

PROGRAM FOR PROBLEM SOLVING (PPS)

MINI PROJECT

TOPIC

HOSTEL MANAGEMENT SYSTEM

Submitted by –

- > Deepayan Bhattacharyya (RA211003010306)
- > Tansu Gangopadhyay (RA2111003010334)
- > Dept CSE CORE
- > Section - E1

AIM-To create a **HOSTEL MANAGEMENT SYSTEM** using c language

WHAT IS HOSTEL MANAGEMENT SYSTEM ?

Checking the rooms' availability, allocating room to new students, managing hostel fees, visitors' records, mess, and all such hostel management related activities manually consume efforts, time, and resources. Moreover, hostel management also requires looking after the security of students which can only be achieved with close supervision of daily students' activities.

Hostel management system is designed to manage all hostel activities like hostel admissions, fees, room, mess allotment, hostel stores & generates related reports for smooth transactions. It is also used to manage monthly mess bill calculation, hostel staff payroll, student certificates, etc.

Designed & developed keeping in mind all the hostel management concerns of staff, MasterSoft's Hostel Management Software is a one-stop-solution for higher education institutions. Right from students' registration to allocating rooms & mess, managing student count records, and ins-and-outs of students, it streamlines & automates all the day-to-day hostel activities

It also ensures the following details to be fed for effective management.

- > Hostel infrastructure
- > Mess creation and management
- > Registered students
- > Merit lists
- > Room and mess allotment
- > Mess bill calculation

BENEFITS OF HOSTEL MANAGEMENT SOFTWARE

Systematic Control Over Hostel Activities

Verifies the students' applications & creates a merit list on the basis of pre-defined hostel room allocation rules and helps the staff to allocate rooms & wings. Thus, provides systematic control over hostel processes.

Accurate Student Data Management

Creates a centralized database of student information including their personal information such as name, address, guardian's mobile number, etc. The information can be accessed by the hostel warden or any other authority at any given point in time.

Complete Student Security

Keeps tab on students in-and-out time, name & other details of their visitors, student attendance, and count thus induces a disciplined life & ensures transparency & complete student security in the hostel campus.

FEATURES OF HOSTEL MANAGEMENT SYSTEM

Online Admission & Fees Management

The software entirely automates the student registration & fees collection process. With the hostel management software login portal, students from any geographical region can apply to avail hostel facilities online.

Room Allocation & Transfers

The staff can define room allocation rules and allot the vacant rooms to students depending on their requirements in real-time. The student hostel management software also enables the wardens to transfer students' rooms in case of any discrepancy.

Alerts & Notifications

The SMS & email notification facility integrated into the hostel management app allows the hostel authorities to send alerts, student attendance history reports, mess bills, fees-related information, etc to students & parents on their registered number & email ids.

Mess Management

The software auto-calculates & maintains the record of hostel bills, mess bills, any type of fines, dues, etc, and helps in managing the hostel efficiently. This module can be integrated with the college ERP or university ERP software for faster & quicker student fees management.

Paperless Student Attendance

Enables institutions to avoid wrong attendance details by replacing the manual attendance with the paperless student attendance system module. With the hostel software, student's attendance can be made quick and errorless

Visitor Management Records

The hostel management software for higher education institutions manages the records of visitors by keeping a track of their vehicle number, their intent or purpose to visit the hostel, their relation with the student, and thus, ensures only authorized people get entry into the hostel premises.

Hostel Reports

The hostel staff & authorities can generate the following reports with the hostel management software.

Room wise list of students

vacant room report

Hostel ID cards

Attendance report

Student address labels

Hostel certificates

Mess Bills & balance payment

Payroll register

Financial accounting reports

Stock register

Fine report

Students complete information

Dues & Refund report at the end of semester and year

CODE

/including header files

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

//defining global variables

```
int i,j,main_exit;
```

//declaring functions

void menu();

void new_entry();

void edit();

void view_list();

void erase();

void see();

void sorter();

void closer();


```
//defining structures
struct date{
    int month,day,year;
};
struct {

    char name[60];
    int age;
    char city[60];
    char mail_id[30];
    char room_no[8];
    char father_name[60];
    int father_phone;
    int phone;
    char hostel_name[6];
    char roll_no[10];
    struct date dob;
}add,upd,check,rem,sort;
```

```
//-----  
//main function  
int main()  
{  
    pass();  
    menu();  
    return 0;  
}
```

```
//menu function  
void menu()  
{  
system("CLS");  
    int choice;  
    printf("\n\n\t\t\tHOSTEL MANAGEMENT SYSTEM");  
    printf("\n\n\n\t\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME TO THE MAIN MENU  
\xB2\xB2\xB2\xB2\xB2\xB2\xB2");  
    printf("\n\n\t\t\t1>> Enter new Student's data\n\t\t\t2>> Update information of existing account\n\t\t\t3>>  
Check the details of existing account\n\t\t\t4>> View all Students of a particular Hostel\n\t\t\t5>> Removing  
existing account\n\t\t\t6>> View all students list\n\t\t\t7>> Exit\n\n\n\n\n\t\t\tEnter your choice:");  
    scanf("%d",&choice);
```

```
switch(choice)
{

    case 1:new_entry();
    break;
    case 2:edit();
    break;
    case 3:see();
    break;
    case 4:sorter();
    break;
    case 5:erase();
    break;
    case 6:view_list();
    break;
    case 7:closer();
    break;
    default:menu();

}
}
```

```
//function to display all students of a hostel
void sorter()
{
    system("CLS");
    int choice,test=0;

    printf("Enter year choice:\n\n\t\t1. BH-1 \n\t\t2. BH-2 \n\t\t3. BH-3 \n\t\t4. GH-1 \n\n");
    scanf("%d",&choice);

    FILE *fp, *hostel;
    switch(choice)
    {
        case 1: hostel=fopen("BH-1.dat","a+");break;
        case 2: hostel=fopen("BH-2.dat","a+");break;
        case 3: hostel=fopen("BH-3.dat","a+");break;
        case 4: hostel=fopen("GH-1.dat","a+");break;
        default: sorter();
    }
}
```

```

while(fscanf(hostel,"%s\n",check.roll_no)!=EOF)
{
    fp=fopen("record.dat","r");
    while(fscanf(fp,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",sort.roll_no, sort.name,
&sort.dob.month, &sort.dob.day, &sort.dob.year, &sort.age, sort.city, &sort.phone, sort.mail_id, sort.father_name,
&sort.father_phone, sort.room_no, sort.hostel_name)!=EOF)
    {
        if(strcmp(check.roll_no,sort.roll_no)==0)
        {
            if(test==0)
            {
                printf("\t\tFollowing is the list of requested students:");
                printf("\n\tROLL NO.\tNAME\tROOM NO.\n");
            }
            printf("\n\t%8s\t %8s\t %s", sort.roll_no, sort.room_no, sort.name);
            test++;
        }
    }
}

```

```
fclose(fp);
    if (test==0)
        printf("\nNO RECORDS!!\n");

    sort_list_invalid:
    printf("\n\nEnter 1 to go to the
main menu and 0 to exit:");
    scanf("%d",&main_exit);

    if (main_exit==1)
        menu();
    else if(main_exit==0)
        closer();
    else
    {
        printf("\nInvalid!\a");
        goto sort_list_invalid;
    }
}
```

```

//new entry function
void new_entry()
{
    system("CLS");
    int choice;
    FILE *ptr,*hostel;

    ptr=fopen("record.dat","a+");
    roll_no:

    printf("\t\t\t\xB2\xB2\xB2 ADD STUDENTS DATA \xB2\xB2\xB2\xB2");
    printf("\nEnter your roll number:");
    scanf("%s",check.roll_no);
    while(fscanf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id,
add.father_name, &add.father_phone, add.room_no, add.hostel_name)!=EOF)
    {
        if (strcmp(check.roll_no,add.roll_no)==0)
            {printf("Students roll no. already in use!");
            goto roll_no;
            }
    }
}

```

```
strcpy(add.roll_no,check.roll_no);
printf("\nEnter the name:");
scanf("%s",add.name);

printf("\nEnter the phone number: ");
scanf("%d",&add.phone);
printf("\n Enter your email id :");
scanf("%s",add.mail_id);
printf("\n Enter your Father's name:");
scanf("%s",add.father_name);
printf("\n Enter your Father's phone no.:");
scanf("%d",&add.father_phone);
printf("\n Enter your room no.:");
scanf("%s",add.room_no);
printf("\n Enter your hostel name from BH-1, BH-2, BH-3, GH-1 :");
scanf("%s",add.hostel_name);
fprintf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name, add.dob.month,
add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id, add.father_name, add.father_phone,
add.room_no, add.hostel_name);
```



```
//adding student roll no. to hostels' list
if(strcmp(add.hostel_name,"BH-1")==0)
{
    hostel=fopen("BH-1.dat","a+");
    fprintf(hostel, "%s\n", add.roll_no);
    fclose(hostel);
}
else if(strcmp(add.hostel_name,"BH-2")==0)
{
    hostel=fopen("BH-2.dat","a+");
    fprintf(hostel, "%s\n", add.roll_no);
    fclose(hostel);
}
else if(strcmp(add.hostel_name,"BH-3")==0)
{
    hostel=fopen("BH-3.dat","a+");
    fprintf(hostel, "%s\n", add.roll_no);
    fclose(hostel);
}
else if(strcmp(add.hostel_name,"GH-1")==0)
{
    hostel=fopen("GH-1.dat","a+");
    fprintf(hostel, "%s\n", add.roll_no);
    fclose(hostel);
}
```

```
close(ptr);
printf("\nStudent added successfully!");
add_invalid:
printf("\n\n\n\t\tEnter 1 to go to the main menu and 0 to exit:");
scanf("%d",&main_exit);

if (main_exit==1)

    menu();
else if(main_exit==0)
    closer();
else
{
    printf("\nInvalid!\a");
    goto add_invalid;
}
}
```

```
//nsbfunction to view all students
void view_list()
{
    system("CLS");
    FILE *view;
    view=fopen("record.dat","r");
    int test=0;

    printf("\nROLL NO.\tNAME\t\t\tcity\t\t\tPHONE\n");

    while(fscanf(view,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id, add.father_name,
&add.father_phone, add.room_no, add.hostel_name)!=EOF)
    {
        printf("\n%8s\t %10s\t\t\t %10s\t\t %d",add.roll_no,add.name,add.city,add.phone);
        test++;
    }
    fclose(view);

    if (test==0)
    {
```

```
printf("\nNO RECORDS!!\n");}
```

```
view_list_invalid:
```

```
printf("\n\nEnter 1 to go to the main menu and 0 to exit:");
```

```
scanf("%d",&main_exit);
```

```
if (main_exit==1)
```

```
    menu();
```

```
else if(main_exit==0)
```

```
    closer();
```

```
else
```

```
{
```

```
    printf("\nInvalid!\a");
```

```
    goto view_list_invalid;
```

```
}
```

```
}
```

```
//-----
```

```
//function to view all data of 1 student
void see()
{
    system("CLS");
    FILE *ptr;
    int test=0;
    int choice;
    ptr=fopen("record.dat","r");
    printf("Do you want to check by\n1.Roll no\n2.Name\nEnter your choice:");
    scanf("%d",&choice);
    if (choice==1)
    {   printf("\n\nEnter the roll number:");
        scanf("%s",check.roll_no);

        while (fscanf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id, add.father_name,
&add.father_phone, add.room_no, add.hostel_name)!=EOF)
```

```
{
    if(strcmp(add.roll_no,check.roll_no)==0)
    {
        test=1;

        printf("\nROLL NO.:%s\nName:%s \nDOB:%d/%d/%d \nAge:%d \ncity:%s \nPhone number:%d
\nE-Mail id:%s \nFather's name:%s \n Father's Phone No.:%d \nRoom No:%s \nHostel:%s \n",add.roll_no,
add.name, add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id,
add.father_name, add.father_phone, add.room_no, add.hostel_name);
    }
}
else if (choice==2)
{ printf("\n\nEnter the name:");
scanf("%s",check.name);
while (fscanf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id, add.father_name,
&add.father_phone, add.room_no, add.hostel_name)!=EOF)
```

```
{
    printf("\n%8s\t %10s\t\t\t %10s\t\t %d",add.roll_no,add.name,add.city,add.phone);
    test++;
}
fclose(view);

if (test==0)
{
    printf("\nNO RECORDS!!\n");}
```

view_list_invalid:

```
printf("\n\nEnter 1 to go to the main menu and 0 to exit:");
scanf("%d",&main_exit);

if (main_exit==1)
    menu();
else if(main_exit==0)
    closer();
else
{
    printf("\nInvalid!\a");
    goto view_list_invalid;
}
```

```
//function to view all data of 1 student
void see()
{
    system("CLS");
    FILE *ptr;
    int test=0;
    int choice;
    ptr=fopen("record.dat","r");
    printf("Do you want to check by\n1.Roll no\n2.Name\nEnter your choice:");
    scanf("%d",&choice);
    if (choice==1)
    {   printf("\n\nEnter the roll number:");
        scanf("%s",check.roll_no);

        while (fscanf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id, add.father_name,
&add.father_phone, add.room_no, add.hostel_name)!=EOF)
```



```
{
    if(strcmp(add.roll_no,check.roll_no)==0)
    {
        test=1;

        printf("\nROLL NO.:%s\nName:%s \nDOB:%d/%d/%d \nAge:%d \ncity:%s \nPhone number:%d
\nE-Mail id:%s \nFather's name:%s \n Father's Phone No.:%d \nRoom No:%s \nHostel:%s \n",add.roll_no,
add.name, add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id,
add.father_name, add.father_phone, add.room_no, add.hostel_name);
    }
}
else if (choice==2)
{ printf("\n\nEnter the name:");
  scanf("%s",check.name);
  while (fscanf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id,
add.father_name, &add.father_phone, add.room_no, add.hostel_name)!=EOF)
```

```

{
    if(strcmp(add.roll_no,check.roll_no)==0)
    {
        test=1;

        printf("\nROLL NO.:%s\nName:%s \nDOB:%d/%d/%d \nAge:%d \ncity:%s \nPhone number:%d
\nE-Mail id:%s \nFather's name:%s \n Father's Phone No.:%d \nRoom No:%s \nHostel:%s \n",add.roll_no,
add.name, add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id,
add.father_name, add.father_phone, add.room_no, add.hostel_name);
    }
}
else if (choice==2)
{ printf("\n\nEnter the name:");
scanf("%s",check.name);
while (fscanf(ptr,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id,
add.father_name, &add.father_phone, add.room_no, add.hostel_name)!=EOF)
{
    if(strcmp(add.name,check.name)==0)
    {
        test=1;
        printf("\nROLL NO.:%s\nName:%s \nDOB:%d/%d/%d \nAge:%d \ncity:%s \nPhone number:%d
\nE-Mail id:%s \nFather's name:%s \n Father's Phone No.:%d \nRoom No:%s \nHostel:%s \n" add.roll_no

```

```
fclose(ptr);
    if(test!=1)
        printf("\nRecord not found!!\a\a\a");
see_invalid:
    printf("\nEnter 0 to try again,1 to return to main menu and 2 to exit:");
    scanf("%d",&main_exit);
    if (main_exit==1)
        menu();
    else if (main_exit==2)
        closer();
    else if(main_exit==0)
        erase();
    else
    {
        printf("\nInvalid!\a");
        goto see_invalid;
    }
}
```

//-----

```

//function to update student info
void edit()
{
    system("CLS");
    int choice,test=0;
    FILE *old,*newrec;
    old=fopen("record.dat","r");
    newrec=fopen("new.dat","w");

    printf("\nEnter the roll no. of the student whose info you want to change:");
    scanf("%s",upd.roll_no);
    while(fscanf(old,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id,
add.father_name, &add.father_phone, add.room_no, add.hostel_name)!=EOF)

    {
        if (strcmp(add.roll_no,upd.roll_no)==0)
        { test=1;
            printf("\nWhich information do you want to change?\n1.Room no.\n2.Phone\n\nEnter your choice(1 for
room no. and 2 for phone ):");
            scanf("%d",&choice);

```

```

if(choice==1)
    {printf("Enter the new room no.:");
    scanf("%s",upd.room_no);
    fprintf(newrec,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id, add.father_name,
add.father_phone, upd.room_no, add.hostel_name);
    printf("Changes saved!");
    }
else if(choice==2)
    {
        printf("Enter the new phone number:");
        scanf("%d",&upd.phone);
        fprintf(newrec,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
add.dob.month, add.dob.day, add.dob.year, add.age, add.city, upd.phone, add.mail_id, add.father_name,
add.father_phone, add.room_no, add.hostel_name);
        printf("Changes saved!");
    }

}
else
    fprintf(newrec,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id, add.father_name,
add.father_phone, add.room_no, add.hostel_name);
}

```

```
fprintf(newrec,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name, add.dob.month,
add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id, add.father_name, add.father_phone,
add.room_no, add.hostel_name);
```

```
}
fclose(old);
fclose(newrec);
remove("record.dat");
rename("new.dat","record.dat");
```

```
    if(test!=1)
        printf("\nRecord not found!!\a\a\a");
```

```
edit_invalid:
```

```
    printf("\nEnter 0 to try again,1 to return to main menu and 2 to exit:");
    scanf("%d",&main_exit);
    if (main_exit==1)
        menu();
    else if (main_exit==2)
        closer();
    else if(main_exit==0)
        edit();
    else
    {
        printf("\nInvalid!\a");
```

```

//function to update student info
void edit()
{
    system("CLS");
    int choice,test=0;
    FILE *old,*newrec;
    old=fopen("record.dat","r");
    newrec=fopen("new.dat","w");

    printf("\nEnter the roll no. of the student whose info you want to change:");
    scanf("%s",upd.roll_no);
    while(fscanf(old,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id,
add.father_name, &add.father_phone, add.room_no, add.hostel_name)!=EOF)

    {
        if (strcmp(add.roll_no,upd.roll_no)==0)
        { test=1;
            printf("\nWhich information do you want to change?\n1.Room no.\n2.Phone\n\nEnter your choice(1 for
room no. and 2 for phone ):");
            scanf("%d",&choice);

            if(choice==1)
                {printf("Enter the new room no ."):

```

```

//cfunction to delete records
void erase()
{
    system("CLS");
    FILE *old,*newrec;
    int test=0;
    old=fopen("record.dat","r");
    newrec=fopen("new.dat","w");
    printf("Enter the roll np. of student whose data you want to delete:");
    scanf("%s",rem.roll_no);
    while (fscanf(old,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
&add.dob.month, &add.dob.day, &add.dob.year, &add.age, add.city, &add.phone, add.mail_id,
add.father_name, &add.father_phone, add.room_no, add.hostel_name)!=EOF)
    {
        if(add.roll_no!=rem.roll_no)
            fprintf(newrec,"%s %s %d/%d/%d %d %s %d %s %s %d %s %s\n",add.roll_no, add.name,
add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone, add.mail_id, add.father_name,
add.father_phone, add.room_no, add.hostel_name);

        else
            {test++;
            printf("\nRecord deleted successfully!\n");
            }
    }
}

```



```
//function displaying developer credits 1353functioncalls
void closer()
{
    system("CLS");
    printf("\n\n\n\nBrought To You by Tansu and Deepayan\n\n\n\n");
}
```

```
int pass(void)
{
    int a=0,i=0;
    char uname[10],c=' ';
    char pword[10],code[10];
    char user[10]="user";
    char pass[10]="pass";
    do
    {
        system("cls");

        printf("\n ***** LOGIN FORM ***** ");
        printf("\n          ENTER USERNAME:-");
        scanf("%s", &uname);
        printf("\n          ENTER PASSWORD:-");
        while(i<10)
        {
```

USER NAME — user

PASSWORD – pass

Do you want to check by

1.Roll no

2.Name

Enter your choice:1

Enter the roll number:RA2111003010321

ROLL NO.:RA2111003010321

Name:Rudra

DOB:842215473/49/1997

Age:21

city:Gangopadhyay

Phone number:792149776

E-Mail id:rudra@gmail.com

Father's name:AbC

Father's Phone No.:1014182200

Room No:1022

Hostel:BH-1

Enter 0 to try again,1 to return to main menu and 2 to exit:█

Do you want to check by

1.Roll no

2.Name

Enter your choice:1

Enter the roll number:RA2111003010321

ROLL NO.:RA2111003010321

Name:Rudra

DOB:842215473/49/1997

Age:21

city:Gangopadhyay

Phone number:792149776

E-Mail id:rudra@gmail.com

Father's name:AbC

Father's Phone No.:1014182200

Room No:1022

Hostel:BH-1

Enter 0 to try again,1 to return to main menu and 2 to exit:█

```
***** LOGIN FORM *****  
ENTER USERNAME: -█
```

HOSTEL MANAGEMENT SYSTEM

██████████ WELCOME TO THE MAIN MENU ██████████

- 1>> Enter new Student's data
- 2>> Update information of existing account
- 3>> Check the details of existing account
- 4>> View all Students of a particlar Hostel
- 5>> Removing existing account
- 6>> View all students list
- 7>> Exit

Enter your choice:█

DECLARATION

We hereby declare that this project work entitled “**HOSTEL MANAGEMENT SYSTEM**” has been prepared by us during the year 2021-2022 under the guidance of **MS.M.HEMA MA'AM** Department of Computer Science , SRMIST.KTR.