HEMWATI NANDAN BAHUGUNA GARHWAL UNIVERSITY

(A CENTRAL UNIVERSITY)

Mini Project Report File

on

HOSTEL ACCOMMODATION SYSTEM

Submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology

Guide by-

Mr Shailesh Panwar

Submitted by-

Md Minhaj Alam

Roll no-20134503014

Yash

Roll no- 20134503029

Btech (It) 4th sem

Session(2020-2024)

CANDIDATE'S SELF DECLARATION

We hereby declare that the content presented in this Mini Project Report titled **" Hostel Accommodation System"** is done by us is an authentic work carried out for the Partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in Information and technology submitted to **School of Engineering and Technology, H.N.B.Garhwal University, Srinagar (Garhwal), Uttarakhand**, under the guidance of Mr. Shailesh Panwar.

We further declare that the matter embodied in this report has not been submitted earlier for award of any degree or diploma to the best of our knowledge and belief.

Md Minhaj Alam (2020-2024)

Yash (2020-2024)

CERTIFICATE

This is to certify that the content presented in this report of mini project titled "**Hostel Accommodation System**" submitted in Partial fulfilment of the requirements for the degree of **Bachelor of Technology**(Information and Technology) to the Department of Information and Technology , **School of Engineering and Technology, H.N.B.Garhwal University, Srinagar (Garhwal), Uttarakhand**, done by **Md Minhaj Alam and Yash** is an authentic work carried out by them under my guidance.

The matter embodied in this has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

ACKNOWLEDGEMENT

We would like to thank respected Mr. Shailesh panwar for giving us such a wonderful opportunity to expand our knowledge for our own branch and giving us guidelines to make a mini project. It helped us a lot to realize of what we study for.

We are highly indebted to Mr. Vinay Tamta, Head of Department, Department of Information and Technology for his support during the mini project.

Thirdly, we would like to thank our parents who patiently helped us as we went through our work and helped to modify and eliminate some of the irrelevant or unnecessary stuffs.

Fourthly, we would like to thank our friends who helped us to make our work more organized and well-stacked till the end.

Last but clearly not the least, we would thank The Almighty for giving us strength to complete our project on time.

ABSTRACT

As the name specifies “HOSTEL ACCOMODATION SYSTEM” is a software developed for managing various activities in the hostel. For the past few years, the number of educational institutions are increasing rapidly. Thereby, the number of hostels are also increasing for the accommodation of the students studying in this institution. And hence there is a lot of strain on the person who are running the hostel and software’s are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

CONTENTS

1. INTRODUCTION

1.1 Problem definition

We have got four hostels in our university, which consist of three boy's hostel and one girl's hostel. All these hostels at present are managed manually by the hostel office. The Registration form is filled manually.

Thus, there are a lot of repetitions which can be easily avoided. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

1.2 Solution

We have developed a platform that will help the both the administration and the student to manage hostel information more accurately.

LOGIN PORTAL

1. Home

2. New user

3. Administrator login

4. Student login

5. Exit

1. HOME

1.1 Hostel Brochure

1.2. Notice Board

1.3. Registration Form

1.4. Complaints and Suggestions

1.5. Rules and Regulations 1.6.

Exit

2. ADMINISTRATOR

LOGIN 2.1. All hostel

allotments

2.2. Allotments of particular hostel

2.3. Vacating of rooms

2.4. Hostel transfer

2.5. Complaints

2.6. Exit

3. STUDENT LOGIN

3.1. Personal Information

3.2. Update Information

3.3. Complaints and

Suggestions 3.4 Exit

2. SYSTEM ENVIRONMENT

2.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

Hardware Configuration

Processor: Intel dual Core, i3

RAM: 4GB

Hard disk: 80 GB

2.2 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software Configuration

Operating System: Windows 10/ XP/8

Programming Language: C

IDE: Code blocks / Dev C++

3. SYSTEM ANALYSIS

3.1 Existing System

For the past few years, the number of educational institutions are increasing rapidly. Thereby, the number of hostels are also increasing for the accommodation of the students studying in the institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context.

This particular project deals with the problems on managing hostels and avoids the problems which occur when carried manually.

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly. We can improve the efficiency of the system, thus overcome the following drawbacks of the existing system.

1. more human error
2. more strength and strain of manual labour needed
3. repetition of the same procedures
4. low security
5. data redundancy
6. difficult to handle
7. difficult to update data
8. record keeping is difficult

4. SYSTEM DESIGN

4.1 Input Design

There is a Login Portal from which user can choose what he want to

do. Login portal is divided into four parts:

- Home
- New user
- Administrator login
- Student login

Home

In this user can perform the following tasks:

- see hostel brochure
- see and write notice
- fill registration form
- see and write complaints and suggestions
- see rules and regulations of hostel

New user

New user can check username and password to login the

system. One can check it by entering

- Admin
- Student

Administrator login

In this user can perform the following tasks:

- see all hostel allotments

- see allotments of particular hostel
- vacate rooms of hostel
- transfer student to other hostel
- see and write complaints and suggestions

Student login

In this user can perform the following tasks:

- see personal information
- update information
- see and write complaints and suggestions

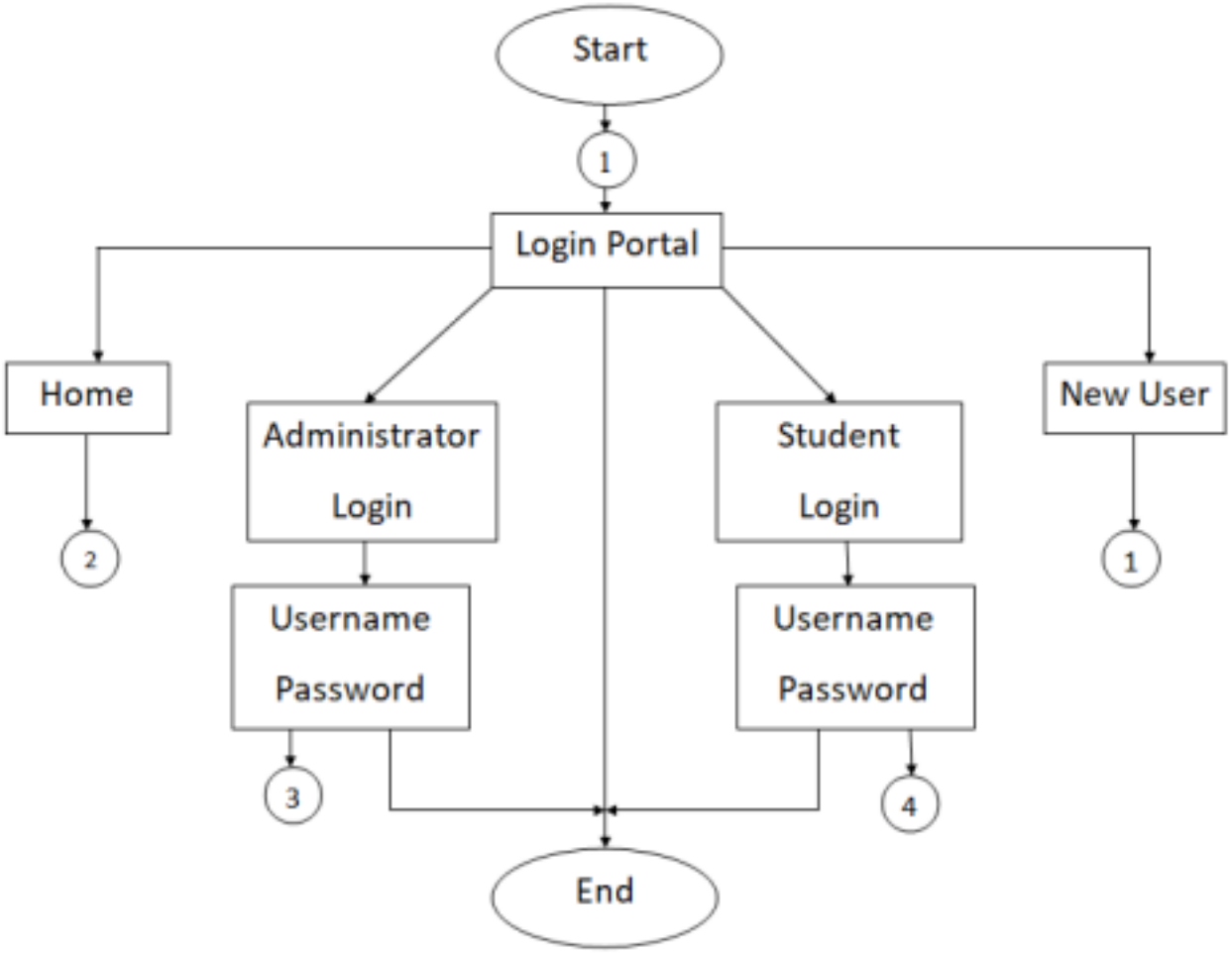
*There is a login before Administrator login and Student login where user has to enter username and password to perform specific tasks.

4.2 Process Design

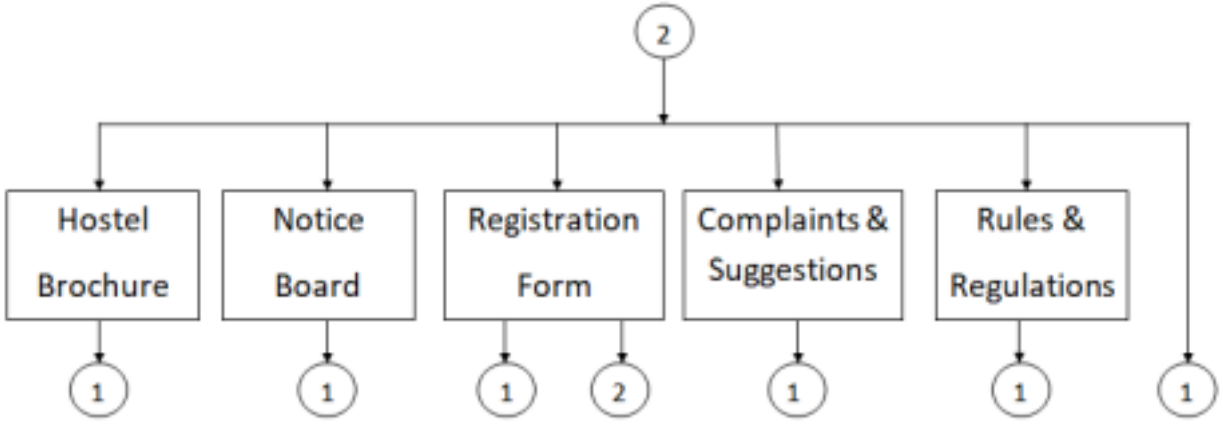
Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. System Flow charts are the tools used for process design.

System Flow Chart is a graphical representation of the system showing the overall flow of control in processing at the job level; specifies what activities must be done to convert from a physical to logical model.

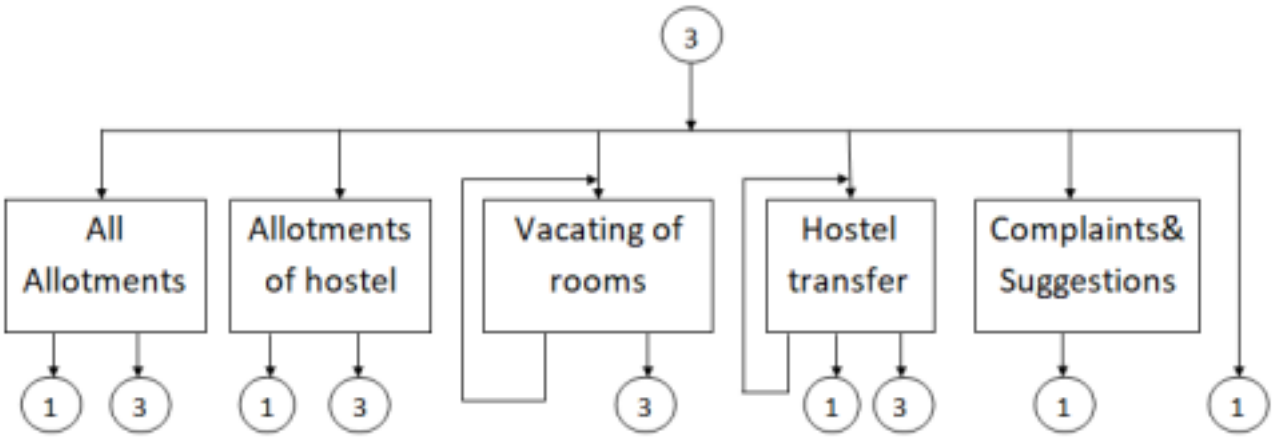
For Login Portal



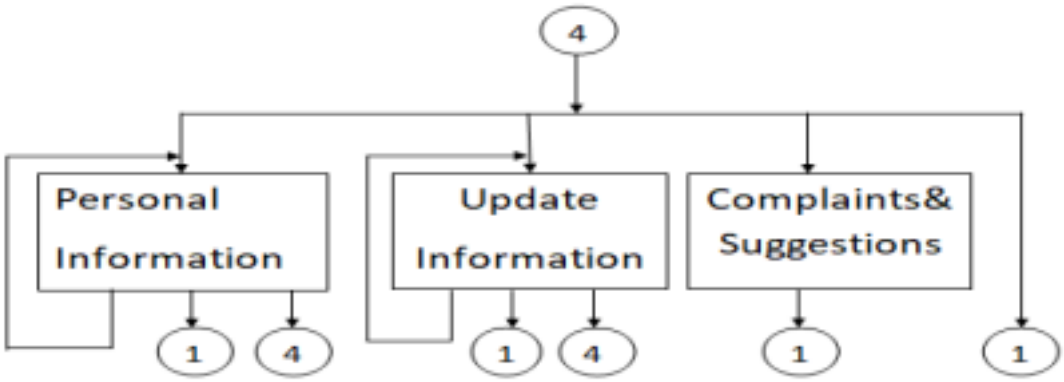
For Home



For Administration Login



For Student login



4.3 Output Design

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system well executed. When we design an output we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of their output.

Once the output requirements are determined, the system designer can decide what to include in the system and how to structure it so that the require output can be produced. For the proposed software, it is necessary that the output reports be compatible in format with the existing reports.

The output must be concerned to the overall performance and the system's working, as it should.

It consists of developing specifications and procedures for data preparation, those steps necessary to put the inputs and the desired output, i.e. maximum user friendly. Proper messages and appropriate directions can control errors committed by users.

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the system's working, as it should.

Output design consists of displaying specifications and procedures as data presentation. User never left with the confusion as to what is happening without appropriate error and acknowledges message being received. Even an unknown person can operate the system without knowing anything about the system.

5. CODE

```
//including header files

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<conio.h>

//defining global variables

int i,j,main_exit;

//declaring functions

void login(void);

void home(void);

void newuser(void);

void pass(void);
```

```
void brochure(void);

void notice(void);

void new_reg(void);

void complaint(void);

void rule(void);

void admin(void);

void allallotments(void);

void allotments(void);

void vacating(void);

void transfer(void);

void student(void);

void info(void);

void update(void);
```

```
//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];

 char roll_no[10];

 char hostel_name[6];
```

```
 struct date dob;

 char phone[10];

 }add,upd,check,rem,sort;

//-------------------------------------------------------------------------------------------------------

//main function

int main()

{

 login();

}

//-------------------------------------------------------------------------------------------------------

//login portal function

void login()

{ int x,a=1;
```

```
while(a!=0)

{ system("CLS");

printf("\n\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2 HOSTEL ACCOMMODATION SYSTEM
\xB2\xB2\xB2\xB2\xB2\xB2\xB2 ");

printf("\n\t\t\tLOGIN PORTAL");

printf("\n\n 1. HOME \n 2. NEW USER \n 3. ADMINISTRATOR LOGIN \n 4. STUDENT LOGIN
\n 5. EXIT ");

printf("\n\n Enter your choice : ");

scanf("%d",&x);

switch(x)

{

case 1 : home();

break;

case 2 : newuser();

break;
```

```
case 3 : {pass();

admin();

}

break;

case 4 : {pass();

student();

}

break;

case 5 : exit(0);

break;

default: printf("\n Wrong choice entered !!!!!! ");

}

printf("\n Want to go LOGIN PORTAL (0/1): ");
```

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

 }

}

//-----------------------------------------------------------------------------------------------------------

//function for home portal

void home()

{

 int x,a=1,b=1;

 while(a!=0)

 { system("CLS");

 printf("\n\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2 HOSTEL ACCOMMODATION SYSTEM
\xB2\xB2\xB2\xB2\xB2\xB2\xB2 ");

 printf("\n\t\t\t HOME");

 printf("\n\n 1. HOSTEL BROCHURE \n 2. NOTICE BOARD \n 3. REGISTRATION FORM \n 4.
COMPLAINTS AND SUGGESTIONS \n 5. RULES AND REGULATIONS \n 6. EXIT ");
```

```
printf("\n\n Enter your choice : ");

scanf("%d",&x);

switch(x)

{

case 1 : brochure();

break;

case 2 : notice();

break;

case 3 : new_reg();

break;

case 4 : complaint();

break;

case 5 : rule();
```

```
break;

case 6 : login();

break;

default: printf("\n Wrong choice entered !!!!!! ");

}

printf("\n\n Want to go HOME (0/1): ");

scanf("%d",&a);

}

printf("\n Want to go login Portal (0/1) : ");

scanf("%d",&b);

if(b==1)

login();

else
```

```
 exit(0);

}

//-----------------------------------------------------------------------------------------------------------

//brochure function

void brochure()

{ system("CLS");

 printf("\n\t\t HOSTEL PROSPECTUS AND INFORMATION BROCHURE ");

 printf("\n\n\t\t HOSTEL ADMINISTRATION \n\t\t Chief Warden - Prof. Sunil Sharma \n\n Wardens for Boys Hostels \n\t 1. BH-1 (J. C. Bose Sadan) - Dr. Hardev Singh \n\t 2. BH-2 (Aryabhatt Sadan) - Dr. Vivek Gupta \n\t 3. Bh-3 (Madan Lal Dhingra Sadan) - Dr. Manoj Kumar \n\n Wardens for Girls Hostels\n\t 1. GH-1 (Saraswati Bhawan) - Dr. Anu Gupta ");

}

//-----------------------------------------------------------------------------------------------------------

//notice function

void notice()
```

```
{ system("CLS");

int a;

FILE *ptr,*f1;

char z,b;

f1=fopen("notice.dat","r");

printf("\n \t\t NOTICE \n");

while((b=getc(f1))!=EOF)

printf("%c",b);

fclose(f1);

printf("\n\n Want to write a notice (0/1): ");

scanf("%d",&a);

if(a==1)

{ ptr=fopen("notice.dat","a+");
```

```
 printf("\n Write date : ");

 scanf("%d/%d/%d",&add.dob.day,&add.dob.month,&add.dob.year);

 fprintf(ptr,"\n%d/%d/%d",add.dob.day,add.dob.month,add.dob.year);

 printf("\n\t\t Notice\n");

 while((z=getchar())!='.')

 {

 putc(z,ptr);

 }

 fprintf(ptr,"\n");

 fclose(ptr);

 }

}

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

```
//function for new registration

void new_reg()

{ system("CLS");

 int choice;

 FILE *ptr,*hostel;

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

 roll_no:

 printf("\t\t\t\xB2\xB2\xB2 NEW REGISTRATION \xB2\xB2\xB2\xB2");

 printf("\n\t\t\t ADD STUDENTS DATA");

 printf("\nEnter your roll number:");

 scanf("%s",check.roll_no);

 while(fscanf(ptr,"%s %s %d/%d/%d %d %s %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.room_no, add.hostel_name)!=EOF)

 {
```

```
if (strcmp(check.roll_no,add.roll_no)==0)

{printf("Students roll no. already in use!\n\n");

goto roll_no;

}

}

strcpy(add.roll_no,check.roll_no);

printf("\nEnter the name:");

scanf("%s",add.name);

printf("\nEnter your Father's name:");

scanf("%s",add.father_name);

printf("\nEnter the date of birth(mm/dd/yyyy):");

scanf("%d/%d/%d",&add.dob.month,&add.dob.day,&add.dob.year);

printf("\nEnter the age:");
```

```
scanf("%d",&add.age);

printf("\nEnter the city:");

scanf("%s",add.city);

printf("\nEnter the phone number: ");

scanf("%s",&add.phone);

printf("\n Enter your email id :");

scanf("%s",add.mail_id);

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 %s %s %s %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.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);

}
```

```
fclose(ptr);

printf("\nStudent added successfully!");

add_invalid:

printf("\n\n\n\t\tEnter 1 to go to HOME and 0 to LOGIN PORTAL:");

scanf("%d",&main_exit);

if (main_exit==1)

home();

else if(main_exit==0)

login();

else

{

printf("\nInvalid!\a");

goto add_invalid;
```

```
 }

}

//-------------------------------------------------------------------------------------------------------------

//complaint function

void complaint()

{ system("CLS");

 int a;

 FILE *ptr,*f1;

 char z,b;

 f1=fopen("complaint.dat","r");

 printf("\n \t\t\xB2\xB2\xB2 COMPLAINTS AND SUGGESTIONS \xB2\xB2\xB2");

 while((b=getc(f1))!=EOF)

 printf("%c",b);
```

```
fclose(f1);

printf("\n\n Want to write a complaint or suggestion (0/1) : ");

scanf("%d",&a);

if(a==1)

{ ptr=fopen("complaint.dat","a+");

printf("\n Write date : ");

scanf("%d/%d/%d",&add.dob.day,&add.dob.month,&add.dob.year);

fprintf(ptr,"\n%d/%d/%d",add.dob.day,add.dob.month,add.dob.year);

while((z=getchar())!='.')

{

putc(z,ptr);

}

fprintf(ptr,"\n");
```

```
 fclose(ptr);

 }

 }

//-----------------------------------------------------------------------------------------------------------

//rules function

void rule()

{ system("cls");

printf("\t\t\t HOSTEL RULES & REGULATIONS \n 1. GENERAL: \n 1.1 These rules shall be known as “Hostel Rules & Regulations” \n 1.2 These rules shall apply to all hostel residents of the hostels of the University. \n\n 2. HOSTEL ADMINISTRATION: \n 2.1 Every Hostel shall have a Coordinator/Warden, who will be responsible for the administration of the Hostel and for the enforcement of the Hostel Rules. The Coordinator/Warden will be appointed by the Hon'ble Vice-Chancellor on the recommendations of the Chief Warden from amongst the members of the Faculty. \n 2.2 The Coordinator/Warden will have the following responsibilities: \n (a) He/She will be responsible for the allotment of Hostel rooms in accordance with the policy laid down by the university and for the maintenance of discipline within the hostel. \n (b) He/She (In case of teacher Coordinator/Warden) will normally attend the Hostel office daily for one hour, on all working days. \n (c) He/She will appoint the prefects of the Hostel as prescribed in the relevant rule and supervise their work. \n (d) He/She shall ensure that no unauthorized person stays in the hostel and will grant permission for the stay of guests. \n (e) The Powers for disciplinary action of the Chief Warden/Coordinator/Warden shall be as under, the punishment depending on the nature and severity of the offence. \n (f) A register will be maintained for the purpose by the security guard in which due entries will be made by boarders coming late in the night or after closing time of hostel gate whichever is applicable. \n (g) FOR GIRLS HOSTELS ONLY - All the residents are expected to be in their rooms at the time of roll-call which will be taken by the lady warden on timings fixed as under: Summer 7:00 p.m. (April to Sept.) Winter 6:00 p.m. (Oct. to March).");printf("\n\n \t\t ZERO TOLERANCE ON RAGGING \n• Ragging in the University Campus is totally banned. \n• Any student found indulging in the act of ragging shall be liable to disciplinary action as per the University rules and be punished according to law. • Anyone who finds ragging or being ragged in the campus can submit his/her complaint to Proctor on Email
```

```
23ID: proctor@gjust.org \n• Anyone who finds ragging or being ragged in the hostels can submit his/her complaint to Warden, Coordinator,and Chief Warden.");

}

//-----------------------------------------------------------------------------------------------------------
//function for new uservoid newuser()

void newuser()

{

 int a;

 system("cls");

 printf("\n\t NEW USER");

 printf("\n You are \n 1. ADMIN \n 2. STUDENT ");

 printf("\n Enter your choice : ");

 scanf("%d",&a);

 if(a==1)

 { printf("\n USERNAME : admin \n PASSWORD : 12345");
```

```
 }

 else

 if(a==2)

 { printf("\n USERNAME : student \n PASSWORD : 54321");

 }

 else printf("\n Wrong choice entered !!!!");

}

//------------------------------------------------------------------------------------------------------------

//function for admin portal

void admin()

{

 int x,a=1,b=1;

 while(a!=0)
```

```
{ system("cls");

printf("\n\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2 HOSTEL ACCOMMODATION SYSTEM
\xB2\xB2\xB2\xB2\xB2\xB2\xB2 ");

printf("\n\t\t\t ADMIN PORTAL");

printf("\n\n 1. ALL HOSTEL ALLOTMENTS \n 2. ALLOTMENTS OF PARTICULAR HOSTEL \n
3. VACATING OF ROOMS \n 4. HOSTEL TRANSFER \n 5. COMPLAINTS \n 6. EXIT ");

printf("\n\n Enter your choice : ");

scanf("%d",&x);

switch(x)

{

case 1 : allallotments();

break;

case 2 : allotments();

break;

case 3 : vacating();
```

```
break;

case 4 : transfer();

break;

case 5 : complaint();

break;

case 6 : login();

break;

default: printf("\n Wrong choice entered !!!!!! ");

}

printf("\n Want to go ADMIN PORTAL (0/1): ");

scanf("%d",&a);

}

printf("\n Want to go login Portal (0/1) : ");
```

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

 if(b==1)

 login();

 else

 exit(0);

}

//-------------------------------------------------------------------------------------------------------------

//all student allotment list

void allallotments()

{ system("CLS");

 printf("\n\t\t \xB2\xB2\xB2\xB2 ALL ALLOTMENTS \xB2\xB2\xB2\xB2");

 FILE *view;

 view=fopen("record.dat","r");
```

```
 int test=0;

 printf("\nHOSTEL NAME \t\t ROLL NO. \t\tNAME \t\t\tCITY\n");

 while(fscanf(view,"%s %s %d/%d/%d %d %s %s %s %s %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.room_no, &add.hostel_name)!=EOF)

 {

 printf("\n%10s \t %8s \t\t %10s\t\t %10s",add.hostel_name,add.roll_no,add.name,add.city);

test++;

 }

 fclose(view);

 if (test==0)

 {

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

```
view_list_invalid:

printf("\n\nEnter 1 to go to ADMIN PORTAL and 0 to LOGIN PORTAL:");

scanf("%d",&main_exit);

if (main_exit==1)

admin();

else if(main_exit==0)

login();

else

{

printf("\nInvalid!\a");

goto view_list_invalid;

}

}
```

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

//allotments of particular hostel

void allotments()

{

//  system("CLS");

    int choice = 0, test = 0;

//  printf("\n\t\t\xB2\xB2\xB2\xB2 ALLOTMENTS OF PARTICULAR HOSTEL
\xB2\xB2\xB2\xB2");

   printf(" ALLOTMENTS OF PARTICULAR HOSTEL");

 printf(" \n\t HOSTELS \n\t1. BH-1 \n\t2. BH-2 \n\t3. BH-3 \n\t4. GH-1 \n\n\tEnter your choice : ");

 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: allotments();

 }

 while(fscanf(hostel,"%s\n",check.roll_no)!=EOF)

 {

 fp=fopen("record.dat","r");

 while(fscanf(fp,"%s %s %d/%d/%d %d %s %s %s %s %s %s\n",sort.roll_no, sort.name,

&add.dob.month, &add.dob.day, &sort.dob.year, &sort.age, sort.city, &sort.phone, sort.mail_id,

sort.father_name, sort.room_no, sort.hostel_name)!=EOF)

 {
```

```
if(strcmp(check.roll_no,sort.roll_no)==0)

{

if(test==0)

{

printf("\tFollowing is the list of requested students:");

printf("\n\t ROLL NO. \t\tROOM NO.\tNAME\n");

}

printf("\n\t%8s\t %8s\t\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\n Enter 1 to go to ADMIN PORTAL and 0 to LOGIN PORTAL:");

scanf("%d",&main_exit);

if (main_exit==1)

admin();

else if(main_exit==0)

login();

else

{

printf("\nInvalid!\a");

goto sort_list_invalid;
```

```
 }

}

//-------------------------------------------------------------------------------------------------------------

//vacating hostel

void vacating()

{ system("CLS");

 printf("\n\t\t \xB2\xB2\xB2\xB2 VACATING HOSTEL \xB2\xB2\xB2\xB2");

 FILE *old,*newrec;

 int test=0;

 old=fopen("record.dat","r");

 newrec=fopen("new.dat","w");

 printf("\n\nEnter the roll no. of student whose data you want to delete:");

 scanf("%s",&rem.roll_no);
```

```
 while (fscanf(old,"%s %s %d/%d/%d %d %s %s %s %s %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.room_no, &add.hostel_name)!=EOF)
 {
 if(strcmp(add.roll_no,rem.roll_no)!=0)
 fprintf(newrec,"%s %s %d/%d/%d %d %s %s %s %s %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.room_no, add.hostel_name);
 else
 {test++;
 printf("\nRecord deleted successfully!\n");
 }
 }
```

```
fclose(old);

fclose(newrec);

remove("record.dat");

rename("new.dat","record.dat");

if(test==0)

printf("\nRecord not found!!\a\a\a");

erase_invalid:

printf("\nEnter 0 to try again,1 to return to ADMIN PORTAL :");

scanf("%d",&main_exit);

if (main_exit==1)

admin();

else if(main_exit==0)

vacating;
```

```
 else

 {

 printf("\nInvalid!\a");

 goto erase_invalid;

}

}

//-----------------------------------------------------------------------------------------------------------

//vacating of hostel

void transfer()

{ system("CLS");

 printf("\n\t\t \xB2\xB2\xB2\xB2 HOSTEL TRANSFER \xB2\xB2\xB2\xB2");

 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 which you want to transfer:");

 scanf("%s",&upd.roll_no);

 while(fscanf(old,"%s %s %d/%d/%d %d %s %s %s %s %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.room_no, add.hostel_name)!=EOF)

 {

 if (strcmp(add.roll_no,upd.roll_no)==0)

 { printf("Enter the new hostel name:");

 scanf("%s",&upd.hostel_name);

 fprintf(newrec,"%s %s %d/%d/%d %d %s %s %s %s %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.room_no, upd.hostel_name);

 printf("Changes saved!");

 test=1;

 }

 else

 fprintf(newrec,"%s %s %d/%d/%d %d %s %s %s %s %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.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 ADMIN PORTAL and 2 to LOGIN PORTAL :");

 scanf("%d",&main_exit);

 if (main_exit==1)

admin();

 else if (main_exit==2)

 login();

 else if(main_exit==0)

 transfer();

 else

 {
```

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

 goto edit_invalid;

 }

}

//-----------------------------------------------------------------------------------------------------------

//student portal

void student()

{

 int x,a=1,b=1;

 while(a!=0)

 { system("cls");

 printf("\n\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2 HOSTEL ACCOMMODATION SYSTEM
\xB2\xB2\xB2\xB2\xB2\xB2\xB2 ");

 printf("\n\t\t\t STUDENT PORTAL");
```

```
printf("\n\n 1. PERSONAL INFORMATION \n 2. UPDATE INFORMATION \n 3. COMPLAINTS AND SUGGESTIONS \n 4. EXIT ");

printf("\n\n Enter your choice : ");

scanf("%d",&x);

switch(x)

{

case 1 : info();

break;

case 2 : update();

break;

case 3 : complaint();

break;

case 4 : login();

break;
```

```
default: printf("\n Wrong choice entered !!!!!! ");

}

printf("\n Want to go STUDENT PORTAL (0/1): ");

scanf("%d",&a);

}

printf("\n Want to go LOGIN PORTAL (0/1) : ");

scanf("%d",&b);

if(b==1)

login();

else

exit(0);

}

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

```
//view personal information

void info()

{ system("CLS");

 printf("\n\t\t \xB2\xB2\xB2\xB2 PERSONAL INFORMATION \xB2\xB2\xB2\xB2");

 FILE *ptr;

 int test=0;

 int choice;

 ptr=fopen("record.dat","r");

 printf("\n\nDo 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 %s %s %s %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.room_no, &add.hostel_name)!=EOF)

 {

 if(strcmp(add.roll_no,check.roll_no)==0)

 {

 test=1;

 printf("\nRoll No.:%s\nName:%s \nFather's name:%s \nDOB:%d/%d/%d \nAge:%d \ncity:%s
\nPhone number:%s \nE-Mail id:%s \nRoom No:%s \nHostel:%s \n",add.roll_no, add.name,
add.father_name, add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone,
add.mail_id, 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 %s %s %s %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.room_no, add.hostel_name)!=EOF)

 {

 if(strcmp(add.name,check.name)==0)

 {

 test=1;

 printf("\nRoll No.:%s\nName:%s \nFather's name:%s \nDOB:%d/%d/%d \nAge:%d \ncity:%s
\nPhone number:%s \nE-Mail id:%s \nRoom No:%s \nHostel:%s \n",add.roll_no, add.name,
add.father_name, add.dob.month, add.dob.day, add.dob.year, add.age, add.city, add.phone,
add.mail_id, add.room_no, add.hostel_name);

 }

 }
```

```
}

fclose(ptr);

if(test!=1)




printf("\nRecord not found!!\a\a\a");

see_invalid:

printf("\nEnter 0 to try again,1 to return to STUDENT PORTAL and 2 to LOGIN PORTAL:");

scanf("%d",&main_exit);

if (main_exit==1)

student();

else if (main_exit==2)

login();

else if(main_exit==0)
```

```
 info();

 else

 {

 printf("\nInvalid!\a");

 goto see_invalid;

}

}

//-----------------------------------------------------------------------------------------------------------

//update information

void update()

{ system("CLS");

 printf("\n\t\t \xB2\xB2\xB2\xB2 UPDATE INFORMATION \xB2\xB2\xB2\xB2");

 int choice,test=0;
```

```
 FILE *old,*newrec;

 old=fopen("record.dat","r");

 newrec=fopen("new.dat","w");

 printf("\n\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 %s %s %s %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.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 %s %s %s %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, upd.room_no, add.hostel_name);

printf("Changes saved!");

}

else if(choice==2)

{

printf("Enter the new phone number:");

scanf("%s",&upd.phone);

fprintf(newrec,"%s %s %d/%d/%d %d %s %s %s %s %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.room_no, add.hostel_name);

 printf("Changes saved!");

 }

 }

 else

 fprintf(newrec,"%s %s %d/%d/%d %d %s %s %s %s %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.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 STUDENT PORTAL and 2 to LOGIN PORTAL :");

 scanf("%d",&main_exit);

 if (main_exit==1)

student();

 else if (main_exit==2)

 login();

 else if(main_exit==0)

 update();

 else
```

```
{

 printf("\nInvalid!\a");

 goto edit_invalid;

}

}

//-------------------------------------------------------------------------------------------------------------

//login function

void pass() {

 int a=0,i=0;

 char uname[10],c=' ';

 char pword[10],code[10];

 char user[10]="admin";

 char pass[10]="12345";
```

```
 char suser[10]="student";

 char spass[10]="54321";

 do {

system("cls");

 printf(" \n\n ENTER USERNAME:-");

scanf("%s", &uname);

printf(" \n ENTER PASSWORD:-");

while(i<10)

{

 pword[i]=getch();

 c=pword[i];

 if(c==13) break;

 else printf("*");
```

```
 i++;

}

pword[i]='\0';

//char code=pword;

i=0;

//scanf("%s",&pword);

if((strcmp(uname,user)==0 &&

strcmp(pword,pass)==0)||strcmp(uname,suser)==0 && strcmp(pword,spass)==0)

{

printf(" \n\n\n WELCOME !!!! LOGIN IS SUCCESSFUL\n");

system("PAUSE");

break;

}
```

```
else

{

printf("\n SORRY !!!! LOGIN IS UNSUCESSFUL\n");

system("PAUSE");

a++;

getch();

}

}

while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for four times!!!");

exit(0);
```

```
}

}
```

CHAPTER 6

SYSTEM TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance. Tests are carried out and the results are compared with the expected document. In the case of erroneous results, debugging is done. Using detailed testing strategies, a test plan is carried out on each module. The various tests performed are unit testing, integration testing and user acceptance testing.

6.1 Unit Testing

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining. The various controls are tested to ensure that each performs its action as required.

6.2 Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

6.3 User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

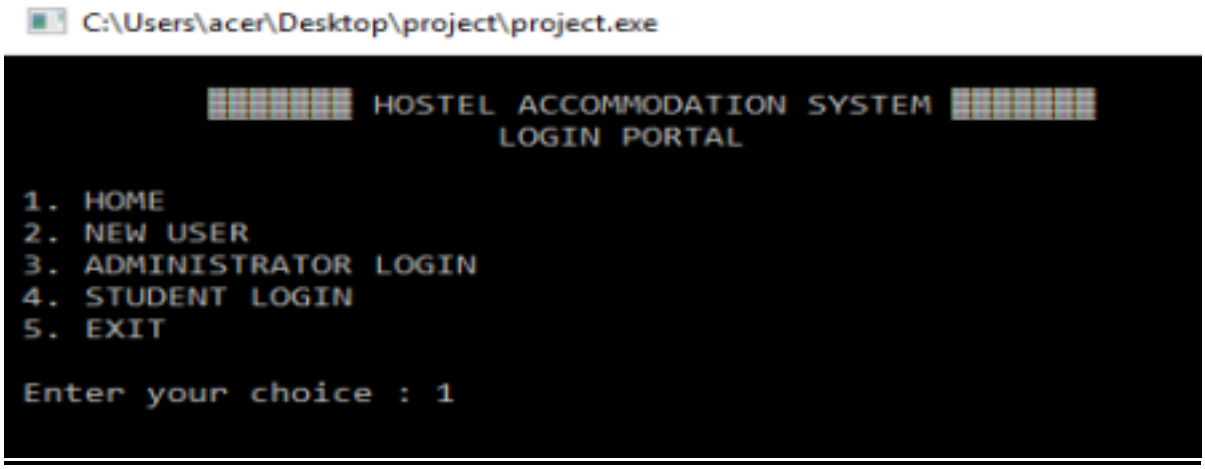
7. IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

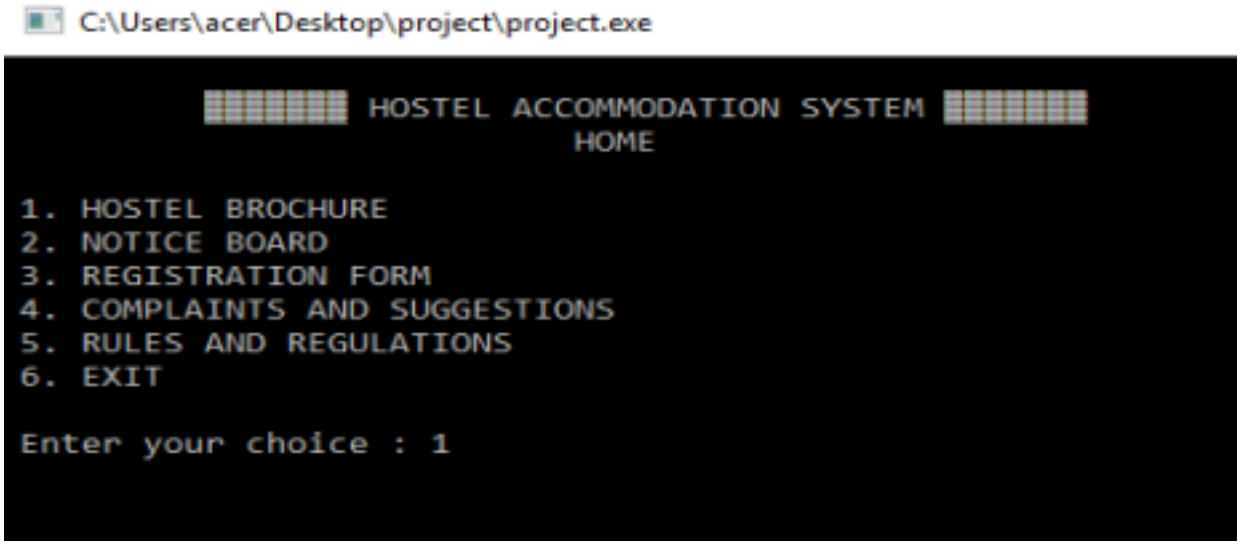
Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification.

7.1 Sample Screenshots

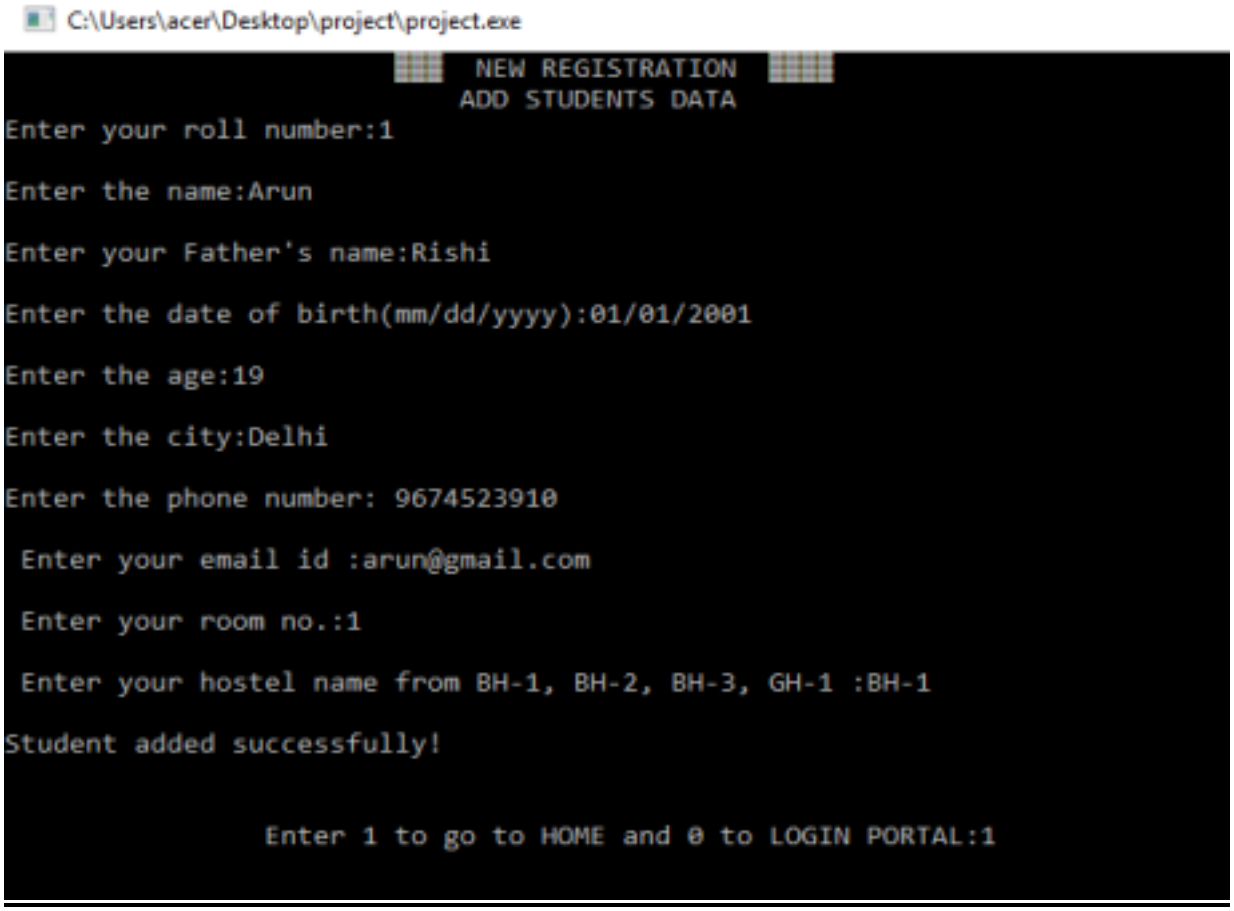
Login Portal



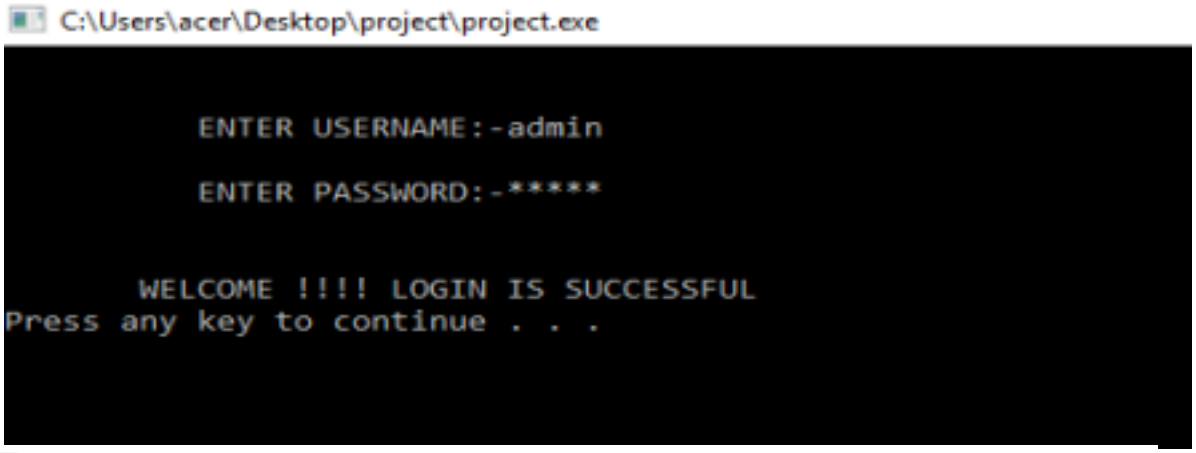
Home

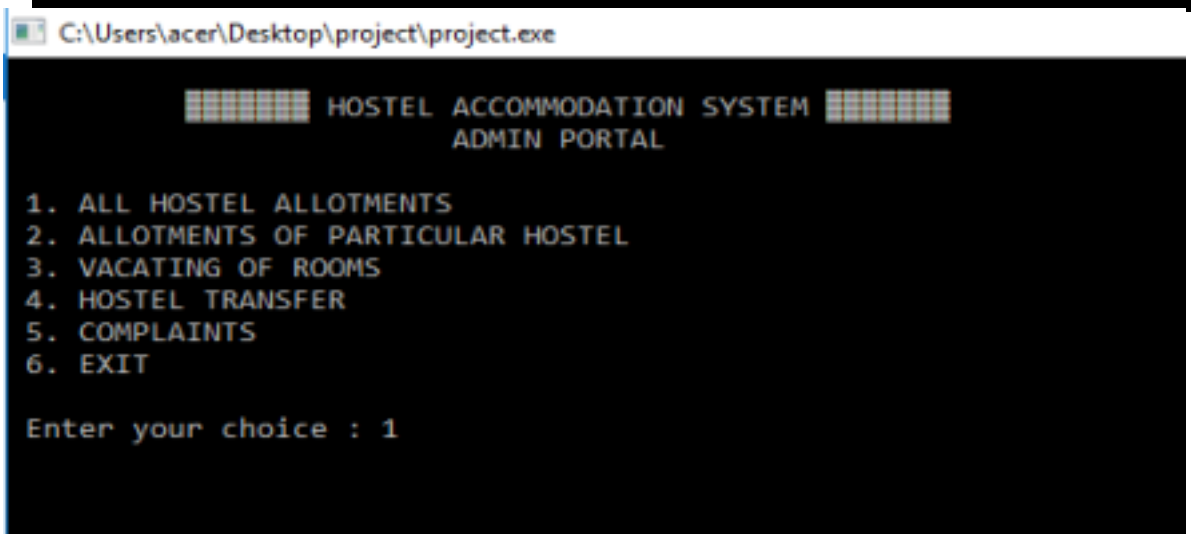


Registration form

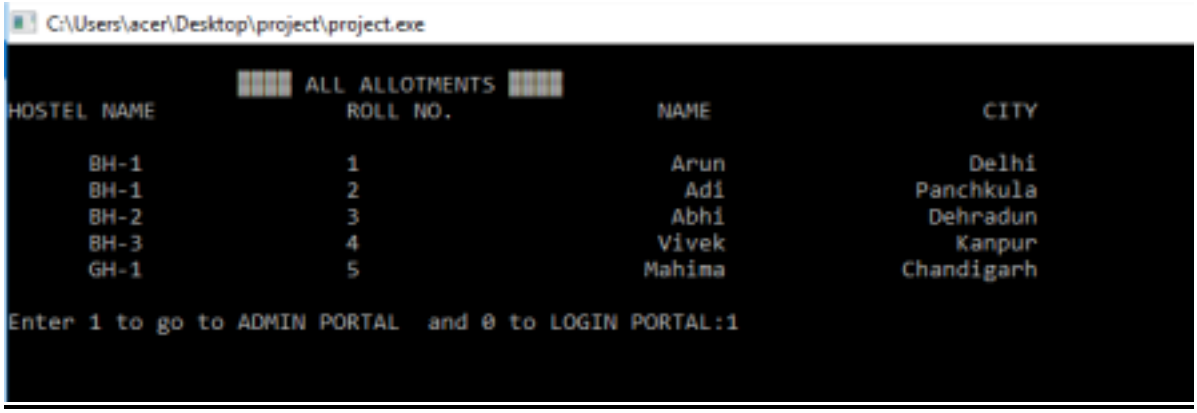


Administrator login

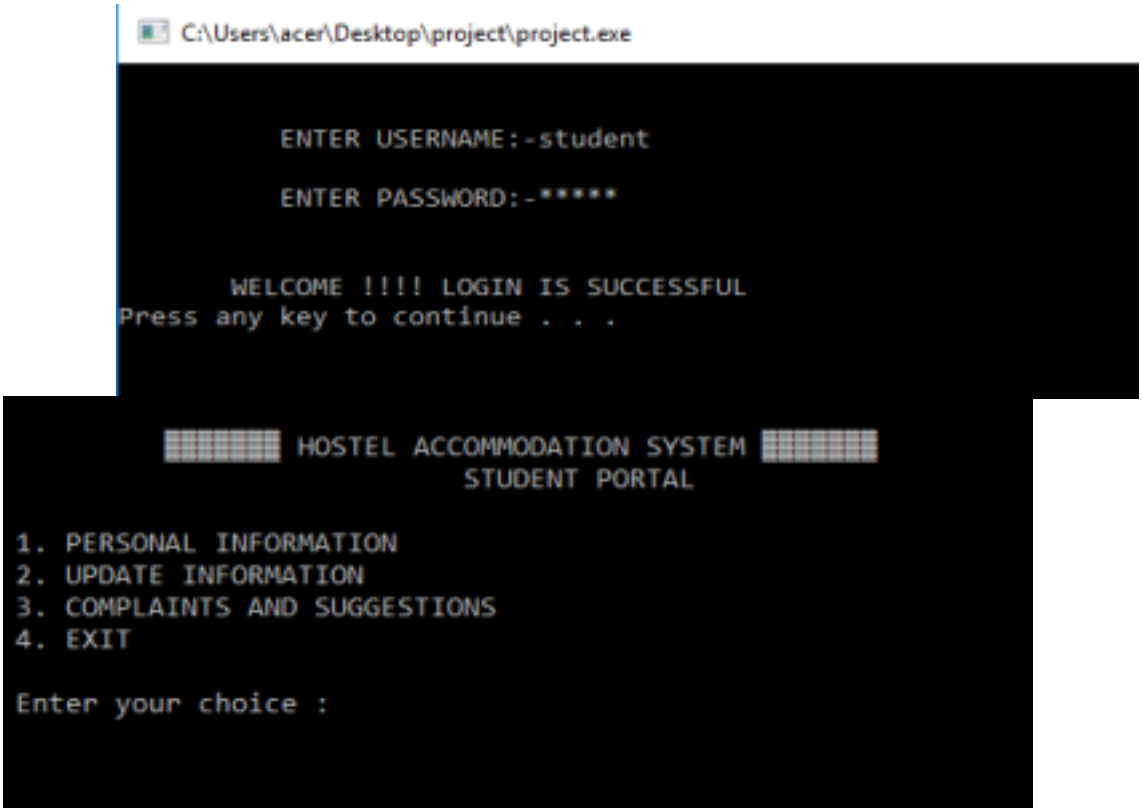




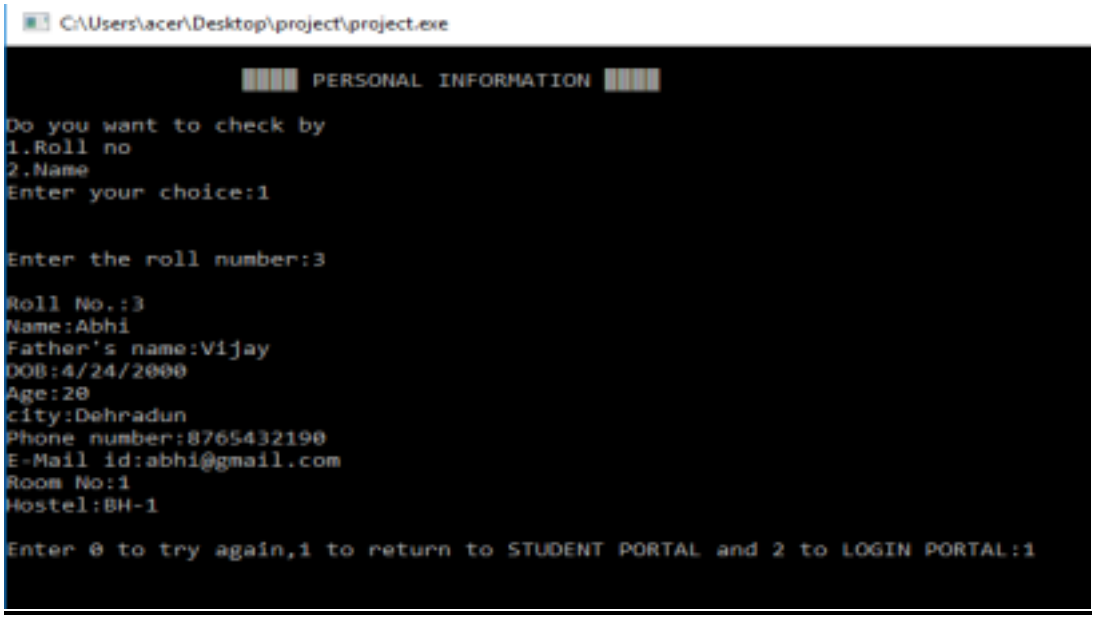
All hostels allotments



Student login



Personal information



8. CONCLUSION

The project developed using C Programming language is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement.

The expanded functionality of today's software requires an appropriate approach towards software development. This hostel management software is designed for people who want to manage various activities in the hostel. For the past few years, the number of educational institutions are increasing rapidly. Thereby the number of hostels are also increasing for the accommodation of the students studying in institution. And hence there is a lot of strain on the person who are running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly.

9. REFERENCES

- www.academia.edu
- https://code-projects.org
- www.codewithc.com
- www.tutorialspoint.com