close();

}