**A**

**PROJECT**

**ON**

**HOSPITAL**

**MANAGEMENT**

**SUBMITTED BY** : **LAKSHAYA SHARMA**

***INDEX***

***S.NO DESCRIPTION PAGE***

***NO.***

*1. ACKNOWLEDGEMENT 2*

*1.1 INTRODUCTION 4*

*1.2 HISTORY OF C 5-6*

*1.3 ADVANTAGE OF C 6*

*1.4 ABOUT HOSPITAL MANAGEMENT 7-11*

*2. PROGRAMMING 12-64*

*3. OUTPUT OF THE PROJECT 65-71*

*4.BIBLIOGRAPHY 72*

**ACKNOWLEDGEMENT**

I, LAKSHAYA SHARMA have successfully

made this project under the guidance of Mr. Pradeep Sir.

I would like to thank my teacher Mr. Pradeep Pandey

for his valuable guidance. The enthusiasm and supervision

for doing this project work. Special thanks to my parents and

friends, who were very keen in my studies and project work

throughout till completion.

Thanks to the support extended by my guide Mr.

Pradeep, whose academic experience and vast experience

helped me to complete this project work and finally my

special thanks to DOEACC, New Delhi, India and DICS,

Pusa Road for providing all their help and environment

reaching up to this point and all my friends and colleagues

whose support were unforgettable.

**INTRODUCTION**

I have undertaken the project of HOSPITAL MANAGEMENT

for hospital; proposed project is present in C in duration of one

year. The main aim of developing hospital management is to

eliminate errors in the present manual system.

**HISTORY OF C**

C seems strange name for a programming language. But this

strange sounding language is one of the most popular computer

languages today because it is a structured, high level machine

independent language. It allows software deposers program

without worrying about the hardware platform where they will be

implemented.

The root of all modern language is ALGOL, introduced in the

early 1960s. ALGOL waste first computer language to use a block

structure. Although it never becomes popular in USA, it was

widely used in Europe. ALGOL gave the concept of structured

programming to the computer since community. Computer

scientist likes Corred Bohm, Guiseppe Jacopin and Edsger Dijkstra

popularized this concept during 1960s.subseqentaly, several

languages were announced.

C is evolved from ALGOL, BCPL and B by Dennis

Ritchie at the Bell Laboratories in 1972.the language becomes

more popular after publication of the book ‘THE C

PROGRAMMING LANGAUGE’ **by B.KERNINGHAM and**

**DENNIS RITCHIE in 1978.**

**Advantages of C**

The C compiler combines the capabilities of an assembly

language with the future of a high level language.

This due to its variety of data types and powerful operators.

It is many times faster than BASIC.

C is highly portable. This means that C programs written for

one computer can be run on another with little or no

modification.

**Introduction**

Hospital Management is a new theory in management faculty.

Earlier a senior doctor used to perform the role of a hospital

manager. However, nowadays everything demands a specialist.

Almost all the things related to hospital have changed. Many

categories concerning medical sciences and hospital have altered

totally. There are various types of hospitals today, including

ordinary hospitals, specialty hospitals and super specialty

hospitals. The categories are regarding to the types of facilities

they offer to the people. Eligible professionals are needed for the

smooth operating of a hospital. Various courses and training

programs have been developed to find out eligible hospital

managers. Such professionals are well trained to solve the rising

challenges and specific necessities of modern day hospitals. The

Hospital Management courses are open to non-medical

background graduates also.

In recent times, health care concept of the people have undergone a

tremendous change which has lead to higher expectations and an

increased demand for high quality medical care and facilities. Here

comes the importance of Health/ Hospital Management. It is the

latest concept in the field of management and one of the most

lucrative and important careers in the health sector. According to a

recent US survey, Hospital management is one among the top ten

millennium professions which provides a direct link between

8

healthcare facilities and those supplying the services they need. A

career in Hospital Management involves ensuring effective

utilization of physical and financial resources of the hospital as

well as creating an organizational climate beneficial to the growth

and development of the personnel. It is a difficult, challenging,

responsible and more over, a demanding job.

A hospital manager is in a way responsible for administrative

dealings of the hospital. He accepts the charge of various aspects

of hospital management and health administration reverencing to

the patients and healthcare.

Professional services of trained manpower namely Hospital

administrators/ managers have now become essential to manage

hospitals and medical facilities. It is the hospital administrators/

managers who manage hospitals, outpatient clinics, hospices, drugabuse

treatment centers etc. They would be responsible for overall

patient care, education, research and community health care

associated with the organization, and make sure that they conform

to the rules of the Hospital. These people with the help of a team of

assistants look after the administrative affairs i.e. plan coordinate

and direct the delivery of health services within a hospital. In early

days, senior doctors used to perform the role of a hospital manager.

But now things have changed and the services of these qualified

professionals are required for the smooth functioning of hospitals.

The services of these professionals go a long way in making the

hospital more productive, profitable and comfortable for the

patients.

9

**Eligibility required for Hospital Management**

With the growing importance of hospital industry in providing

health care to people across the country, Hospital Management

courses are becoming very popular. An individual from a medical

or a non-medical background can pursue a career in Hospital

Administration/ Management. The pre-requisite is a graduation

(medical/ non-medical) or a degree/ diploma in Hospital

Administration. Bachelor of Hospital Administration is a threeyear

degree course for the undergraduate students. The students

should have finished their 10+2 with Biology with total 50%

marks, for BHA degree. Masters in Hospital Administration is a

two-year duration postgraduate course. This course is available for

both medical and non-medical background candidates. Graduates

in any stream can apply for this program.

**Personal skills required for Hospital Management**

A number of potentials are required for a candidate for this field.

The candidate must have rapid judgment capability related to upto-

date management doctrines and techniques. Friendly attitude,

ability to handle public and pressure, ability of handling deadlines,

brilliant communication skills and leadership skills are required to

an efficient hospital manager.

Career opportunities in the field of Hospital Management are

increasing very rapidly not only because of the growing

importance of hospital industry, but also with the difference in the

nature of work in hospitals. It is an apt career for those who have

an empathic nature along with a flair for organizing, and an eye for

detail. A Hospital Manager requires good knowledge of finance

and information systems, interpretation of data, and must be able to

keep up good communication with the staff of various

departments, and patients. They can work as Deans and Directors

of medical colleges, Hospital superintendents, Deputy

Superintendents, Medical directors, Nursing directors, Heads of

10

departments and as Administrators with substantial managerial

responsibilities. Hospital manager will be responsible for the

overall organization and management of the hospital to ensure its

smooth functioning with the objective of ensuring complete

satisfaction.

Job opportunities in the Hospital Management

There are plenty of job opportunities in the field of Hospital

Management. Assistant Hospital Administrator is the best option

for the fresh degree holders. They can start their career as

managers of non-health departments like finance. Anyone having

this degree can find job in large corporate and public zonal

hospitals, international and domestic healthcare institutes and

health insurance companies, nursing homes.

Top five Institutes in India

All India Institute of Medical Sciences, New Delhi.

Symbiosis Center of Healthcare.

Tata Institute of Social Science.

Armed Forces Medical College, Pune.

The Apollo Hospitals.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// PROJECT HOSPITAL MANAGEMENT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INCLUDED HEADER FILES

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <dos.h>

#include <ctype.h>

#include <process.h>

#include <math.h>

struct MENU

{

int st1, st2;

} ;

struct ROOM

{

int roomno, bedno ;

char status ;

} ;

13

struct PATIENT

{

int roomno, bedno, dd, mm, yy,age;

char name[26],address[36], phone[10], disease[15], doctor[26],sex;

} ;

struct DOCTOR

{

char name[26] ;

} ;

void LINE\_HOR(int, int, int, char) ;

void LINE\_VER(int, int, int, char) ;

void BOX(int, int, int, int, char) ;

void CLEARUP(int, int) ;

void CLEARDOWN(int, int) ;

void EDIT\_MENU() ;

int CREATE\_MENU(char\*[], int, int, int) ;

void ADMIT() ;

void DISCHARGE() ;

void MODIFY() ;

void PAT\_LIST() ;

14

void DISPLAY() ;

void DISPLAY\_RECORD(int, int) ;

void DELETE\_RECORD(int, int) ;

void ADD\_ROOMS() ;

int AVAILABLE(int, int) ;

int EMPTY(int) ;

void CHANGE\_STATUS(int, int, char) ;

void ROOM\_LIST() ;

int ROOMNO() ;

int BEDNO(int) ;

int LAST\_ROOMNO() ;

int RECORDNO(int, int) ;

void ADD\_DOC() ;

void DOC\_LIST() ;

void DELETE() ;

char \*DOCTOR\_NAME(int) ;

int RECORDS() ;

void DEFAULT() ;

void main()

{

void MAIN\_MENU() ;

MAIN\_MENU() ;

}

15

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE HORRIZONTAL LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINE\_HOR(int column1, int column2, int row, char c)

{

int col;

for ( col=column1; col<=column2; col++ )

{

gotoxy(col,row) ;

printf("%c",c) ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE VERTICAL LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LINE\_VER(int row1, int row2, int column, char c)

{

int r;

for ( r=row1; r<=row2; r++ )

{

gotoxy(column,r) ;

printf("%c",c) ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

16

// THIS FUNCTION DRAWS THE BOX

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void BOX(int column1, int row1, int column2, int row2, char c)

{

char ch=218 ;

char c1, c2, c3, c4 ;

char l1=196, l2=179 ;

if (c == ch)

{

c1=218 ;

c2=191 ;

c3=192 ;

c4=217 ;

l1 = 196 ;

l2 = 179 ;

}

else

{

c1=c ;

c2=c ;

c3=c ;

c4=c ;

l1 = c ;

l2 = c ;

}

gotoxy(column1,row1) ;

printf("%c",c1) ;

gotoxy(column2,row1) ;

printf("%c",c2) ;

gotoxy(column1,row2) ;

17

printf("%c",c3) ;

gotoxy(column2,row2) ;

printf("%c",c4) ;

column1++ ;

column2-- ;

LINE\_HOR(column1,column2,row1,l1) ;

LINE\_HOR(column1,column2,row2,l1) ;

column1-- ;

column2++ ;

row1++ ;

row2-- ;

LINE\_VER(row1,row2,column1,l2) ;

LINE\_VER(row1,row2,column2,l2) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CLEAR THE SCREEN LINE BY LINE UPWARD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CLEARUP(int start, int end)

{

int i;

for (i=start; i>=end; i--)

{

delay(20) ;

gotoxy(1,i) ; clreol() ;

}

}

18

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CLEAR THE SCREEN LINE BY LINE DOWNWORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void CLEARDOWN(int start, int end)

{

int i;

for (i=start; i<=end; i++)

{

delay(20) ;

gotoxy(1,i) ; clreol() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CONTROL ALL THE FUNCTIONS IN THE MAIN MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void MAIN\_MENU()

{

int ch=0 ;

char \*options[7]={

" ADMISSION ",

" DISCHARGE ",

" ROOM STATUS ",

" PATIENT LIST ",

" PATIENT RECORD ",

" EDIT ",

" QUIT "

19

} ;

while (1)

{

textmode(C40) ;

clrscr() ;

textbackground(WHITE) ;

textcolor(BLACK) ;

gotoxy(13,8) ;

cprintf("O P T I O N S ") ;

textbackground(BLACK) ;

textcolor(LIGHTGRAY) ;

ch = CREATE\_MENU(options,7,12,11) ;

textmode(C80) ;

clrscr() ;

switch(ch)

{

case 0:

ADMIT() ;

break;

case 1:

DISCHARGE() ;

break;

case 2:

ROOM\_LIST() ;

break;

case 3:

PAT\_LIST() ;

20

break;

case 4:

DISPLAY() ;

break;

case 5:

EDIT\_MENU() ;

break;

case 6:

return;

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CONTROL ALL THE FUNCTIONS IN THE EDIT MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void EDIT\_MENU()

{

int ch=0 ;

char \*options[5]={

" ADD ROOMS ",

" MODIFY PATIENT RECORD ",

" ADD DOCTOR RECORD ",

" DELETE DOCTOR RECORD ",

21

" RETURN "

} ;

while (1)

{

textmode(C40) ;

clrscr() ;

textbackground(WHITE) ;

textcolor(BLACK) ;

gotoxy(13,8) ;

cprintf(" EDIT MENU ") ;

textbackground(BLACK) ;

textcolor(LIGHTGRAY) ;

ch = CREATE\_MENU(options,5,10,11) ;

textmode(C80) ;

clrscr() ;

if (ch == 0)

{

ADD\_ROOMS() ;

}

else

if (ch == 1)

{

MODIFY() ;

}

else

if (ch == 2)

{

ADD\_DOC() ;

}

else

22

if (ch == 3)

{

DELETE() ;

}

else

if (ch == 4)

break ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CREATE MENU FOR THE GIVEN OPTIONS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int CREATE\_MENU(char \*options[], int size, int y, int x)

{

struct MENU mnu;

int valid ;

int i,t,n = 0 ;

char ch1, ch2 ;

char chr1=219 ;

char c1=24, c2=25 ;

mnu.st1 = 12 ;

mnu.st2 = 8 ;

gotoxy(3,24) ;

printf("%c",c1) ;

printf("%c",c2) ;

printf("=Move <ENTER>=Select <ESC>=Exit" );

gotoxy(mnu.st1,mnu.st2) ;

23

printf("%c",chr1) ;

t = x ;

for (i=0; i<size; i++)

{

gotoxy(y,t) ;

printf("%s",options[i]) ;

t++ ;

}

textbackground(WHITE) ; textcolor(BLACK) ;

gotoxy(y,x) ;

cprintf("%s",options[n]) ;

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

gotoxy(mnu.st1,mnu.st2) ;

do

{

do

{

do

{

ch1 = getch() ;

if ( ch1 == 27 )

return(size-1) ;

} while (( ch1 != 0 ) && ( ch1 != 13 )) ;

if ( ch1 != 13 )

{

do

{

ch2 = getch() ;

if ( ch2 == 27 )

return(size-1) ;

} while ((ch2 != 72) && (ch2 != 80) && (ch2 != 13)) ;

24

}

} while (((ch1 != 0) || ((ch2 != 72) && (ch2 != 80))) && ((ch1 != 13) &&

(ch2 != 13))) ;

if ((ch1 == 0) && (ch2 == 80))

{

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

gotoxy(y,x) ;

cprintf("%s",options[n]) ;

if (n == size-1)

{

n = 0 ;

x = x - (size-1) ;

}

else

{

n++ ;

x++ ;

}

textbackground(WHITE) ; textcolor(BLACK) ;

gotoxy(y,x) ;

cprintf("%s",options[n]) ;

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

gotoxy(mnu.st1,mnu.st2) ;

}

if ((ch1 == 0) && (ch2 == 72))

{

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

gotoxy(y,x) ;

cprintf("%s",options[n]) ;

if ( n == 0 )

{

25

n = size-1 ;

x = x + (size-1) ;

}

else

{

n-- ;

x-- ;

}

textbackground(WHITE) ; textcolor(BLACK) ;

gotoxy(y,x) ;

cprintf("%s",options[n]) ;

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

gotoxy(mnu.st1,mnu.st2) ;

}

} while (( ch1 != 13 ) && ( ch2 != 13 )) ;

return n ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS THE CODE OF LAST ROOM NO.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int LAST\_ROOMNO()

{

FILE \*fp ;

int last,n;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

26

fclose(fp) ;

return 0 ;

}

last=0 ;

while(fread(&r, sizeof(r),1,fp)!=NULL)

last++ ;

fclose(fp);

return (last/2) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ADDS ROOMS TO THE FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void ADD\_ROOMS()

{

char ch ;

char p1,p2,p3;

int rno,bno,i;

FILE \*fp;

struct ROOM room;

clrscr() ;

gotoxy(5,5) ; clreol() ;

printf("Enter Password:");

p1=getch();

printf("\*");

p2=getch();

printf("\*");

27

p3=getch();

printf("\*");

if(p1=='a' && p2=='b' && p3=='c')

{

clrscr();

do

{

fp=fopen("RNO.dat","ab");

if(fp==NULL)

{

printf("\n\t\tFile Not Found\n");

return;

}

rno = LAST\_ROOMNO() + 1 ;

bno=1 ;

for (i=1; i<=2; i++)

{

room.roomno = rno ;

room.bedno = bno ;

room.status = 'A' ;

fwrite(&room,sizeof(room),1,fp);

bno = 2 ;

}

fclose(fp);

BOX(8,9,42,15,218) ;

gotoxy(10,10) ;

printf("Room no. :%d ",rno) ;

gotoxy(10,11) ;

28

printf("Total no. of beds : 2" );

gotoxy(10,12) ;

printf("STATUS:");

gotoxy(20,13) ;

printf("Bed no. 1 : Available");

gotoxy(20,14) ;

printf("Bed no. 2 : Available");

gotoxy(5,24) ;

printf("\7Room Added") ;

gotoxy(5,25) ;

printf("Press any key to continue...");

getch() ;

clrscr();

printf("Do you want to add rooms (y/n) ");

ch = getche() ;

ch = toupper(ch) ;

} while (ch == 'Y' || ch == 'y') ;

}

else

return;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS 0 IF GIVEN ROOM IS NOT EMPTY

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int EMPTY(int rno)

{

int empty=1 ;

29

FILE \*fp ;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return 0 ;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (r.roomno == rno && r.status == 'N')

{

empty = 0 ;

break ;

}

}

fclose(fp);

return empty ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS 0 IF GIVEN BED NO. OF THE ROOM IS

// NOT AVAILABLE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int AVAILABLE(int rno, int bno)

{

int avail=0 ;

30

FILE \*fp ;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return 0 ;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (r.roomno == rno && r.bedno == bno && r.status == 'A')

{

avail = 1 ;

break ;

}

}

fclose(fp);

return avail ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS RECORD NO. OF THE GIVEN ROOM AND

// BED NO.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int RECORDNO(int rno, int bno)

{

int recordno=0 ;

31

FILE \*fp ;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return 0 ;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

recordno++ ;

if (r.roomno == rno && r.bedno == bno)

break ;

}

fclose(fp);

return recordno ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAY THE LIST OF THE ROOMS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void ROOM\_LIST()

{

int row = 3, flag ;

int sta = 0 ;

int room=0,bed=0;

char ch ;

FILE \*fp ;

32

struct ROOM r;

clrscr() ;

textbackground(WHITE) ;

textcolor(BLACK) ;

gotoxy(1,1) ;

cprintf("Room no. Bed no. Status Room no. Bed no. Status

") ;

textbackground(BLACK) ;

textcolor(LIGHTGRAY) ;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

flag = 0 ;

delay(20) ;

gotoxy(2,row) ;

printf("%d",r.roomno) ;

gotoxy(12,row) ;

printf("%d",r.bedno) ;

gotoxy(23,row) ;

printf("%c",r.status) ;

if (r.status == 'A')

sta++ ;

if(fread(&r, sizeof(r),1,fp)!=NULL)

{

33

gotoxy(41,row) ;

printf("%d",r.roomno) ;

gotoxy(51,row) ;

printf("%d",r.bedno) ;

gotoxy(62,row) ;

printf("%c",r.status) ;

if (r.status == 'A')

sta++ ;

}

if (row == 22)

{

flag = 1 ;

gotoxy(1,25) ;

printf("Press any key to continue..." );

ch = getch() ;

if (ch == 27)

break ;

row = 3 ;

}

else

row++ ;

room++;bed+=2;

}

if (!flag)

{

gotoxy(1,25) ;

printf("Press any key to continue...");

getch() ;

}

CLEARDOWN(3,25) ;

gotoxy(1,16) ;

34

printf("Total room :%d " ,room) ;

gotoxy(1,17) ;

printf("Total occupied beds :%d ",bed-sta) ;

textbackground(WHITE) ;

textcolor(BLACK) ;

gotoxy(1,18) ;

cprintf("Total available beds : %d",sta) ;

textbackground(BLACK) ;

textcolor(LIGHTGRAY) ;

gotoxy(1,25) ;

printf("Press any key to continue...") ;

getch() ;

CLEARUP(25,1) ;

}

int ROOMNO()

{

int rno=0 ;

FILE \*fp ;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return 0 ;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (r.status == 'A')

35

{

rno = r.roomno ;

break ;

}

}

fclose(fp);

return rno ;

}

int BEDNO(int rno)

{

int bno=0 ;

FILE \*fp ;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return 0 ;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (rno == r.roomno && r.status == 'A')

{

bno = r.bedno ;

break ;

}

}

fclose(fp);

36

return bno ;

}

void CHANGE\_STATUS(int rno, int bno, char rstatus)

{

FILE \*fp,\*temp ;

struct ROOM r;

fp=fopen("RNO.DAT", "r+") ;

temp=fopen("TEMP.DAT", "w+") ;

if (fp==NULL)

{

fclose(fp) ;

return;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (rno == r.roomno && bno == r.bedno)

{

r.status = rstatus ;

fwrite(&r,sizeof(r),1,temp) ;

}

else

fwrite(&r,sizeof(r),1,temp) ;

}

fclose(fp) ;

fclose(temp) ;

37

fp=fopen("RNO.DAT", "w+") ;

temp=fopen("TEMP.DAT", "r+") ;

while(fread(&r, sizeof(r),1,temp)!=NULL)

fwrite(&r,sizeof(r),1,fp) ;

fclose(fp) ;

fclose(temp) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ADD NAME OF DOCTOR IN THE FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void ADD\_DOC()

{

char ch ;

int valid ;

struct DOCTOR doc;

FILE \*fp ;

while(1)

{

do

{

clrscr() ;

gotoxy(5,5) ; clreol() ;

printf("Do you want to add Doctors (y/n) " );

38

ch = getche() ;

ch = toupper(ch) ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

clrscr() ;

DOC\_LIST() ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("Enter the name of the New Doctor" );

gotoxy(5,7) ; clreol() ;

printf("Name : " );

gets(doc.name) ;

strupr(doc.name) ;

if (strlen(doc.name) < 1 || strlen(doc.name) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7Enter correctly (Range: 1..25)" );

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

fp=fopen("DT.dat", "a") ;

if (fp==NULL)

{

fclose(fp) ;

return;

}

39

fwrite(&doc,sizeof(doc),1,fp);

fclose(fp);

DOC\_LIST() ;

gotoxy(5,24) ;

printf("Doctor Name Added" );

gotoxy(5,25) ;

printf("Press any key to continue...") ;

getch() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ADD A NAME OF DOCTOR AS A DEFAULT RECORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void DEFAULT()

{

struct DOCTOR doc;

FILE \*fp ;

strcpy(doc.name,"KUMAR PUSHKAR");

fp=fopen("DT.dat", "a+") ;

fwrite(&doc,sizeof(doc),1,fp);

fclose(fp);

}

40

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAY THE LIST OF THE DOCTORS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void DOC\_LIST()

{

int row = 13, col=2, sno=1 ;

char ch ;

struct DOCTOR r;

FILE \*fp ;

textbackground(WHITE) ;

textcolor(BLACK) ;

gotoxy(1,11) ; clreol() ;

cprintf("S.No. Name ") ;

textbackground(BLACK) ;

textcolor(LIGHTGRAY) ;

fp=fopen("DT.dat", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

delay(20) ;

gotoxy(col,row) ;

printf("%d",sno) ;

gotoxy(col+6,row) ;

41

printf("%s",r.name) ;

sno++ ;

if (row == 22)

{

row = 14 ;

col = 40 ;

}

else

{

row++ ;

}

}

fclose(fp);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS NO. OF RECORDS IN THE DOCTOR'S FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int RECORDS()

{

int records=0 ;

FILE \*fp ;

struct DOCTOR r;

fp=fopen("DT.dat", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return 0 ;

42

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

records++ ;

fclose(fp);

return records ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS NAME OF THE DOCTOR

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char \*DOCTOR\_NAME(int pd)

{

int i=0 ;

FILE \*fp ;

struct DOCTOR r;

fp=fopen("DT.dat", "r+") ;

if (fp==NULL)

{

fclose(fp) ;

return '\0' ;

}

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

i++ ;

if (i == pd)

break ;

43

}

fclose(fp);

return r.name ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DELETES DOCTOR RECORD IN DOCTOR FILE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void DELETE()

{

char t1[5] ;

int t2, sno=1 ;

FILE \*fp,\*temp ;

struct DOCTOR r;

clrscr() ;

DOC\_LIST() ;

do

{

gotoxy(5,5) ; clreol() ;

printf("Enter S.No. of the Doctor to be Deleted(press 0 to EXIT) ") ;

gets(t1) ;

t2 = atof(t1) ;

if (t1[0] == '0')

return ;

} while (sno < 1 || sno > RECORDS()) ;

44

fp=fopen("DT.dat", "r+") ;

temp=fopen("TEMP.DAT", "w+") ;

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (t2 != sno)

fwrite(&r,sizeof(r),1,temp) ;

sno++;

}

fclose(fp) ;

fclose(temp) ;

fp=fopen("DT.dat", "w+") ;

temp=fopen("TEMP.DAT", "r+") ;

while(fread(&r, sizeof(r),1,temp)!=NULL)

fwrite(&r,sizeof(r),1,fp) ;

fclose(fp) ;

fclose(temp) ;

CLEARDOWN(13,23) ;

DOC\_LIST() ;

gotoxy(5,7) ;

printf("\7Record Deleted") ;

gotoxy(5,25) ;

printf("Press any key to continue...") ;

getch() ;

}

45

void ADMIT()

{

int t2, rno, bno, page, pd;

char t1[5], pname[26], paddress[36], psex,pphone[10], pdisease[15], pdoctor[26],

ch ;

int d1, m1, y1, valid ;

struct ROOM r ;

struct date d;

struct PATIENT pat;

FILE \*fp;

rno = ROOMNO() ;

if (rno == 0)

{

gotoxy(5,10) ;

printf("\7Sorry no. room is available for the Patient" );

getch() ;

return ;

}

bno = BEDNO(rno) ;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

textcolor(BLACK) ; textbackground(WHITE) ;

gotoxy(5,1) ;

cprintf("Date: %d/%d/%d",d1,m1,y1) ;

gotoxy(60,1) ;

cprintf("Room no.: %d",rno) ;

gotoxy(60,2) ;

cprintf("Bed no. : %d",bno) ;

46

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

gotoxy(5,3) ;

printf("Name : " );

gotoxy(5,4) ;

printf("Address : " );

gotoxy(5,5) ;

printf("Phone : " );

gotoxy(5,6) ;

printf("Age : " );

gotoxy(5,7) ;

printf("Sex : " );

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER NAME OF THE PATIENT" );

gotoxy(15,3) ; clreol() ;

gets(pname) ;

strupr(pname) ;

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

return ;

if (strlen(pname) < 1 || strlen(pname) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7Enter correctly (Range: 1..25)" );

getch() ;

}

} while (!valid) ;

do

{

47

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER ADDRESS OF THE PATIENT") ;

gotoxy(15,4) ; clreol() ;

gets(paddress) ;

strupr(paddress) ;

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

return ;

if (strlen(paddress) < 1 || strlen(paddress) > 35)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7Enter correctly (Range: 1..35)") ;

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER PHONE NO. OF THE PATIENT or <ENTER> FOR

NONE" );

gotoxy(15,5) ; clreol() ;

gets(pphone) ;

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

return ;

if ((strlen(pphone) < 7 && strlen(pphone) > 0) || (strlen(pphone) > 9))

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7Enter correctly") ;

48

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER AGE OF THE PATIENT") ;

gotoxy(15,6) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

page = t2 ;

if (t1[0] == '0')

return ;

if (page < 1 || page > 100)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

do

{

gotoxy(5,25) ; clreol() ;

printf("ENTER SEX OF THE PATIENT (M/F)");

gotoxy(15,7) ; clreol() ;

psex = getche() ;

psex = toupper(psex) ;

} while (psex != 'M' && psex != 'F') ;

do

49

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER DISEASE OF THE PATIENT") ;

gotoxy(5,8) ; clreol() ;

printf("Disease : ") ;

gets(pdisease) ;

strupr(pdisease) ;

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

return ;

if (strlen(pdisease) < 1 || strlen(pdisease) > 15)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7Enter correctly (Range: 1..15)") ;

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

DOC\_LIST() ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

gotoxy(5,24) ; clreol() ;

printf("ENTER S.No. OF THE DOCTOR ");

gets(t1) ;

t2 = atof(t1) ;

pd = t2 ;

if (t1[0] == '0')

50

return ;

if (pd < 1 || pd > RECORDS())

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

strcpy(pdoctor,DOCTOR\_NAME(pd)) ;

CLEARUP(25,10) ;

gotoxy(5,10) ;

printf("Doctor Assigned : Dr.%s" ,pdoctor);

do

{

gotoxy(5,13) ; clreol() ;

printf("Do you want to save (y/n) ") ;

ch = getche() ;

ch = toupper(ch) ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

CHANGE\_STATUS(rno,bno,'N') ;

pat.roomno = rno ;

pat.bedno = bno ;

strcpy(pat.name,pname) ;

strcpy(pat.address,paddress) ;

strcpy(pat.phone,pphone) ;

pat.age = page ;

pat.sex=psex;

pat.dd = d1 ;

51

pat.mm = m1 ;

pat.yy = y1 ;

strcpy(pat.disease,pdisease) ;

strcpy(pat.doctor,pdoctor) ;

fp=fopen("PT.DAT","r+");

fwrite(&pat,sizeof(pat),1,fp);

fclose(fp);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAYS THE LIST OF THE PATIENTS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void PAT\_LIST()

{

int row = 5 , found=0, flag=0 ;

char ch ;

FILE \*fp;

struct PATIENT pat;

clrscr() ;

textcolor(BLACK+BLINK) ; textbackground(WHITE) ;

gotoxy(31,1) ;

cprintf(" LIST OF PATIENTS ") ;

textcolor(BLACK) ;

gotoxy(1,3) ; clreol() ;

gotoxy(1,3) ;

printf("ROOM NO. BED NO. PATIENT'S NAME") ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

52

fp=fopen("PT.DAT", "r+") ;

while (fread(&pat,sizeof(pat),1,fp))

{

flag = 0 ;

delay(20) ;

found = 1 ;

gotoxy(3,row) ;

printf("%d",pat.roomno) ;

gotoxy(15,row) ;

printf("%d",pat.bedno) ;

gotoxy(28,row) ;

printf("%s",pat.name) ;

if ( row == 23 )

{

flag = 1 ;

row = 5 ;

gotoxy(1,25) ;

printf("Press any key to continue or Press <ESC> to exit") ;

ch = getch() ;

if (ch == 27)

break ;

clrscr() ;

textcolor(BLACK) ; textbackground(WHITE) ;

gotoxy(31,1) ;

printf(" LIST OF PATIENTS ") ;

gotoxy(1,3) ; clreol() ;

gotoxy(1,3) ;

printf("ROOM NO. BED NO. PATIENT'S NAME" );

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

}

53

else

row++ ;

}

if (!found)

{

gotoxy(5,10) ;

printf("\7Records not found") ;

}

if (!flag)

{

gotoxy(1,25) ;

printf("Press any key to continue..." );

getche() ;

}

fclose (fp) ;

}

void DISPLAY\_RECORD(int rno, int bno)

{

FILE \*fp ;

struct PATIENT pat;

fp=fopen("PT.DAT", "r+") ;

while (fread(&pat,sizeof(pat),1,fp))

{

if (rno == pat.roomno && bno == pat.bedno)

{

textcolor(BLACK) ; textbackground(WHITE) ;

gotoxy(5,1) ;

cprintf("Date of Admit : %d/%d/%d",pat.dd,pat.mm,pat.yy) ;

gotoxy(60,1) ;

54

cprintf("Room no.: %d",pat.roomno) ;

gotoxy(60,2) ;

cprintf("Bed no. : %d",pat.bedno) ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

gotoxy(5,3) ;

printf("Name :%s ",pat.name) ;

gotoxy(5,4) ;

printf("Address : %s",pat.address) ;

gotoxy(5,5) ;

printf("Phone :%s ",pat.phone) ;

gotoxy(5,6) ;

printf("Age :%d ",pat.age) ;

gotoxy(5,7) ;

printf("Sex :%c",pat.sex) ;

gotoxy(5,8) ;

printf("Disease : %s",pat.disease) ;

gotoxy(5,10) ;

printf("Doctor Assigned : Dr.%s",pat.doctor) ;

break ;

}

}

fclose(fp) ;

}

void DISPLAY()

{

char t1[5], ch ;

int rno, bno, valid=0, t2 ;

clrscr() ;

do

55

{

valid = 1 ;

gotoxy(5,5) ;

printf("Room no. : ") ;

gotoxy(5,7) ;

printf("Bed no. : ") ;

gotoxy(5,25) ; clreol() ;

printf("ENTER BED NO. OF THE PATIENT or <ENTER> FOR

HELP") ;

gotoxy(16,5) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

rno = t2 ;

if (t1[0] == '0')

return ;

if (strlen(t1) == 0)

{

valid = 0 ;

ROOM\_LIST() ;

clrscr() ;

}

if (rno < 1 && valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

do

{

56

valid = 1 ;

gotoxy(5,5) ;

printf("Room no. :%d ",rno) ;

gotoxy(5,7) ;

printf("Bed no. : ") ;

gotoxy(5,25) ; clreol() ;

printf("ENTER BED NO. OF THE PATIENT or <ENTER> FOR

HELP") ;

gotoxy(16,7) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

bno = t2 ;

if (t1[0] == '0')

return ;

if (strlen(t1) == 0)

{

valid = 0 ;

ROOM\_LIST() ;

clrscr() ;

}

if ((bno < 1 || bno > 2) && valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

if (AVAILABLE(rno,bno))

{

57

gotoxy(5,20) ;

printf("\7Sorry, there is no. Patient in this Room no.") ;

getch() ;

return ;

}

clrscr() ;

DISPLAY\_RECORD(rno,bno) ;

gotoxy(5,25) ;

printf("Press any key to continue...") ;

getch() ;

}

void DELETE\_RECORD(int rno, int bno)

{

FILE \*fp,\*temp ;

struct PATIENT r;

fp=fopen("PT.DAT", "r+") ;

temp=fopen("TEMP.DAT", "w+") ;

while(fread(&r, sizeof(r),1,fp)!=NULL)

{

if (rno != r.roomno || bno != r.bedno)

fwrite(&r,sizeof(r),1,temp) ;

}

fclose(fp) ;

fclose(temp) ;

58

fp=fopen("PT.DAT", "r+") ;

temp=fopen("TEMP.DAT", "r+") ;

while(fread(&r, sizeof(r),1,temp)!=NULL)

fwrite(&r,sizeof(r),1,fp) ;

fclose(fp) ;

fclose(temp) ;

}

void DISCHARGE()

{

char t1[5], ch ;

int rno, bno, valid, t2 ;

clrscr() ;

do

{

valid = 1 ;

gotoxy(5,5) ;

printf("Room no. : " );

gotoxy(5,7) ;

printf("Bed no. : " );

gotoxy(5,25) ; clreol() ;

printf("ENTER ROOM NO. OF THE PATIENT or <ENTER> FOR

HELP") ;

gotoxy(16,5) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

rno = t2 ;

if (t1[0] == '0')

59

return ;

if (strlen(t1) == 0)

{

valid = 0 ;

ROOM\_LIST() ;

clrscr() ;

}

if ((rno < 1 || rno > LAST\_ROOMNO()) && valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY" );

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,5) ;

printf("Room no. : ",rno) ;

gotoxy(5,7) ;

printf("Bed no. : ") ;

gotoxy(5,25) ; clreol() ;

printf("ENTER BED NO. OF THE PATIENT or <ENTER> FOR

HELP") ;

gotoxy(16,7) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

bno = t2 ;

if (t1[0] == '0')

return ;

60

if (strlen(t1) == 0)

{

valid = 0 ;

ROOM\_LIST() ;

clrscr() ;

}

if ((bno < 1 || bno > 2) && valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY" );

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

if (AVAILABLE(rno,bno))

{

gotoxy(5,20) ;

printf("\7Sorry, there is no. Patient in this Room no.") ;

getch() ;

return ;

}

clrscr() ;

DISPLAY\_RECORD(rno,bno) ;

do

{

gotoxy(5,13) ; clreol() ;

printf("Discharge this Patient (y/n) ") ;

ch = getche() ;

ch = toupper(ch) ;

} while (ch != 'Y' && ch != 'N') ;

61

if (ch == 'N')

return ;

CHANGE\_STATUS(rno,bno,'A') ;

DELETE\_RECORD(rno,bno) ;

CLEARDOWN(1,22) ;

gotoxy(5,23) ;

printf("\7Record Deleted") ;

gotoxy(5,25) ;

printf("Press any key to continue...") ;

getch() ;

}

void MODIFY()

{

char t1[5], ch ,doctor[35];

int rno, bno, rno1,bno1,valid, t2 ;

int pd;

FILE \*fp,\*temp ;

struct PATIENT pat;

clrscr() ;

do

{

valid = 1 ;

gotoxy(5,5) ;

printf("Room no. : ") ;

gotoxy(5,7) ;

printf("Bed no. : ") ;

gotoxy(5,25) ; clreol() ;

62

printf("ENTER ROOM NO. OF THE PATIENT or <ENTER> FOR

HELP") ;

gotoxy(16,5) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

rno = t2 ;

if (t1[0] == '0')

return ;

if (strlen(t1) == 0)

{

valid = 0 ;

ROOM\_LIST() ;

clrscr() ;

}

if ((rno < 1 || rno > LAST\_ROOMNO()) && valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,5) ;

printf("Room no. :%d ",rno) ;

gotoxy(5,7) ;

printf("Bed no. : ") ;

gotoxy(5,25) ; clreol() ;

63

printf("ENTER BED NO. OF THE PATIENT or <ENTER> FOR

HELP") ;

gotoxy(16,7) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

bno = t2 ;

if (t1[0] == '0')

return ;

if (strlen(t1) == 0)

{

valid = 0 ;

ROOM\_LIST() ;

clrscr() ;

}

if ((bno < 1 || bno > 2) && valid)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

if (AVAILABLE(rno,bno))

{

gotoxy(5,20) ;

printf("\7Sorry, there is no. Patient in this Room no.") ;

getch() ;

return ;

}

clrscr() ;

64

DISPLAY\_RECORD(rno,bno) ;

do

{

gotoxy(5,13) ; clreol() ;

printf("Modify this Patient Record (y/n) ") ;

ch = getche() ;

ch = toupper(ch) ;

} while (ch != 'Y' && ch != 'N') ;

if (ch == 'N')

return ;

do

{

valid = 1 ;

clrscr() ;

gotoxy(29,1) ;

printf("MODIFY PATIENT RECORD") ;

gotoxy(29,2) ;

printf("~~~~~~~~~~~~~~~~~~~~~") ;

gotoxy(5,4) ;

printf("Room no. : ") ;

gotoxy(5,6) ;

printf("Bed no. : ") ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER ROOM NO. OF THE PATIENT") ;

gotoxy(16,4) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

rno1 = t2 ;

65

if (t1[0] == '0')

return ;

if (rno1 < 1 || rno1 > LAST\_ROOMNO())

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY" );

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

printf("ENTER BED NO. OF THE PATIENT") ;

gotoxy(16,6) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

bno1 = t2 ;

if (t1[0] == '0')

return ;

if (bno1 < 1 || bno1 > 2)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

if (!AVAILABLE(rno1,bno1))

66

{

valid = 0 ;

gotoxy(5,20) ;

printf("\7Sorry, there is already a Patient in this Room no." );

getch() ;

}

} while (!valid) ;

DOC\_LIST() ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

gotoxy(5,24) ; clreol() ;

printf("ENTER S.No. OF THE DOCTOR ") ;

gets(t1) ;

t2 = atof(t1) ;

pd = t2 ;

if (t1[0] == '0')

return ;

if (pd < 1 || pd > RECORDS())

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

printf("\7ENTER CORRECTLY") ;

getch() ;

}

} while (!valid) ;

strcpy(doctor,DOCTOR\_NAME(pd)) ;

CLEARUP(25,10) ;

gotoxy(5,8) ;

printf("Doctor Assigned : Dr.%s",doctor) ;

67

CHANGE\_STATUS(rno,bno,'A') ;

CHANGE\_STATUS(rno1,bno1,'N') ;

fp=fopen("PT.DAT", "r+") ;

temp=fopen("TEMP.DAT", "w+") ;

while(fread(&pat, sizeof(pat),1,fp)!=NULL)

{

if (rno == pat.roomno && bno == pat.bedno)

{

pat.roomno = rno1 ;

pat.bedno = bno1 ;

strcpy(pat.doctor,doctor);

fwrite(&pat,sizeof(pat),1,temp) ;

}

else

fwrite(&pat,sizeof(pat),1,temp) ;

}

fclose(fp) ;

fclose(temp) ;

fp=fopen("PT.DAT", "r+") ;

temp=fopen("TEMP.DAT", "r+") ;

while(fread(&pat, sizeof(pat),1,temp)!=NULL)

fwrite(&pat,sizeof(pat),1,fp) ;

fclose(fp) ;

fclose(temp) ;

clrscr();

68

DISPLAY\_RECORD(rno1,bno1);

gotoxy(5,23) ;

printf("\7Record Modified") ;

gotoxy(5,25) ;

printf("Press any key to continue...") ;

getch() ;

}

69

**OUTPUT OF THE PROJECT**

**ADMISSION**

70

**DISCHARGE**

71

**ROOM STATUS**

**PATIENT LIST**

**PATIENT RECORD**

72

**EDIT MENU**

**ADD ROOMS**

73

**MODIFY PATIENT RECORD**

74

**ADD DOCTOR**

75

**DELETE DOCTOR RECORD**

**BIBLIOGRAPHY**

**BOOK**

“ C IS MAGIC ", by Narender rajput.

**WEB SITE**

"http://www.webindia123.com/career/options/hospitalmanagement/

intro.htm"

"http://www.indiaedu.com/articles/hospital-management.html"