  
  
  
  
INTRODUCTION OF PROJECT:

The purpose of the project entitled as “Hospital Management System” is to computerize the  
Front Office Management of Hospital to develop software which is user friendly simple, fast,  
and cost effective. It deals with the collection of patient’s information like add patient, update  
patient, delete patient, search patient, view patient diagnosis, etc. Traditionally, it was done  
manually.

OBJECTIVE AND SCOPE:

The main function of the system is register and store patient details  and retrieve these details as and when required, and also to manipulate these details  
meaningfully. The Hospital Management System can be entered using a username and  
password. It is accessible by an Admin

Flowchart:  
Diagram

Description automatically generated

WORKING OF PROJECT:

Idea was to work on a c++ program and to create a hospital management system which makes real time management easy. Data can easily be changed, added, updated, viewed and deleted.

## Program include these major parts:

# Admin Login

# Visitor/Patient

# record new patient

# search patient

# delete patient details

# change/add patient medicine

# print patient info

# view wards and availble ward

# reduce or increase bill

# change of patient’s bed

# filing and restoring

This project taught us new things and new ways of implementations of code in C++.The code include most library and methods already taught in the semester course for example

## Reading a file of type .txt using library fstream.

## Using condition statements like if, else if, else and also switch cases.

## Making use of array and strings.

## Creating function by reference.

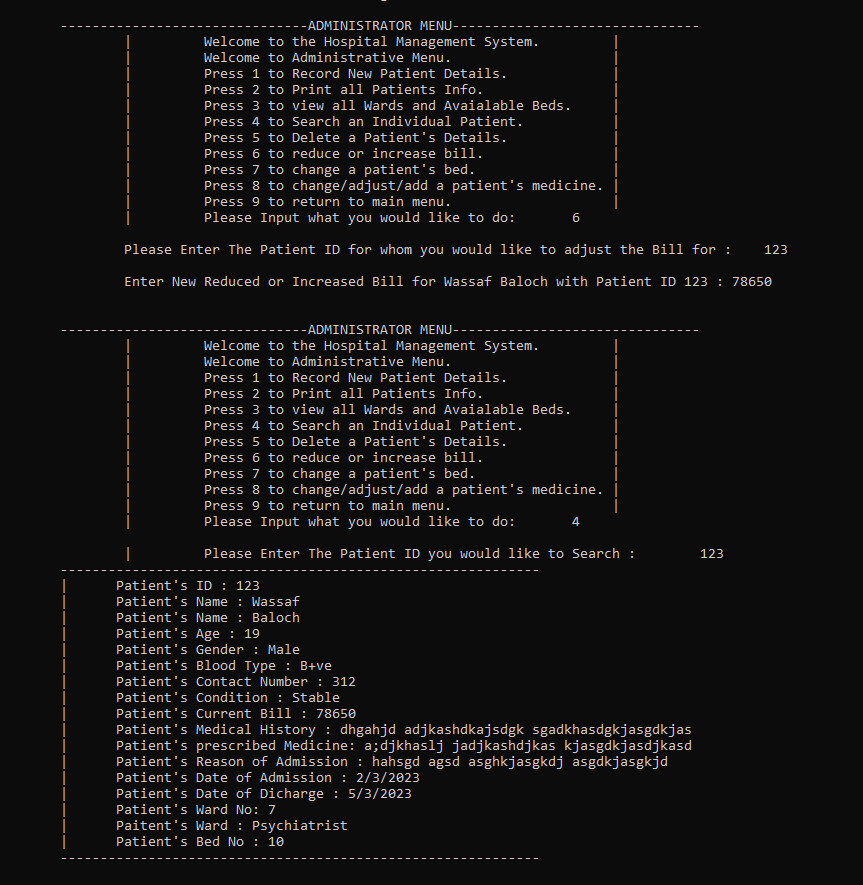
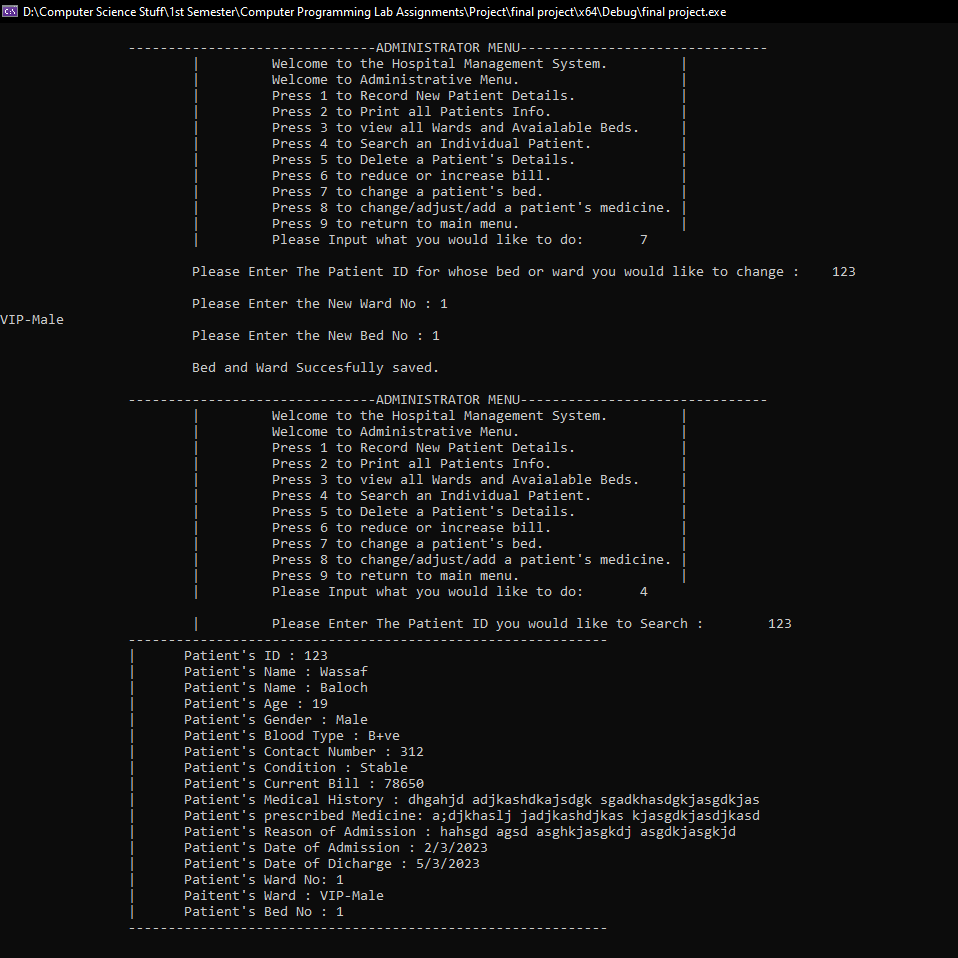
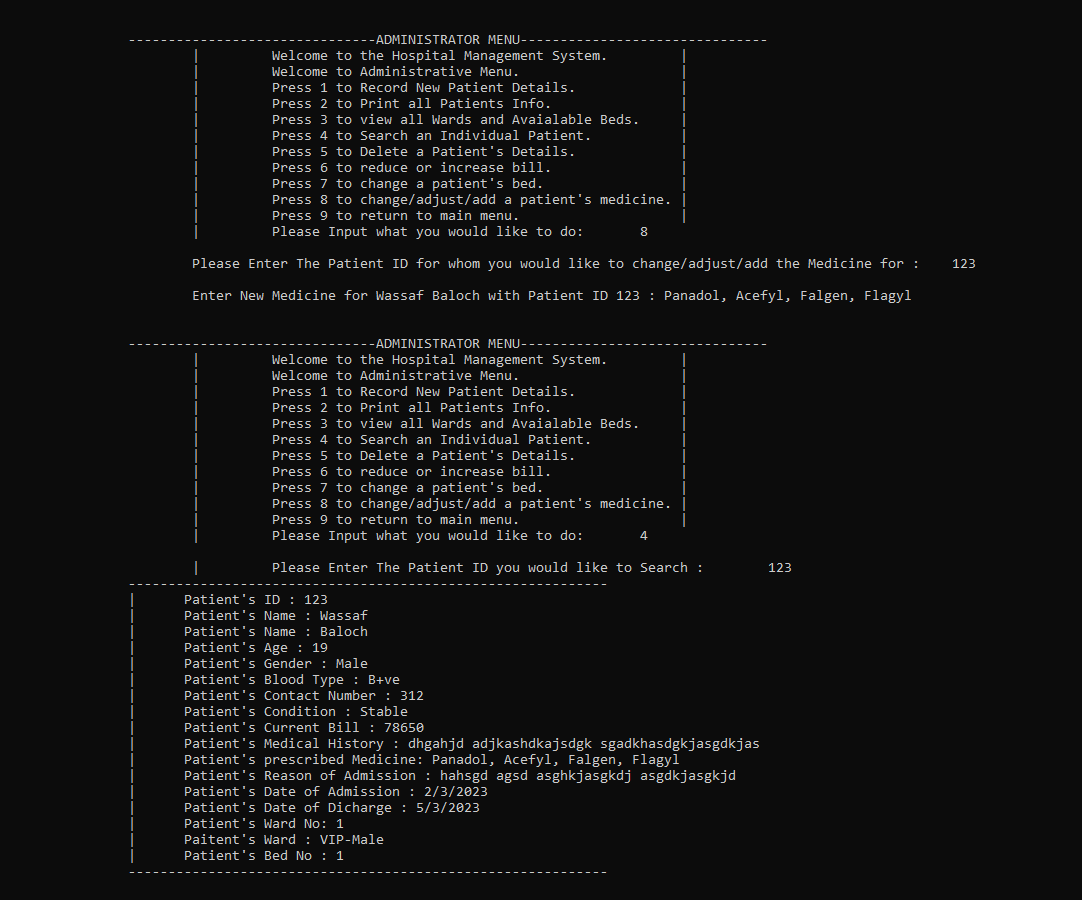
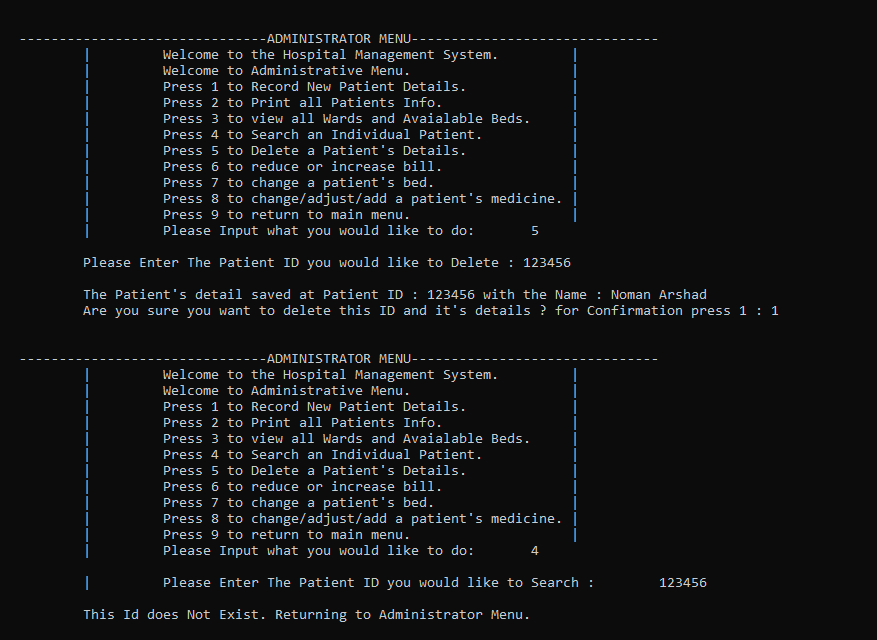
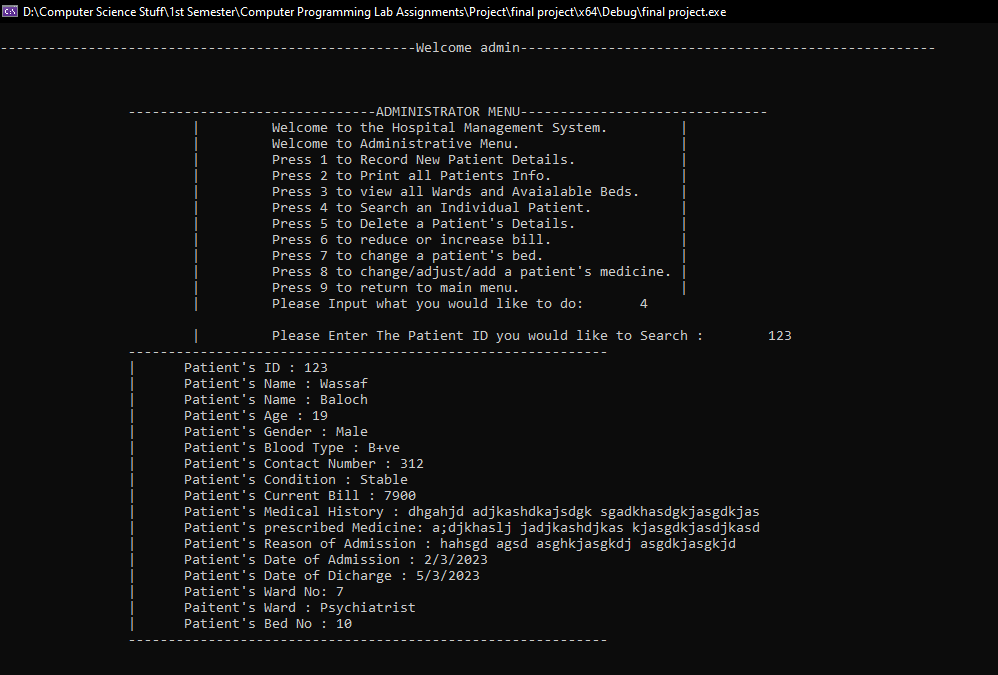
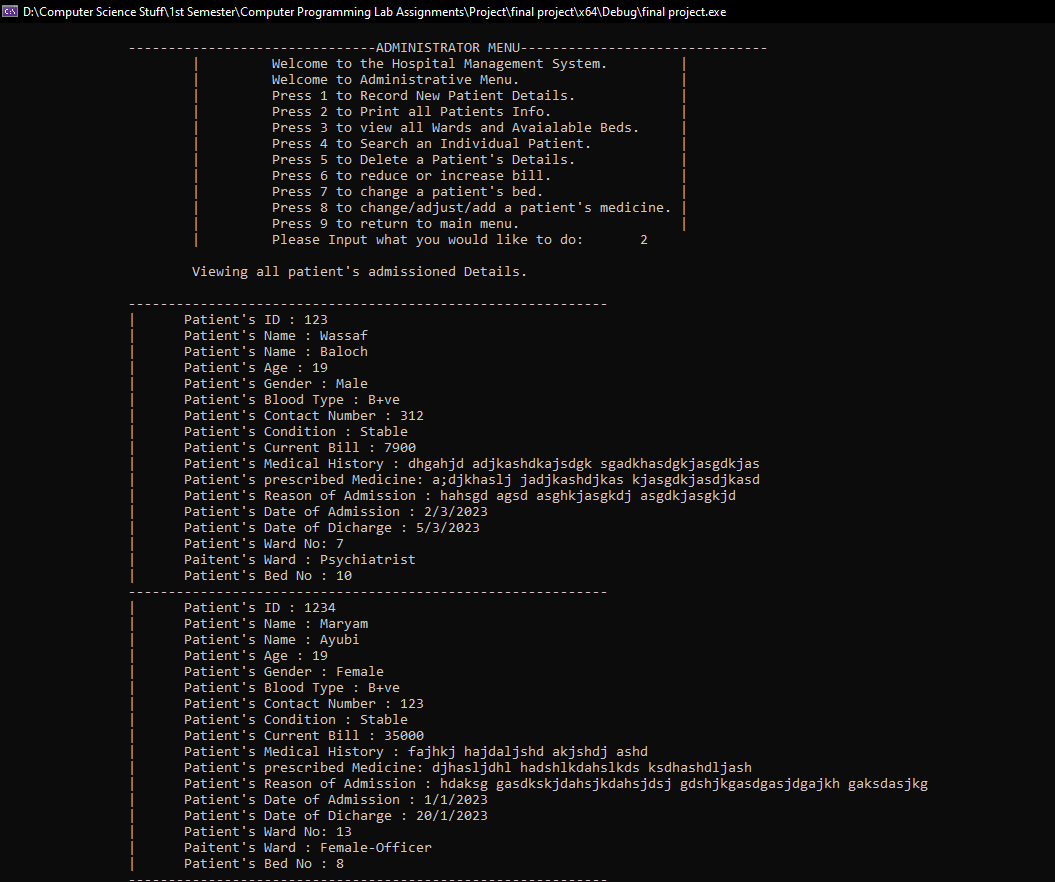
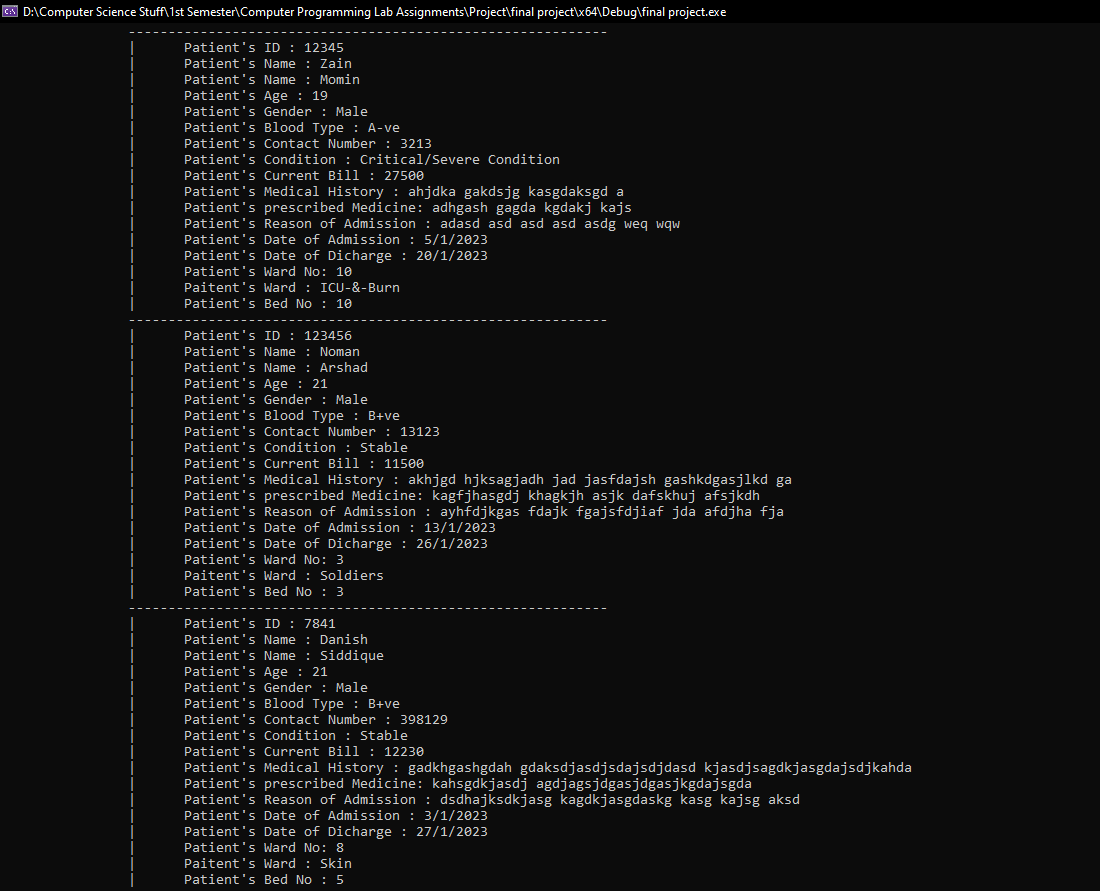
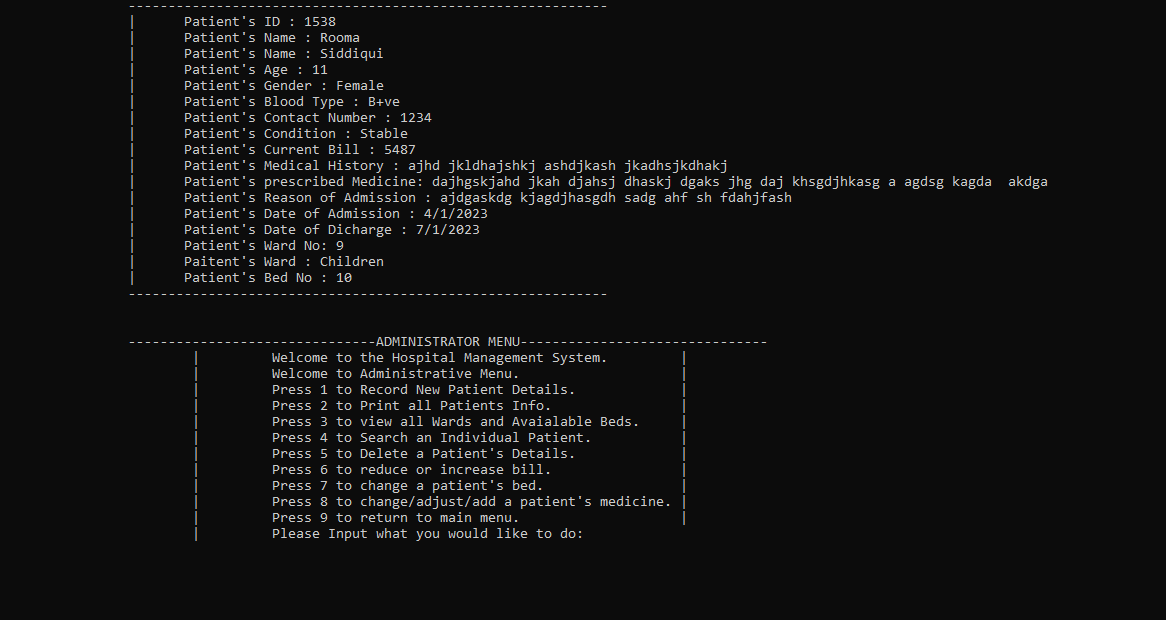
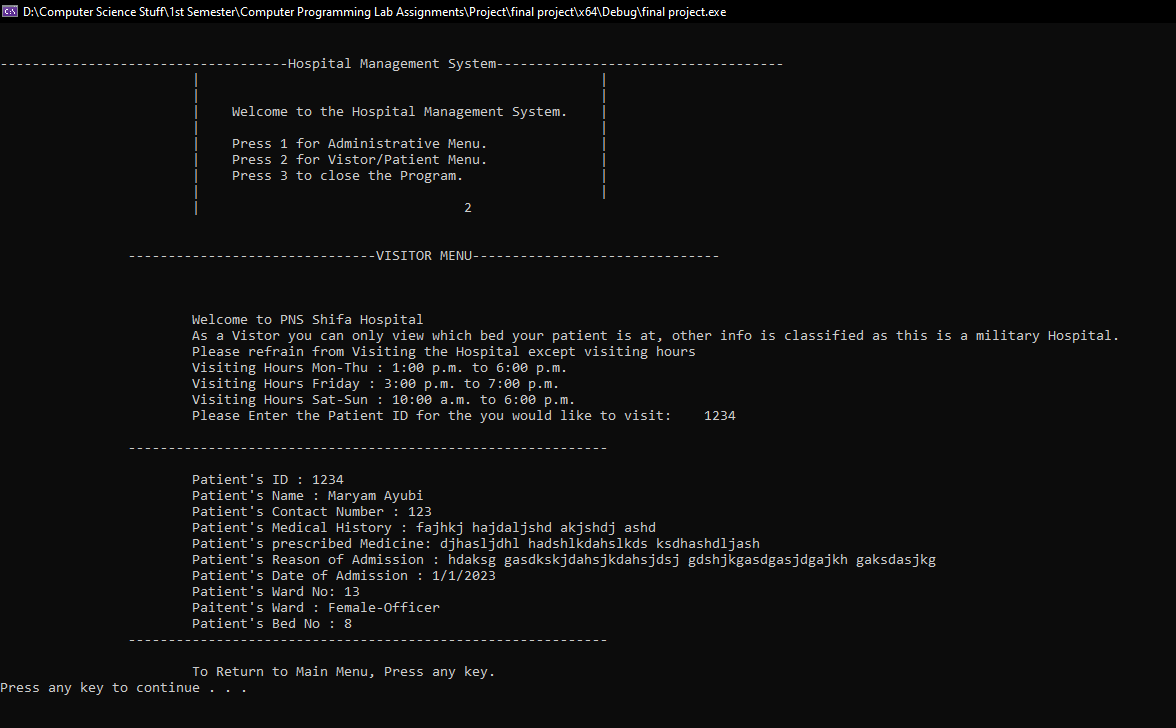
## Using for loops for writing and reading especially arrays. sending value through different methods using complex structures

.

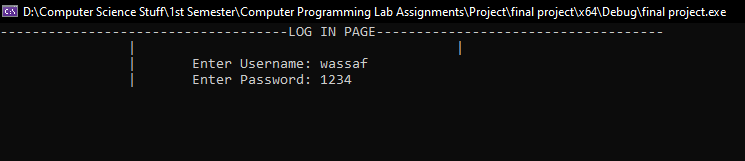
Picture of Program:

The Following screenshots includes:

* Visitor menu
* Viewing all patients
* Searching individual patient
* Deleting id, patient information
* Changing patient’s medicine
* Changing bed
* Changing/updating bill
* Availability of wards and beds
* Administrative menu
* Admin login
* Addition of patient

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated

Code:

#include <iostream>

#include <string>

#include <fstream>

#include <istream>

#include <stdlib.h>

using namespace std;

void title(); //Function for Title

void administratorm(); //Function for Administrator Menu

void visitor\_patientm(); //Function for Visitor to check where their patient is by entering ID

int adminlogin(); //Administrator Log in Function

bool isempty();

void view(); //View all Wards and Beds to see Avialability

void addp(); //Add new Patient

void viewpatients(); //View all Patients and their Details

int checkbed(int x); //Check if Desired Bed is Empty or Not

int searchPindex(int tempID); //return the index at which the desired Unique ID searched by User is stored.

void view1patient(int x); //View Details of Patient searched invidually.

void dpd(); //delete a single patients details.

void cfees(); //change fees/bill of a patient.

void cbed(); //change bed of a patient.

void file(); //store all data from structure to file(s).

void restore(); //restore all data from file(s).

void cmedicine(); //change medicine of a Patient.

struct details {

string fname, lname; //first name and last name

string roa; // reason of admission

string doa; // date of admission

string dod; // date of discharge

int pid = 0; // unique id of patient

int age = 0; //age

long int no = 0; //contact No.

string medicalhistory; //medical history

string ward; //ward name

int wardno = 0; //ward number

int bedno = -1; //bed number

string bloodtype;

string gender;

int severe = 0;

int fees = 0;

string medicine;

};

details patient[210];

string ward[14] = { "VIP-Male", "Male-Officers", "Soldiers", "Orthopedic", "Eye-&-ENT", "Surgical", "Psychiatrist", "Skin" , "Children", "ICU-&-Burn", "Female-Medical-&-Surgical", "Female-Gynaecology", "Female-Officer", "Tuberculosis" };

int wardfees[14] = { 15000, 13000, 5000, 8000, 6000, 7000, 50000, 11000, 4500, 20000, 7500, 13000, 15000, 8000 };

int wardno[14] = { 1,2,3,4,5,6,7,8,9,10,11,12,13,14 };

int wardbed[14][15] = { 0 };

int pindex = -1;

int main()

{

int c;

do {

system("cls");

restore();

title();

cout << "\t\t\t| |\n";

cout << "\t\t\t| |\n";

cout << "\t\t\t| Welcome to the Hospital Management System. |\n";

cout << "\t\t\t| |\n";

cout << "\t\t\t| Press 1 for Administrative Menu. |\n";

cout << "\t\t\t| Press 2 for Vistor/Patient Menu. |\n";

cout << "\t\t\t| Press 3 to close the Program. |\n";

cout << "\t\t\t| |\n";

cout << "\t\t\t| ";

cin >> c;

if (c == 1) {

adminlogin();

}

else if (c == 2) {

visitor\_patientm();

}

} while (c == 3);

return 0;

}

void title() {

cout << "\n\n------------------------------------Hospital Management System------------------------------------\n";

}

void administratorm()

{

int c;

do {

cout << "\n\n\t\t-------------------------------ADMINISTRATOR MENU-------------------------------\n";

cout << "\t\t\t| Welcome to the Hospital Management System. |\n";

cout << "\t\t\t| Welcome to Administrative Menu. |\n";

cout << "\t\t\t| Press 1 to Record New Patient Details. |\n";

cout << "\t\t\t| Press 2 to Print all Patients Info. |\n";

cout << "\t\t\t| Press 3 to view all Wards and Avaialable Beds. |\n";

cout << "\t\t\t| Press 4 to Search an Individual Patient. |\n";

cout << "\t\t\t| Press 5 to Delete a Patient's Details. |\n";

cout << "\t\t\t| Press 6 to reduce or increase bill. |\n";

cout << "\t\t\t| Press 7 to change a patient's bed. |\n";

cout << "\t\t\t| Press 8 to change/adjust/add a patient's medicine. |\n";

cout << "\t\t\t| Press 9 to return to main menu. |\n";

cout << "\t\t\t| Please Input what you would like to do: \t";

cin >> c;

if (c == 1)

{

addp();

file();

}

else if (c == 2)

{

viewpatients();

}

else if (c == 3)

{

view();

}

else if (c == 4)

{

cout << "\n\t\t\t| Please Enter The Patient ID you would like to Search : \t";

int tempID, tempindex;

cin >> tempID;

tempindex = searchPindex(tempID);

view1patient(tempindex);

}

else if (c == 5)

{

dpd();

}

else if (c == 6)

{

cfees();

}

else if (c == 7)

{

cbed();

}

else if (c == 8)

{

cmedicine();

}

} while (c != 9);

main();

}

int adminlogin()

{

bool correct = false;

int x;

string user, pass, u, p;

system("cls");

cout << "------------------------------------LOG IN PAGE------------------------------------\n";

cout << "\t\t| |\n";

cout << "\t\t| Enter Username: ";

cin >> user;

cout << "\t\t| Enter Password: ";

cin >> pass;

ifstream file("login.txt", ios::in);

while (file >> u >> p) {

if (u == user && p == pass) {

correct = 1;

}

}

file.close();

if (correct == 1) {

system("cls");

cout << "\n----------------------------------------------------Welcome " << user << "----------------------------------------------------\n\n";

administratorm();

}

else if (correct == 0) {

cout << "\n\t\t\tIncorrect ID or Password, Press 1 to try again or 0 to go to main menu.";

cin >> x;

if (x == 1) {

adminlogin();

}

else {

main();

}

}

return 0;

}

void view()

{

title();

cout << "\n\t\tNow Viewing all Wards and Beds available.";

for (int i = 0; i < 14; i++)

{

cout << "\n\t\t\tWard No : " << "\t" << wardno[i];

cout << "\n\t\t\tWard Name : " << "\t" << ward[i] << " Ward";

cout << "\n\t-------------------------------All beds-------------------------------\n";

for (int j = 0; j < 15; j++)

{

if (wardbed[i][j] == 0)

{

cout << "\t\tBed Number : " << j + 1 << " is available\n";

}

else {

cout << "\t\tBed Number : " << j + 1 << " already has a patient.\n";

}

}

}

system("pause");

administratorm();

}

void addp()

{

system("cls");

int a, x, y, b, mb, tempID;

a = 0;

pindex++;

cout << "\n\n\t\t\tNew Patient Data Entry.";

cout << "\n\t\t\tEnter Patient Unique ID : ";

bool repeated = false; //code for unique ID starts from here

do {

cout << "\n\t\t\tMust be Numeric!!!";

cout << "\n\t\t\tEnter Id : ";

cin >> tempID;

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

{

if (pindex == i)

{

i++;

}

if (tempID == patient[i].pid)

{

repeated = true;

break;

}

}

if (repeated) {

cout << "\n\t\t\tId is already, Use another\n";

pindex--;

break;

}

} while (repeated == true);

if (!repeated) { //only allows entry of patient if ID is unique or returns to administrator menu.

patient[pindex].pid = tempID;

cout << "\n\t\t\tEnter Patient First Name : ";

cin >> patient[pindex].fname;

cout << "\n\t\t\tEnter Patient Last Name : ";

cin >> patient[pindex].lname;

cout << "\n\t\t\tEnter Patient Gender : ";

getline(cin >> ws, patient[pindex].gender);

cout << "\n\t\t\tEnter Patient Age : ";

cin >> patient[pindex].age;

cout << "\n\t\t\tEnter Contact : ";

cin >> patient[pindex].no;

cout << "\n\t\t\tEnter Patient's Short Medical history if any : ";

getline(cin >> ws, patient[pindex].medicalhistory);

cout << "\n\t\t\tEnter Patient Blood Type : ";

getline(cin >> ws, patient[pindex].bloodtype);

cout << "\n\t\t\tEnter the Medicines Prescribed to the patient : ";

getline(cin >> ws, patient[pindex].medicine);

cout << "\n\t\t\tPlease Enter the Medicinal Bill: ";

cin >> mb;

cout << "\n\t\t\tEnter Reason of Admission : ";

getline(cin >> ws, patient[pindex].roa);

cout << "\n\t\t\tEnter Date of Admission (Format DD/MM/YYYY): ";

getline(cin >> ws, patient[pindex].doa);

cout << "\n\t\t\tEnter Date of Discharge (Format DD/MM/YYYY): ";

getline(cin >> ws, patient[pindex].dod);

cout << "\n\t\t\tPlease Enter the Condition of the Patient : \n\t\t\t1 For Severe & 0 for Stable : ";

cin >> y;

if (y == 1) //condition to send severe patients direct to ICU

{

patient[pindex].severe = 1;

if (patient[pindex].severe == 1)

{

cout << "\n\t\t\tSince Patient is in Severe Condition he/she is being admitted Directly to Ward Number : " << wardno[9] << ward[9] << " Ward.";

patient[pindex].ward = ward[9];

patient[pindex].wardno = 9;

cout << "\n\t\t\tListing all the Beds in Ward Number : " << patient[pindex].wardno + 1 << " " << patient[pindex].ward << " Ward.";

for (int i = 0; i < 15; i++) //will view all beds in Ward

{

if (wardbed[9][i] == 0)

{

cout << "\n\t\t\tBed Number : " << i + 1 << " is available.";

}

else {

cout << "\n\t\t\tBed Number : " << i + 1 << " already has a patient.";

}

}

cout << "\n\t\t\tNow Enter the Bed you would you like to admit the patient to : ";

cin >> b;

checkbed(b);

patient[pindex].bedno = b - 1;

wardbed[patient[pindex].wardno][patient[pindex].bedno] = 1; //changes availability of bed in 2D Array

}

}

else {

if (patient[pindex].age <= 12) //All children sent to Children's Ward.

{

cout << "\n\t\t\tSince The patient's age is less than or equal to 12, He is directly being admitted to Ward Number 9 Children's Ward.";

patient[pindex].ward = ward[8];

patient[pindex].wardno = 8;

cout << "\n\t\t\tListing all the Beds in Ward Number : " << patient[pindex].wardno + 1 << " " << patient[pindex].ward << " Ward.";

for (int i = 0; i < 15; i++) //will view all beds in Ward

{

if (wardbed[patient[pindex].wardno][i] == 0)

{

cout << "\n\t\t\tBed Number : " << i + 1 << " is available.";

}

else {

cout << "\n\t\t\tBed Number : " << i + 1 << " already has a patient.";

}

}

cout << "\n\t\t\tNow Enter the Bed you would you like to admit the patient to : ";

cin >> b;

checkbed(b);

patient[pindex].bedno = b - 1;

wardbed[patient[pindex].wardno][patient[pindex].bedno] = 1; //changes availability of bed in 2D Array

}

else { //other wards condition

cout << "\n\t\t\tPlease Enter the Ward you would like to admit the patient in : ";

cout << "\n\t\t\tTo First See all the Ward Names, Numbers and beds available Press 1 or else Press 0 : ";

cin >> x;

if (x == 1)

{

view();

}

cout << "\n\t\t\tSince Patient is above the age of 12 he/she can be admitted in any ward except Ward 9 since it is a Children Ward.";

cout << "\n\t\t\tPlease Enter Wardno : ";

cin >> patient[pindex].wardno;

patient[pindex].wardno = patient[pindex].wardno - 1;

patient[pindex].ward = ward[patient[pindex].wardno];

cout << "\n\t\t\tListing all the Beds in Ward Number : " << patient[pindex].wardno + 1 << " " << patient[pindex].ward << " Ward.";

for (int i = 0; i < 15; i++) //will view all beds in Ward

{

if (wardbed[patient[pindex].wardno][i] == 0)

{

cout << "\n\t\t\tBed Number : " << i + 1 << " is available.";

}

else {

cout << "\n\t\t\tBed Number : " << i + 1 << " already has a patient.";

}

}

cout << "\n\t\t\tNow Enter the Bed you would you like to admit the patient to : ";

cin >> b;

checkbed(b);

patient[pindex].bedno = b - 1;

wardbed[patient[pindex].wardno][patient[pindex].bedno] = 1; //changes availability of bed in 2D Array

}

}

cout << "\n\t\t\tPlease Recieve Fees of bed for admission : ";

cout << "\n\t\t\tFees for the Desired Ward is : PKR" << wardfees[patient[pindex].wardno];

cout << "\n\t\t\tPlease Enter 1 if fees is recieved : ";

cin >> a;

while (a != 1) {

cout << "\n\t\t\tProgram Cannot Continue without fees being marked as recieved.";

cin >> a;

}

patient[pindex].fees = wardfees[patient[pindex].wardno];

cout << "\n\t\t\tNow adding Medicinal bill to overall.";

patient[pindex].fees = patient[pindex].fees + mb;

cout << "\n\t\t\tNew Combined Fees/Bill is : " << patient[pindex].fees;

}

}

void viewpatients()

{

cout << "\n\t\t\tViewing all patient's admissioned Details.\n\n";

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

{

cout << "\t\t------------------------------------------------------------\n";

cout << "\t\t| Patient's ID : " << patient[i].pid << endl;

cout << "\t\t| Patient's Name : " << patient[i].fname << endl;

cout << "\t\t| Patient's Name : " << patient[i].lname << endl;

cout << "\t\t| Patient's Age : " << patient[i].age << endl;

cout << "\t\t| Patient's Gender : " << patient[i].gender << endl;

cout << "\t\t| Patient's Blood Type : " << patient[i].bloodtype << endl;

cout << "\t\t| Patient's Contact Number : " << patient[i].no << endl;

cout << "\t\t| Patient's Condition : ";

if (patient[i].severe == 1) {

cout << "Critical/Severe Condition" << endl;

}

else {

cout << "Stable" << endl;

}

cout << "\t\t| Patient's Current Bill : " << patient[i].fees << endl;

cout << "\t\t| Patient's Medical History : " << patient[i].medicalhistory << endl;

cout << "\t\t| Patient's prescribed Medicine: " << patient[i].medicine << endl;

cout << "\t\t| Patient's Reason of Admission : " << patient[i].roa << endl;

cout << "\t\t| Patient's Date of Admission : " << patient[i].doa << endl;

cout << "\t\t| Patient's Date of Dicharge : " << patient[i].dod << endl;

cout << "\t\t| Patient's Ward No: " << patient[i].wardno + 1 << endl;

cout << "\t\t| Paitent's Ward : " << patient[i].ward << endl;

cout << "\t\t| Patient's Bed No : " << patient[i].bedno + 1 << endl;

}

cout << "\t\t------------------------------------------------------------\n";

}

int searchPindex(int tempID)

{

bool found = false;

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

{

if (tempID == patient[i].pid)

{

found = true;

return i;

}

else {

found = false;

}

}

if (found == false)

{

cout << "\n\t\t\tThis Id does Not Exist. Returning to Administrator Menu.";

administratorm();

return 0;

}

}

void view1patient(int x)

{

cout << "\t\t------------------------------------------------------------\n";

cout << "\t\t| Patient's ID : " << patient[x].pid << endl;

cout << "\t\t| Patient's Name : " << patient[x].fname << endl;

cout << "\t\t| Patient's Name : " << patient[x].lname << endl;

cout << "\t\t| Patient's Age : " << patient[x].age << endl;

cout << "\t\t| Patient's Gender : " << patient[x].gender << endl;

cout << "\t\t| Patient's Blood Type : " << patient[x].bloodtype << endl;

cout << "\t\t| Patient's Contact Number : " << patient[x].no << endl;

cout << "\t\t| Patient's Condition : ";

if (patient[x].severe == 1) {

cout << "Critical/Severe Condition" << endl;

}

else {

cout << "Stable" << endl;

}

cout << "\t\t| Patient's Current Bill : " << patient[x].fees << endl;

cout << "\t\t| Patient's Medical History : " << patient[x].medicalhistory << endl;

cout << "\t\t| Patient's prescribed Medicine: " << patient[x].medicine << endl;

cout << "\t\t| Patient's Reason of Admission : " << patient[x].roa << endl;

cout << "\t\t| Patient's Date of Admission : " << patient[x].doa << endl;

cout << "\t\t| Patient's Date of Dicharge : " << patient[x].dod << endl;

cout << "\t\t| Patient's Ward No: " << patient[x].wardno + 1 << endl;

cout << "\t\t| Paitent's Ward : " << patient[x].ward << endl;

cout << "\t\t| Patient's Bed No : " << patient[x].bedno + 1 << endl;

cout << "\t\t------------------------------------------------------------\n";

}

int checkbed(int x)

{

bool avail = false;

x = x - 1;

do {

if (wardbed[patient[pindex].wardno][x] == 1)

{

avail = false;

cout << "\n\t\t\tBed is already occupied by a patient enter another one : ";

cin >> x;

x = x - 1;

}

else {

avail = true;

}

} while (avail == false);

return x;

}

void dpd()

{

if (isempty())

{

cout << "\n\t\t\tNo Patient Details have been saved.";

}

else {

cout << "\n\t\t\tPlease Enter The Patient ID you would like to Delete : ";

int tempID, tempindex, c;

cin >> tempID;

tempindex = searchPindex(tempID);

cout << "\n\t\t\tThe Patient's detail saved at Patient ID : " << patient[tempindex].pid << " with the Name : " << patient[tempindex].fname << " " << patient[tempindex].lname;

cout << "\n\t\t\tAre you sure you want to delete this ID and it's details ? for Confirmation press 1 : ";

cin >> c;

if (c == 1)

{

wardbed[patient[tempindex].wardno][patient[tempindex].bedno] = 0;

for (int i = tempindex; i < pindex; i++)

{

patient[i].pid = patient[i + 1].pid;

patient[i].fname = patient[i + 1].fname;

patient[i].lname = patient[i + 1].lname;

patient[i].age = patient[i + 1].age;

patient[i].gender = patient[i + 1].gender;

patient[i].bloodtype = patient[i + 1].bloodtype;

patient[i].no = patient[i + 1].no;

patient[i].severe = patient[i + 1].severe;

patient[i].fees = patient[i + 1].fees;

patient[i].medicalhistory = patient[i + 1].medicalhistory;

patient[i].medicine = patient[i + 1].medicine;

patient[i].roa = patient[i + 1].roa;

patient[i].doa = patient[i + 1].doa;

patient[i].dod = patient[i + 1].dod;

patient[i].wardno = patient[i + 1].wardno;

patient[i].ward = patient[i + 1].ward;

patient[i].bedno = patient[i + 1].bedno;

}

pindex--;

file();

}

else {

administratorm();

}

}

}

bool isempty()

{

if (pindex == -1)

{

return true;

}

else {

return false;

}

}

void cfees()

{

cout << "\n\t\t\tPlease Enter The Patient ID for whom you would like to adjust the Bill for : \t";

int tempID, tempindex, n;

cin >> tempID;

tempindex = searchPindex(tempID);

cout << "\n\t\t\tEnter New Reduced or Increased Bill for " << patient[tempindex].fname << " " << patient[tempindex].lname << " with Patient ID " << patient[tempindex].pid << " : ";

cin >> n;

patient[tempindex].fees = n;

file();

}

void cbed()

{

cout << "\n\t\t\tPlease Enter The Patient ID for whose bed or ward you would like to change :\t";

int tempID, tempindex, newbed, newward;

cin >> tempID;

tempindex = searchPindex(tempID);

wardbed[patient[tempindex].wardno][patient[tempindex].bedno] = 0; //changes availability of bed in 2D Array

cout << "\n\t\t\tPlease Enter the New Ward No : ";

cin >> newward;

patient[tempindex].wardno = newward - 1;

patient[tempindex].ward = ward[newward - 1];

cout << ward[newward - 1];

cout << "\n\t\t\tPlease Enter the New Bed No : ";

cin >> newbed;

patient[tempindex].bedno = newbed - 1;

wardbed[patient[tempindex].wardno][patient[tempindex].bedno] = 1; //changes availability of bed in 2D Array

cout << "\n\t\t\tBed and Ward Succesfully saved.";

file();

}

void file()

{

ofstream outfile("pdata.txt", ios::out);

ofstream outfile2("pmedicalhistory.txt", ios::out);

ofstream outfile3("pmedicine.txt", ios::out);

ofstream outfile4("roa.txt", ios::out);

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

{

outfile4 << patient[i].roa << endl;

outfile3 << patient[i].medicine << endl;

outfile2 << patient[i].medicalhistory << endl;

outfile << patient[i].pid << " ";

outfile << patient[i].fname << " ";

outfile << patient[i].lname << " ";

outfile << patient[i].age << " ";

outfile << patient[i].gender << " ";

outfile << patient[i].bloodtype << " ";

outfile << patient[i].no << " ";

outfile << patient[i].severe << " ";

outfile << patient[i].fees << " ";

outfile << patient[i].doa << " ";

outfile << patient[i].dod << " ";

outfile << patient[i].wardno << " ";

outfile << patient[i].ward << " ";

outfile << patient[i].bedno << endl;

}

outfile.close();

outfile2.close();

outfile3.close();

outfile4.close();

}

void visitor\_patientm()

{

cout << "\n\n\t\t-------------------------------VISITOR MENU-------------------------------\n\n\n";

cout << "\n\t\t\tWelcome to PNS Shifa Hospital";

cout << "\n\t\t\tAs a Vistor you can only view which bed your patient is at, other info is classified as this is a military Hospital.";

cout << "\n\t\t\tPlease refrain from Visiting the Hospital except visiting hours";

cout << "\n\t\t\tVisiting Hours Mon-Thu : 1:00 p.m. to 6:00 p.m.";

cout << "\n\t\t\tVisiting Hours Friday : 3:00 p.m. to 7:00 p.m.";

cout << "\n\t\t\tVisiting Hours Sat-Sun : 10:00 a.m. to 6:00 p.m.";

cout << "\n\t\t\tPlease Enter the Patient ID for the you would like to visit:\t";

int tempID, tempindex;

cin >> tempID;

tempindex = searchPindex(tempID);

cout << "\n\t\t------------------------------------------------------------\n";

cout << "\n\t\t\tPatient's ID : " << patient[tempindex].pid;

cout << "\n\t\t\tPatient's Name : " << patient[tempindex].fname << " " << patient[tempindex].lname;

cout << "\n\t\t\tPatient's Contact Number : " << patient[tempindex].no;

cout << "\n\t\t\tPatient's Medical History : " << patient[tempindex].medicalhistory;

cout << "\n\t\t\tPatient's prescribed Medicine: " << patient[tempindex].medicine;

cout << "\n\t\t\tPatient's Reason of Admission : " << patient[tempindex].roa;

cout << "\n\t\t\tPatient's Date of Admission : " << patient[tempindex].doa;

cout << "\n\t\t\tPatient's Ward No: " << patient[tempindex].wardno + 1;

cout << "\n\t\t\tPaitent's Ward : " << patient[tempindex].ward;

cout << "\n\t\t\tPatient's Bed No : " << patient[tempindex].bedno + 1;

cout << "\n\t\t------------------------------------------------------------\n";

cout << "\n\t\t\tTo Return to Main Menu, Press any key.\n";

system("pause");

main();

}

void restore()

{

ifstream file("pdata.txt", ios::in);

file.seekg(0, ios::end);

if (file.tellg() == 0)

{

cout << "\n\t\t\tNo Records Stored";

}

else {

ifstream file("pdata.txt", ios::in);

string fname, lname, gender, bt, doa, dod, ward;

int i = -1, ID, age, no, fees, wardno, bedno, severe;

while (file >> ID >> fname >> lname >> age >> gender >> bt >> no >> severe >> fees >> doa >> dod >> wardno >> ward >> bedno)

{

i++;

patient[i].pid = ID;

patient[i].fname = fname;

patient[i].lname = lname;

patient[i].age = age;

patient[i].gender = gender;

patient[i].bloodtype = bt;

patient[i].no = no;

patient[i].severe = severe;

patient[i].fees = fees;

patient[i].doa = doa;

patient[i].dod = dod;

patient[i].wardno = wardno;

patient[i].ward = ward;

patient[i].bedno = bedno;

wardbed[wardno][bedno] = 1;

}

pindex = i;

ifstream file2("pmedicalhistory.txt", ios::in);

string medicalhistory;

int j = -1;

while (getline(file2 >> ws, medicalhistory))

{

j++;

patient[j].medicalhistory = medicalhistory;

}

ifstream file3("pmedicine.txt", ios::in);

string medicine;

int k = -1;

while (getline(file3 >> ws, medicine))

{

k++;

patient[k].medicine = medicine;

}

ifstream file4("roa.txt", ios::in);

string roa;

int l = -1;

while (getline(file4 >> ws, roa))

{

l++;

patient[l].roa = roa;

}

}

}

void cmedicine()

{

cout << "\n\t\t\tPlease Enter The Patient ID for whom you would like to change/adjust/add the Medicine for : \t";

int tempID, tempindex;

string newmedicine;

cin >> tempID;

tempindex = searchPindex(tempID);

cout << "\n\t\t\tEnter New Medicine for " << patient[tempindex].fname << " " << patient[tempindex].lname << " with Patient ID " << patient[tempindex].pid << " : ";

getline(cin >> ws, newmedicine);

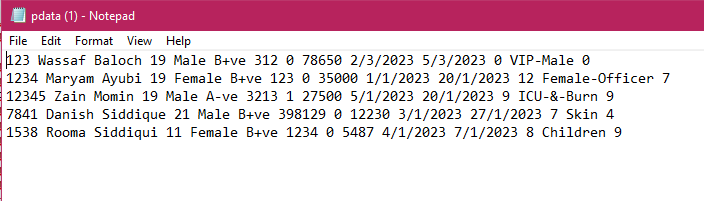
patient[tempindex].medicine = newmedicine;

file();

}

Admin credentialsText

Description automatically generated

Text

Description automatically generatedGraphical user interface, text, application

Description automatically generated