Siddaganga Institute of Technology, Tumakuru

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi,
Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

A Report on Open Ended Problem titled "HOSTEL MANAGEMENT"

submitted
in the partial fulfillment of the requirements for III semester,
Data structures Laboratory
Bachelor of Engineering
in
Computer Science and Engineering

by

SINCHANA NM. 1SI21CS101

S. NIVEDITHA 1SI21CS088

Department of Computer Science & Engineering
(Program Accredited by NBA)

Siddaganga Institute of Technology
B.H Road, Tumakuru-572 103, Karnataka, India.
Web:www.sit.ac.in

March, 2023

Siddaganga Institute of Technology, Tumakuru

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

Department of Computer Science and Engineering

(Program Accredited by NBA)

CERTIFICATE

This is to certify that open ended problem titled "HOSTEL MANAGEMENT" is a bonafide work carried out by **SINCHANA NM(1SI21CS101**) **S NIVEDITHA(1SI21CS088),** of III semester ,**Bachelor of Engineering in Computer Science and Engineering** of the **SIDDAGANGA INSTITUTE OF TECHNOLOGY** (An Autonomous Institution, affiliated to VTU, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified) during the academic year 2022-2023.

Name of the Examiners **Signature with Date**

1. **Mrs. Kavitha M.**
2. **Mrs. Nousheen Taj**

ABSTRACT

The abstract for a hostel management system using singly linked list in C would describe a software application designed to automate the management of a hostel. The system would be implemented in C programming language and utilize a singly linked list data structure to store information about hostel rooms, students, and other relevant details.

The system would be designed to allow hostel administrators to easily manage and monitor various aspects of the hostel, such as room allocation, room occupancy, student information, and billing. The system would also allow students to view their room and fee details, submit requests for room changes, and make payments online.

The singly linked list data structure would be used to store information about rooms and students in a structured and efficient manner, allowing for easy retrieval and modification of data as required. The system would also implement various algorithms and sorting techniques to optimize the performance of the system and reduce the processing time required for various tasks.

Overall, the hostel management system using singly linked list in C would provide an efficient and user-friendly solution for managing hostel operations and would be a valuable tool for both hostel administrators and students.

Data Structures used:

- Singly linked lists
- Arrays

TABLE OF CONTENTS

Sl.No.	Contents	Page No.
1	Introduction	5
2	Problem Statement and Objectives	7
3	Design & Implementation	8
4	Results	22
5	Conclusion	24
6	References	25

CHAPTER 1

INTRODUCTION

Hostel management system using singly linked lists in C is a software application that automates the management of hostels. It provides an efficient and user-friendly solution for managing hostel operations and is a valuable tool for both hostel administrators and students. The system is implemented in the C programming language and uses a singly linked list data structure to store information about hostel rooms, students, and other relevant details.

The hostel management system is designed to simplify the process of managing a hostel. It provides hostel administrators with an easy-to-use interface that allows them to manage room allocation, room occupancy, student information, and billing. The system also enables students to view their room and fee details, submit requests for room changes, and make payments online.

The singly linked list data structure is used to store information about rooms and students in a structured and efficient manner. The linked list provides a way to store data dynamically, allowing for the creation and deletion of nodes as required. This feature of singly linked lists is particularly useful in hostel management systems as the number of students and rooms can vary, and the system needs to be able to handle these changes in real-time.

The system implements various algorithms and sorting techniques to optimize its performance and reduce the processing time required for various tasks. For instance, the system may use algorithms such as bubble sort or quicksort to sort student and room details in ascending or descending order, making it easier for hostel administrators to retrieve information.

In conclusion, hostel management system using singly linked lists in C is an efficient and effective solution for managing hostel operations. It simplifies the process of managing a hostel and provides hostel administrators and students with an easy-to-use interface for managing their activities

- Administrator

  Password

  Main Menu
- User

  Own Information

  Hostel Rule Regulation

  Food menu

  Main menu

- Visitor

  Hostel 1

  Hostel 2

  Hostel 3

  Hostel 4

  Hostel 5

  Hostel 6

  Sort by Distance

  Sort by Cost

  Sort by Quality

  Hostel Rules

  special_service

CHAPTER 2

PROBLEM STATEMENT AND OBJECTIVES

2.1 PROBLEM STATEMENT

The problem statement of hostel management in C using linked lists is the efficient management of hostel operations, including room allocation, occupancy, student information, and billing.

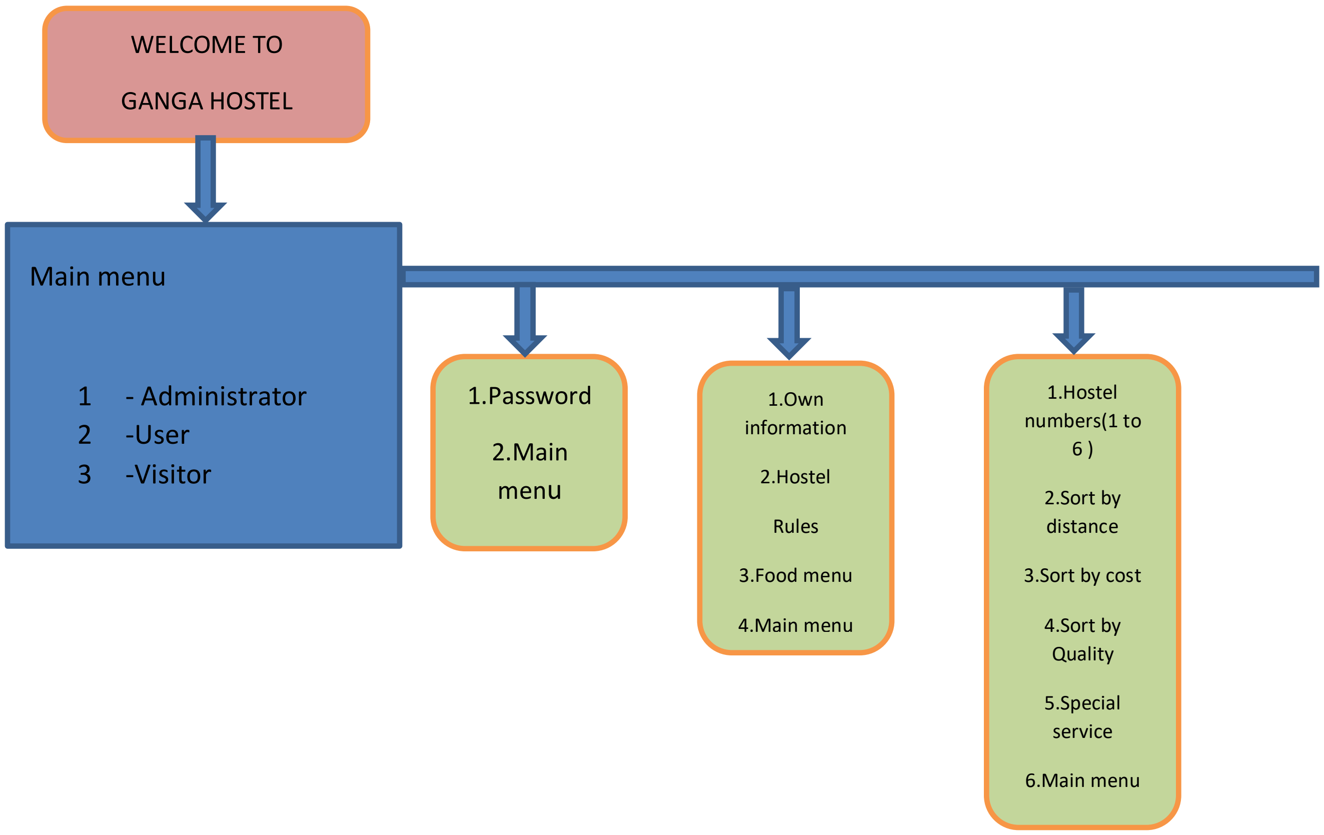
1.2 OBJECTIVES

- To automate the process of managing hostel operations, including room allocation, occupancy, student information, and billing.
- To provide hostel administrators with an easy-to-use interface for managing hostel operations and accessing student and room details.
- To enable students to view their room and fee details, submit requests for room changes, and make payments online.
- To store information about hostel rooms, students, and other relevant details in a structured and efficient manner using the singly linked list data structure.
- To implement various algorithms and sorting techniques to optimize the performance of the system and reduce the processing time required for various tasks.
- To enable hostel administrators to generate reports on hostel operations, including occupancy rates, billing statements, and student details.
- To provide a secure and reliable system for managing hostel operations, including data backup and recovery mechanisms.

CHAPTER 3

DESIGN AND IMPLEMENTATION

3.1 DESIGN OF ALGORITHM



3.2 IMPLEMENTATION

```
#include<stdio.h>

int a, b, c, d, e, f,sf=0,ff=0;
int i,c,s=-1,g=1;
int temp,admin=4321,user=1234;
struct node
{
int id;
char name[50];
int mobile_number[20];
int seat_fee;
int food_fee;
int total;
struct node *next;
}*start,*head,*tail;

int main()
{
welcome();
main_menu();
}
welcome()
{   printf("                  ------------------------------------------------------------      \n");
printf("\t\t                    Welcome To Ganga Hostel     \n");
printf("                 -------------------------------------------------------      \n");
}
main_menu()
{

printf("\n\n\t\t     press 1 for Administrator         \n");
printf("\t\t     press 2 for User                \n");
printf("\t\t     press 3 for Visitor             \n");
printf("Enter your Choice:");
scanf("%d",&a);
if(a == 1)
{
printf("Welcome Administrator\n");
Administrator();
}
else if(a == 2)
{
printf("Welcome User\n");
reg_user();
}
else if(a == 3)
{
printf("Welcome visitor\n");
visitor();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
main_menu();
}
}

Administrator()
{
printf("Enter Password for log in:\n");
printf("or\n");
printf("Press 0 for Main Menu or press 1 for modifying food menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp == admin)
{
admin_interface();
}
else if(temp == 0)
{
main_menu();
}
else if(temp==1)
{
food_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
Administrator();
}
}

admin_interface()
{
```

```
printf("\t\t|     Press 1 for student information     |\n");
printf("\t\t|     Press 2 for Account                 |\n");
printf("\t\t|     Press 3 for Employee                |\n");
printf("\t\t|     Press 0 for Main Menu               |\n");
printf("Enter your Choice: ");
scanf("%d",&temp);
if(temp == 1)
{
student_info();
printf("\n\n");
admin_interface();
}
else if(temp == 2)
{
printf("\n\t\t\t\t  Total Seat fee : %d          \n", sf);
printf("\t\t\t---------------------------------------\n");
printf("\n\t\t\t\t  Total Food fee : %d          \n", ff);
printf("\t\t\t---------------------------------------\n");
printf("\n\t\t\t\t  Total amount is: %d          \n", sf+ff);
printf("\n\n");
admin_interface();
}
else if(temp == 3)
{
printf("\n\n\t|| Hostel in charge  ||  Cooker  ||  Gate keeper  ||\n");
printf("\t||    Tapasy Rabeya       ||  Fatema banu ||   Jolil Mia       ||\n");
printf("\t||    Sanjida Ferdaus     ||  Bilkis banu ||   Mamun Mia       ||\n");
printf("\t||    Aklima Akter        ||  Sokina Banu ||   Saiful hasan    ||\n");
printf("\n\n");
admin_interface();
}
else if(temp == 0)
{
main_menu();
printf("\n\n");
}
else
{
printf("Invalid Input\n");
admin_interface();
}
}

student_info()
```

```
{
printf("\n\n\t\t\tTotal number of Student is:%d",s);
head=start->next;
while(head!=NULL)
{
printf("\n\n %d\n %s\n %d\n %d\n %d\n\n ",head->id,head->name,head->mobile_number,head->seat_fee,head->food_fee);
head=head->next;
}
}

user_interface()
{
printf("\n\n\t\t\tEnter Password for log in:\n");
printf("\t\t\tor\n");
printf("\t\t\tPress 0 for Main Menu\n\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp == user)
{
reg_user();
}
else if(temp == 0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
user_interface();
}
}

reg_user(){

printf("\t\t\t    press 1 for own information         \n");
printf("\t\t\t    press 2 for hostel rule regulation  \n");

printf("\t\t\t    press 4 for main menu              \n");
printf("\n\nEnter your choice: ");
scanf("%d",&i);
if(i==1){
search();
printf("\n\n");
reg_user();
}
else if(i==2){
rulls();
printf("\n\n");
reg_user();
}

else if(i==4){
main_menu();
printf("\n\n");
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
reg_user();
}
}

search(){
int n;
head=start;
printf("\t\t\t    Enter Your ID or press 0 for back   \n");
printf("\n\nEnter Your ID : ");
scanf("%d",&n);

while (head->next!=NULL)
{
if (head->id==n)
{
printf("\t\t   ID        : %d\n",head->id);
printf("\t\t   Name      : %s\n",head->name);
printf("\t\t   Mobile no : 01%d\n",head->mobile_number);
printf("\t\t   Seat fee  : %d\n",head->seat_fee);
printf("\t\t   Food fee  : %d\n",head->food_fee);
return 0;
}
head=head->next;
}
if(n==0)
{
reg_user();
}
```

```
else
{
printf("\n\n\t\t\tYour ID is Not Match :%d\n\t\t\t\tTRY AGAIN\n)",n);
return search ();
}
}

visitor()
{
printf("press 1 for Hostel 1\n");
printf("press 2 for Hostel 2\n");
printf("press 3 for Hostel 3\n");
printf("press 4 for Hostel 4\n");
printf("press 5 for Hostel 5\n");
printf("press 6 for Hostel 6\n");
printf("press 7 for Sort by Distance\n");
printf("press 8 for Sort by Cost\n");
printf("press 9 for Sort by Quality\n");
printf("press 11 Hostel Rules\n");
printf("press 11 special_service\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&b);
if(b==1)
{
hostel_1();
}
else if(b==2)
{
hostel_2();
}
else if(b==3)
{
hostel_3();
}
else if(b==4)
{
hostel_4();
}
else if(b==5)
{
hostel_5();
}
else if(b==6)
```

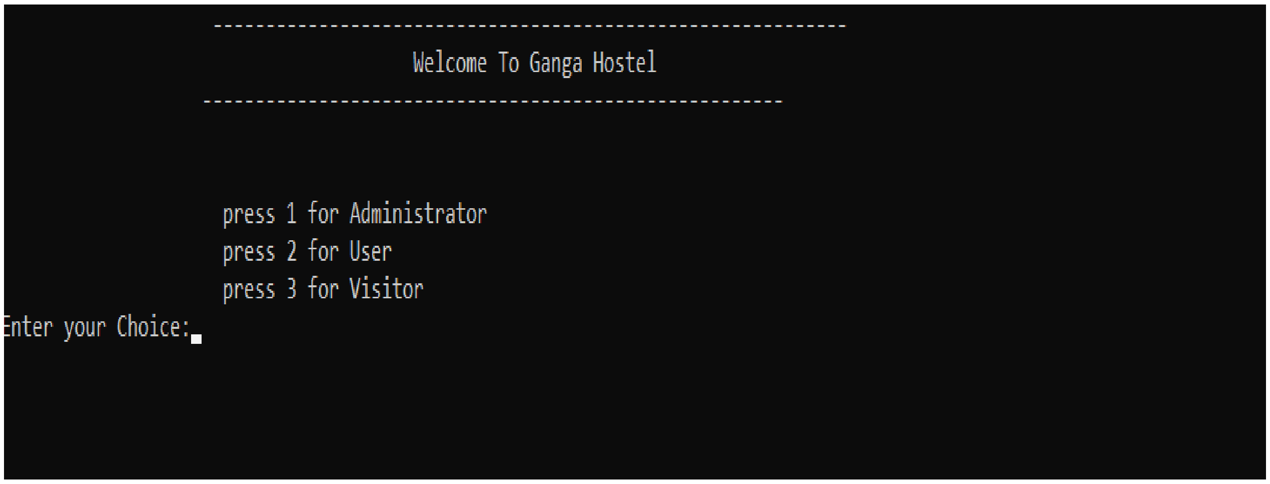
```
{
hostel_6();
}
else if(b==7)
{
hostel_1();
}
else if(b==8)
{
hostel_1();
}
else if(b==9)
{
hostel_1();
}
else if(b==10)
{
rulls();
}
else if(b==11)
{
special_service();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
visitor();
}
}
hostel_1()
{
printf("welcome to Hostel 1\n");
printf("Cost 5000\n");
printf("Distance 5KM\n");
printf("Quality Good\n");
printf("Press 1 for registration\n");
printf("Press 2 for Back\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp==1)
{
registration();
```

```
}
else if(temp==2)
{
visitor();
}
else if(temp==0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
hostel_1();
}
}
hostel_2()
{
printf("welcome to Hostel 2\n");
printf("welcome to Hostel 1\n");
printf("Cost 5000\n");
printf("Distance 5KM\n");
printf("Quality Good\n");
printf("Press 1 for registration\n");
printf("Press 2 for Back\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp==1)
{
registration();
}
else if(temp==2)
{
visitor();
}
else if(temp==0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
```

```
hostel_2();
}
}
hostel_3()
{
printf("welcome to Hostel 3\n");
printf("welcome to Hostel 1\n");
printf("Cost 5000\n");
printf("Distance 5KM\n");
printf("Quality Good\n");
printf("Press 1 for registration\n");
printf("Press 2 for Back\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp==1)
{
registration();
}
else if(temp==2)
{
visitor();
}
else if(temp==0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
hostel_3();
}
}
hostel_4()
{
printf("welcome to Hostel 4\n");
printf("welcome to Hostel 1\n");
printf("Cost 5000\n");
printf("Distance 5KM\n");
printf("Quality Good\n");
printf("Press 1 for registration\n");
printf("Press 2 for Back\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp==1)
{
registration();
}
else if(temp==2)
{
visitor();
}
else if(temp==0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
hostel_4();
}
}
hostel_5()
{
printf("welcome to Hostel 5\n");
printf("welcome to Hostel 1\n");
printf("Cost 5000\n");
printf("Distance 5KM\n");
printf("Quality Good\n");
printf("Press 1 for registration\n");
printf("Press 2 for Back\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp==1)
{
registration();
}
else if(temp==2)
{
visitor();
}
else if(temp==0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
hostel_5();
}
}
hostel_6()
{
printf("welcome to Hostel 6\n");
printf("welcome to Hostel 1\n");
printf("Cost 5000\n");
printf("Distance 5KM\n");
printf("Quality Good\n");
printf("Press 1 for registration\n");
printf("Press 2 for Back\n");
printf("Press 0 for Main Menu\n");
printf("Enter your Choice:");
scanf("%d",&temp);
if(temp==1)
{
registration();
}
else if(temp==2)
{
visitor();
}
else if(temp==0)
{
main_menu();
}
else
{
printf("Invalid Input\n");
printf("Try again\n");
hostel_6();
}
}
registration()
{
printf(" \n\n\t\t\t:-) ;-) Create your Account !!!!\n\n");
printf(" \n\n\t\t\t:-) ;-) Create your Account !!!!\n\n");
printf(" \n\n\t\t\t:-) ;-) Create your Account !!!!\n\n");
printf("\n\n");
if(g==1)
{
start=(struct node*)malloc(sizeof(struct node));
printf("Enter the id: ");
scanf("%d",&start->id);
printf("Enter your name: ");
scanf("%s",&start->name);
printf("Enter your mobile number:  ");
scanf("%d",&start->mobile_number);
printf("Enter seat fee: ");
scanf("%d",&start->seat_fee);
printf("Enter food fee: ");
scanf("%d",&start->food_fee);
start->next=NULL;
head=start;
}
else
{
tail=(struct node*)malloc(sizeof(struct node));
printf("Enter the id: ");
scanf("%d",&tail->id);
printf("Enter your name: ");
scanf("%s",&tail->name);
printf("Enter your mobile number:  ");
scanf("%d",&tail->mobile_number);
printf("Enter seat fee: ");
scanf("%d",&tail->seat_fee);
printf("Enter food fee: ");
scanf("%d",&tail->food_fee);
tail->next=NULL;
head->next=tail;
head=tail;
sf=sf+tail->seat_fee;
ff=ff+tail->food_fee;
}
g=g+1;
s=s+1;
confirm();
}

special_service()
{
printf("\n\n");
printf("\t\n\t| Begume Kokaiya girls hostle two|\n\t  \n 1. single room with attached bathroom and balcony.\n 2. room service\n 3. cost : 5000tk only.\n\n");
printf("\t\n\t| Begume Kokaiya girls hostle one  |\n\t  \n 1. single room with attached bathroom and balcony.\n 2. cost : 4000tk only.\n\n");
printf("\t\n\t| Begume Kokaiya girls hostle three|\n\t  \n 1. single room with attached bathroom and balcony.\n 2. cost : 4000tk only.\n\n");
}


rulls()
{
printf("\n\t1.you have to pay hostel charge every month within first 10 days.\n  ");
printf("\tafter 10 day you have to pay 20tk for each day & 200tk for each month.\n");
printf("\t2.you have to back in hostel in due time. \n\t\t\tsummer: last time 7.00pm\n\t\t\twinter: last time 6.00pm \n");
printf("\t3.before 6 month you cant leave the hostel.\n");
printf("\t4.if you want to leave this hostel you have to inform the hostel \n\tauthority before 2 month.\n");
printf("\t5.if you have computer or laptop, we have to pay 100tk as fee.\n");
printf("\t6.only your two local guardian can come to meet with you\n");
}
food_menu()
{
int k;
char
s1[20],s2[20],s3[20],m1[20],m2[20],m3[20],t1[20],t2[20],t3[20],w1[20],w2[20],w3[20],th1[20],th2[20],th3[20],f1[20],f2[20],f3[20],sa1[20],sa2[20],sa3[20];
printf("\n\n");
printf("Enter the food menu for following days(Morning,Afternoon,Night\n");
printf("Sunday \n");
scanf("%s%s%s",s1,s2,s3);
printf("Monday \n");
scanf("%s%s%s",m1,m2,m3);
printf("Tuesday\n");
scanf("%s%s%s",t1,t2,t3);
printf("Wednesday\n");
scanf("%s%s%s",w1,w2,w3);
printf("Thursday\n");
scanf("%s%s%s",th1,th2,th3);
printf("Friday\n");
scanf("%s%s%s",f1,f2,f3);
printf("Saturday\n");
scanf("%s%s%s",sa1,sa2,sa3);
printf("Day            \t\t Morning     \t\t   Afternoon      \t\t      Night\n");
printf("Sunday          \t\t%s          \t\t   %s          \t\t      %s   \n ",s1,s2,s3);
printf("Monday          \t\t%s          \t\t   %s          \t\t      %s   \n ",m1,m2,m3);
printf("Tuesday         \t\t%s          \t\t   %s          \t\t      %s   \n ",t1,t2,t3);
printf("Wednesday       \t\t%s          \t\t   %s          \t\t      %s   \n ",w1,w2,w3);
printf("Thursday        \t\t%s          \t\t   %s          \t\t      %s   \n ",th1,th2,th3);
printf("Friday          \t\t%s          \t\t   %s            \t\t      %s    \n ",f1,f2,f3);
printf("Saturday         \t\t%s          \t\t   %s            \t\t      %s    \n ",sa1,sa2,sa3);
printf("\n\nif u want to modify press 1 or else  0 for main menu\n");
scanf("%d",&k);
if(k==0)
main_menu();
else
food_menu();
}


confirm()
{
printf(" \n\n\t\t\t:-) ;-) Congratulation !!!!\n\n");
printf(" \n\n\t\t\t:-) ;-) Your Account is Create !!!!\n\n");
printf("\t\t  You become a member of our Hostel.\n");
main_menu();
}
```

CHAPTER 4

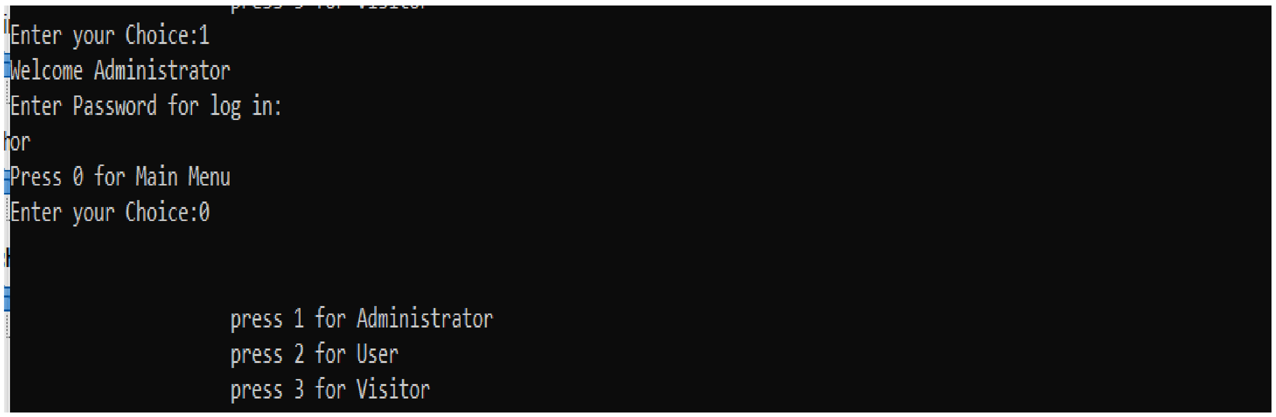
RESULTS



```
                   --------------------------------------------------------
                                  Welcome To Ganga Hostel
                  ----------------------------------------------------



                    press 1 for Administrator
                    press 2 for User
                    press 3 for Visitor
Enter your Choice:
```

Figure 4.1:welcome screen

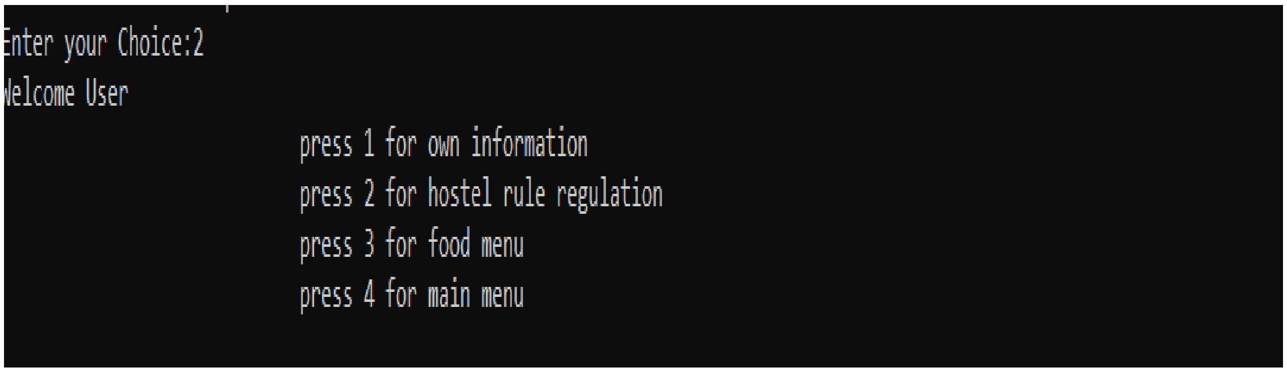
Fig 4.1 shows the welcome screen here we get 3 types of entries i.e Administrator,User,Visitor



```
Enter your Choice:1
Welcome Administrator
Enter Password for log in:
or
Press 0 for Main Menu
Enter your Choice:0



                    press 1 for Administrator
                    press 2 for User
                    press 3 for Visitor
```

Figure 4.2:Adminstrator

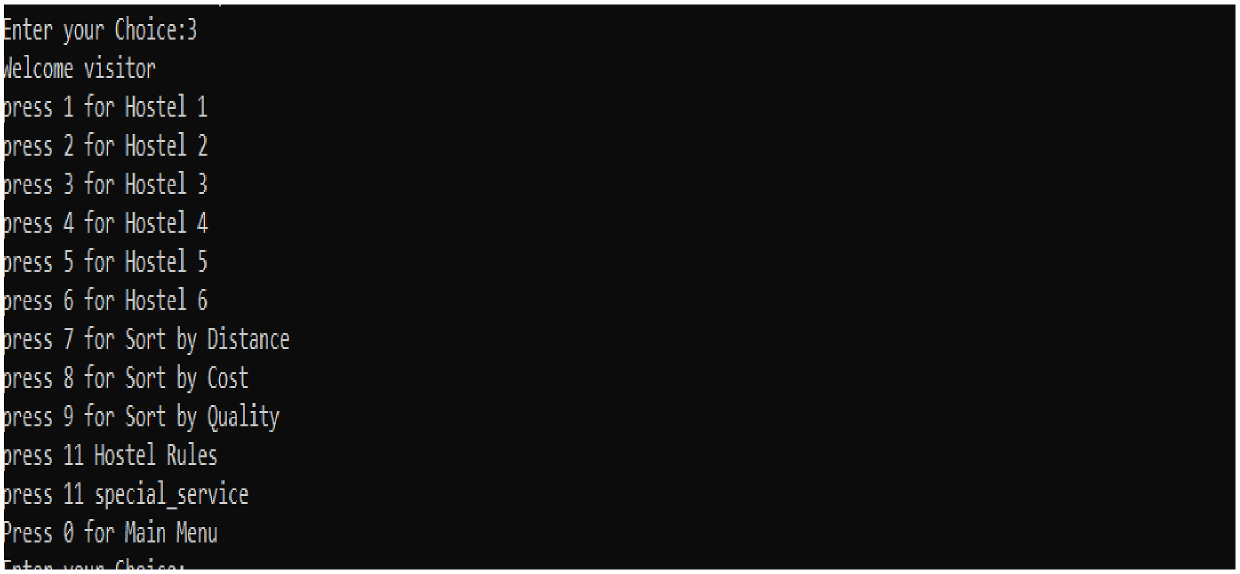
Fig 4.2 shows the 2 options i.e password for login or zero for main menu



```
Enter your Choice:2
Welcome User
                    press 1 for own information
                    press 2 for hostel rule regulation
                    press 3 for food menu
                    press 4 for main menu
```

Figure 4.3 :User

Fig 4.3 shows the 4 options i.e own information,hostel rule regulation,food menu,main menu



```
Enter your Choice:3
Welcome visitor
press 1 for Hostel 1
press 2 for Hostel 2
press 3 for Hostel 3
press 4 for Hostel 4
press 5 for Hostel 5
press 6 for Hostel 6
press 7 for Sort by Distance
press 8 for Sort by Cost
press 9 for Sort by Quality
press 11 Hostel Rules
press 11 special_service
Press 0 for Main Menu
```

Figure 4.4:Visitor

Fig 4.4 shows the options i.e from Hostels from 1 to 6,Sort by Distance,Sort by Distance,Sort by cost,Sort by quality,Hostel Rules,Special service,Main menu

CONCLUSION

In conclusion, hostel management using linked lists in C is an effective and efficient way of automating hostel operations. The use of linked lists enables hostel administrators to store and manage student and room details in a structured and dynamic manner, allowing for real-time updates and changes.

The system provides an easy-to-use interface for hostel administrators to manage various aspects of hostel operations, including room allocation, occupancy, student information, and billing. It also allows students to view their room and fee details, submit requests for room changes, and make payments online.

Furthermore, the implementation of various algorithms and sorting techniques helps to optimize the system's performance, reducing the processing time required for various tasks. This makes it possible for hostel administrators to generate reports on hostel operations and access student and room details quickly.

Overall, the use of linked lists in hostel management is a valuable tool for both hostel administrators and students. It provides an efficient and user-friendly solution for managing hostel operations and is scalable to handle changes in the number of students and rooms over time.

REFERENCES

1.YouTube and Website links

- https://youtu.be/1JH0ErOL74U
- https://youtu.be/xI7jCGXBiPM

2.ANCI, Programming in C, 7th Edition- E.Balagurusamy.

3.Data Structures using C- Yedidyah Langsam, Moshe J., Augenstein, Aaron M. Tenenbaum.