KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Electronics and Communication

COURSE PROJECT REPORT

For

Ability Enhancement Course

(Software Development Concepts – 21AECEC381)

Title of the project

Hotel Management System Using C

Submitted by

SONAL S POOJARY(2GI21EC144)

Guide

Prof. Maya Upadhye

2022 – 2023

Institute Vision

KLS Gogte Institute of Technology shall stand out as an institution of excellence in technical education and in training individuals for outstanding caliber, character coupled with creativity and entrepreneurial skills**.**

Mission

To train the students to become quality engineers with high standards of professionalism and ethics who have positive attitude, a perfect blend of techno-managerial skills and problem-solving ability with an analytical and innovative mindset.

Department Vision:

The Electronics and Communication Engineering department shall impart quality technical education and entrepreneurship skills to develop creative individuals to face changing global scenario.

Mission:

To augment the national talent pool, with Electronics and Communication Engineers having all-encompassing technical knowledge, principled practices and nationalistic outlook.

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Electronics and Communication

Certificate of Completion

Certified that the Course Project Report entitled Hotel Management System Using C at KLS Gogte Institute of Technology is a bonafide work carried out by Sonal S Poojary (2GI21EC144) , in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication**.

The project report has been approved as it satisfies the academic requirements in respect of Ability Enhancement Course prescribed for the said Degree.

Signature of the Guide Signature of the HOD Signature of the Principal

SEE Viva-Voce

	Name of the examiners	**Date of Viva -voce**	**Signature**
1.			
2.			
3.			

DECLARATION BY THE STUDENT

I**, Sonal S Poojary**, hereby declare that the Ability Enhancement Course Project report (Subject Name with code) entitled **Hotel Management System Using C** submitted by me to KLS Gogte Institute of Technology, Belagavi, in partial fulfillment of the Degree of Bachelor of Engineering in **Electronics and Communication** is a record of the Ability Enhancement Course carried out at **KLS Gogte Institute of Technology**. This report is for the academic purpose.

Place: Name of the student: Sonal S Poojary
Date: 25/03/2023 USN: 2GI21EC144

Signature of the student

ACKNOWLEDGEMENT

On this great occasion of accomplishment of our project titled **Hotel Management System Using C** , I would like to sincerely express my gratitude to Name of the guide, who has supported us throughout the completion of this project.

I would also like to express my gratitude to our principal, Prof. D.A. Kulkarni and HOD ECE, Dr.Supriya Shanbagh, for providing all the required facilities for the completion of the project.

Name of the Student: Sonal S Poojary

ABSTRACT

This is a hotel management program written in C language. It enables the user to reserve a room, check visitors in and out, see rooms that are available, and get information about all guests who have checked in. The application employs a structure to record data about each room, including the room number, the name of the visitor, their cell phone number, their nationality, and the length of their stay. Each room's status is likewise recorded as either available (0) or occupied (1).

The program implements its features via a variety of functions. The user can view every available room by the display_available_rooms function. By inputting the room number and the appropriate visitor details, the user can reserve a room using the book_room function. If the room is occupied, the check_in function allows the user to check in a guest by providing the room number and subsequently displays the guest details. By inputting the room number and changing the state of the room to available, the check_out function enables the user to check out a visitor. All checked-in visitors' information is shown using the display_guests function.

Overall, this program offers fundamental hotel management system functions and it could be expanded to include more features like billing and room services.

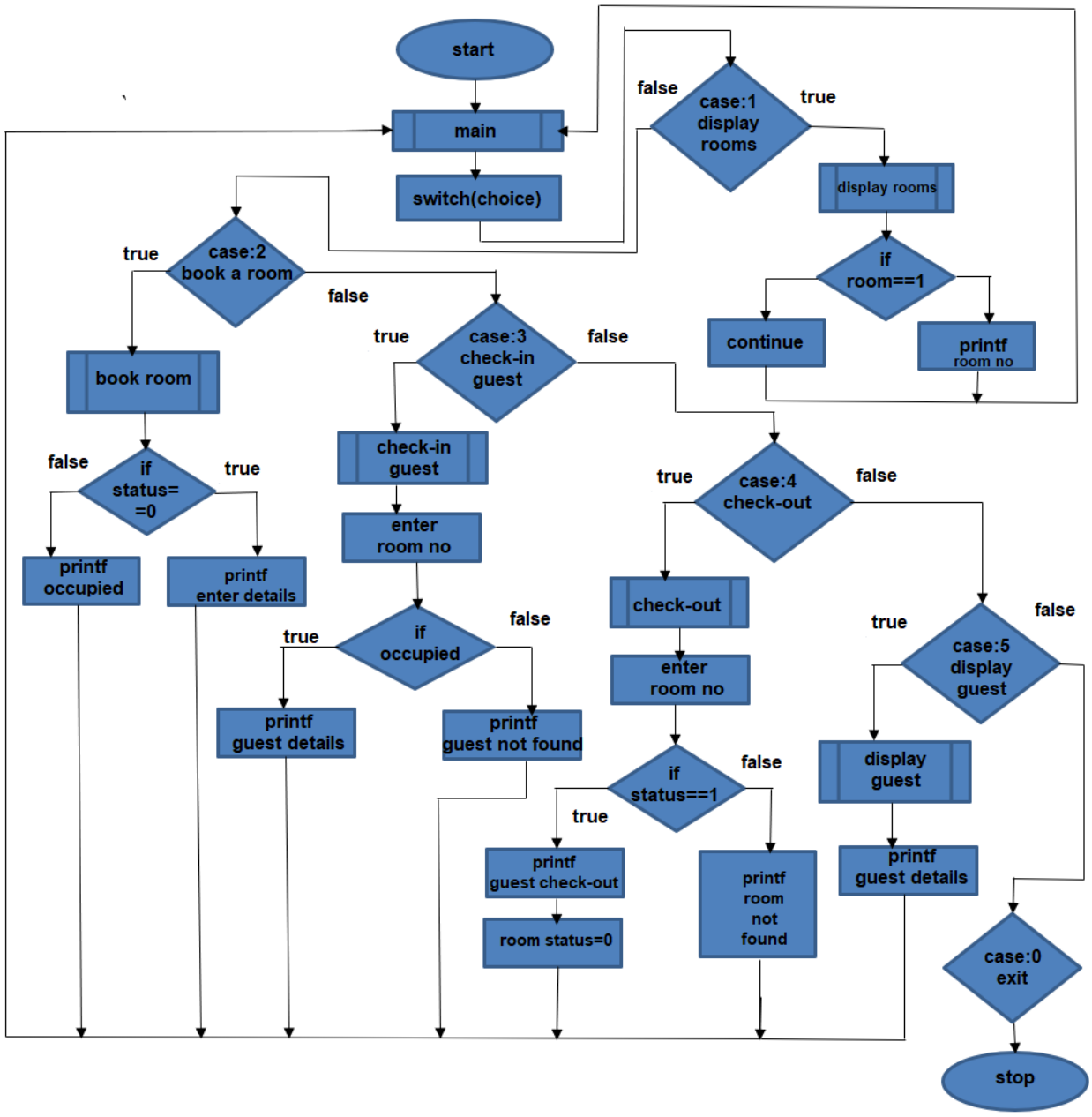
Table of Contents

Objectives of Ability Enhancement Course Software Development Concepts (21ECAEC381)

- To understand fundamental concepts of system software
- To understand multifunction programs using advanced concepts of C programming
- To provide an understanding of basic concepts of the Object-Oriented paradigm.

Problem Statement: Hotel Management System using C

METHODOLOGY (flowchart)



IMPLEMENTATION:

```
#include <stdio.h>
#include <string.h>
#include<time.h>
#include<stdlib.h>

// Structure for storing room information
typedef struct {
    int room_number;
    char guest_name[30];
    int status; // 0 for available, 1 for occupied
            char mobile[15];
            char nat[15];
            char period[10];
} Room;

// Function to display available rooms
void display_available_rooms(Room rooms[], int num_rooms) {
    system("cls");
    printf("\n\n\nAvailable rooms:\n\n");
    for (int i = 0; i < num_rooms; i++) {
        if (rooms[i].status == 0) {
            printf("%d\t", rooms[i].room_number);
        }
```

```
    }
}

// Function to book a room
void book_room(Room rooms[], int num_rooms) {
    int room_number;
    char guest_name[30];
    char mobile[15];
            char nat[15];
            char period[10];

    printf("\n\nEnter room number: ");
    scanf("%d", &room_number);

    // Check if room is available
    int found = 0;
    for (int i = 0; i < num_rooms; i++) {
        if (rooms[i].room_number == room_number) {
            if (rooms[i].status == 0) {
                found = 1;
                system("cls");
                printf("\nPlease Provide The Following Details To Complete Booking :-\n\n");
                printf("Enter guest name: ");
                scanf("%s", guest_name);
                strcpy(rooms[i].guest_name, guest_name);
```

```
            rooms[i].status = 1;
            printf("Mobile No. :");
            scanf("%s", mobile);
            strcpy(rooms[i].mobile, mobile);
            printf("Nationality :");
            scanf("%s", nat);
            strcpy(rooms[i].nat,nat);
            printf("Period Of Stay :");
            scanf("%s", period);
            strcpy(rooms[i].period,period);
            printf("\n\n\nRoom booked successfully.\n");
            printf(" \n Press any key to continue:");

        getch();
    system("cls");
        } else {
            printf("Room is already occupied.\n");
        }
        break;
      }
    }
    if (!found) {
      printf("Room not found.\n");
    }
}
```

```
// Function to check in a guest
void check_in(Room rooms[], int num_rooms) {
    int found=0;
    int room_number;
            system("cls");
    printf("\n\nEnter room number: ");
    scanf("%d", &room_number);

    // Check if room is occupied
    for (int i = 0; i < num_rooms; i++) {
        if (rooms[i].room_number == room_number) {
            if (rooms[i].status == 1) {
                found = 1;
                printf("\n\nGUEST DETAILS :\n");
                printf("\n------------------------\n\n");
                printf("Guest name      : %s\n", rooms[i].guest_name);
                printf("Mobile No.      : %s\n",rooms[i].mobile);
                printf("Nationality     : %s\n",rooms[i].nat);
                printf("Period Of Stay  : %s\n",rooms[i].period);
                printf(" \n Press any key to continue:");

            getch();
    system("cls");
            } else {
```

```
            printf("\n\nGuest Not Found.\n");
        printf(" \n Press any key to continue:");

            getch();
    system("cls");
            break;
        }
    }
    }
}
// Function to check out a guest
void check_out(Room rooms[], int num_rooms) {
    int room_number;
    system("cls");
    printf("Enter room number: ");
    scanf("%d", &room_number);

    // Check if room is occupied
    int found = 0;
    for (int i = 0; i < num_rooms; i++) {
        if (rooms[i].room_number == room_number) {
            if (rooms[i].status == 1) {
                found = 1;
                printf("Guest %s has checked out.\n", rooms[i].guest_name);
                strcpy(rooms[i].guest_name, "");
```

```
            rooms[i].status = 0;
        } else {

            printf("\n\nRoom is not occupied.\n");
        }
        break;
      }
   }
   if (!found) {
      printf("\nGuest not found.\n");
   }
   printf(" \n Press any key to continue:");
        getch();
   system("cls");
}
//mergefunction
void display_guests(Room rooms[], int num_rooms)
{
   system("cls");
   int found=0;
    printf("\n\nGUEST DETAILS :\n\n");

for (int i = 0; i < num_rooms; i++) {
        if (rooms[i].status == 1) {
              found=1;
```

```
            printf("\n------------------------\n\n");
            printf("\n||ROOM No.: %d\t",rooms[i].room_number);
            printf("||Guest name : %s\t", rooms[i].guest_name);
            printf("||Mobile No. : %s\t",rooms[i].mobile);
            printf("||Nationality: %s\t",rooms[i].nat);
            printf("||Period Of Stay  : %s\t",rooms[i].period);
        }
    }
    if(found==0)
       printf("NO RECORDs FOUND\n\n");
    printf(" \n Press any key to continue:");

        getch();
    system("cls");
       }
//mergefunction

int main() {
//merge
int i=0;
        time_t t;
        time(&t);
        int choice;

        system("cls");
```

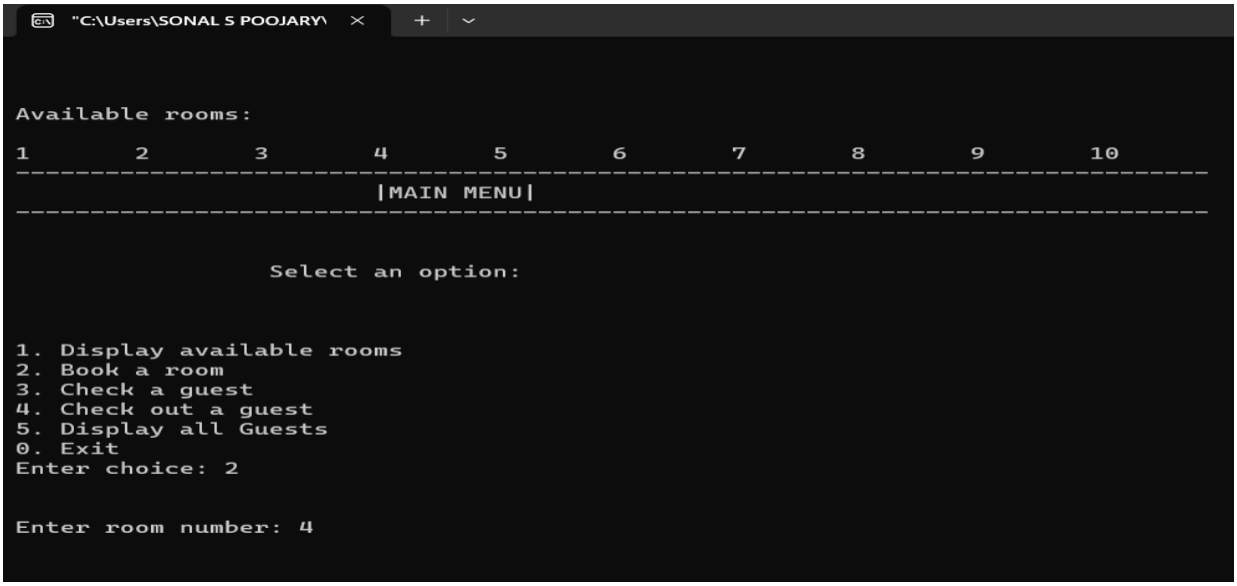
```
        printf("\t\t|                                |\n");
        printf("\t\t|      ------------------------------      |\n");
        printf("\t\t|       WELCOME TO HOTEL EIFA          |\n");
        printf("\t\t|      ------------------------------      |\n");


                for(i=0;i<80;i++)
                printf("-");
           printf("\nDate and Time : %s",ctime(&t));
           for(i=0;i<80;i++)
                printf("-");
        printf(" \n Press any key to continue:");
        getc();
system("cls");
system("cls");
                system("cls");
Room rooms[10];
for (int i = 0; i < 10; i++) {
    rooms[i].room_number = i + 1;
    strcpy(rooms[i].guest_name, "");
    rooms[i].status = 0;
}
do{
    printf("\n");
    for(i=0;i<80;i++)
```

```
                printf("-");
                printf("\n");
                printf("\t\t\t|MAIN MENU|  \n");
                for(i=0;i<80;i++)
                printf("-");
                printf("\n");
                printf("\n\n\t\t Select an option:");
                printf("\n");
//merge
    // Initialize rooms
        printf("\n\n\n");
        printf("1. Display available rooms\n");
        printf("2. Book a room\n");
        printf("3. Check a guest\n");
        printf("4. Check out a guest\n");
        printf("5. Display all Guests\n");
        printf("0. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                display_available_rooms(rooms, 10);
                break;
            case 2:
```

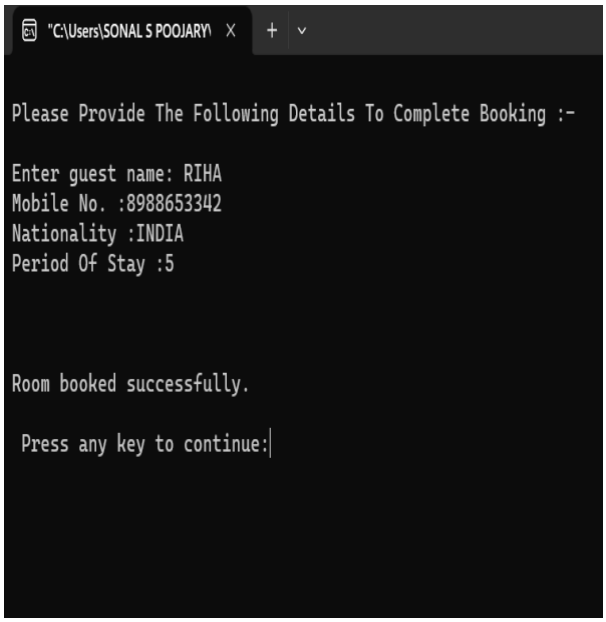
```
            book_room(rooms, 10);
            break;
        case 3:
            check_in(rooms, 10);
            break;
        case 4:
            check_out(rooms, 10);
            break;
        case 5 :
            display_guests(rooms,10);
            break;
        case 0:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice.\n");
            system("cls");
            break;
      }
   } while (choice != 0);

   return 0;
}
```

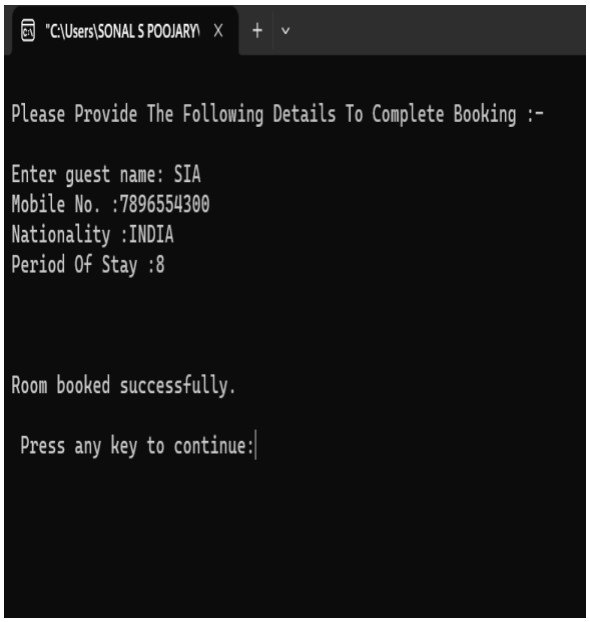
SAMPLE INPUT/OUTPUT SCREENSHOTS

1. Display available rooms

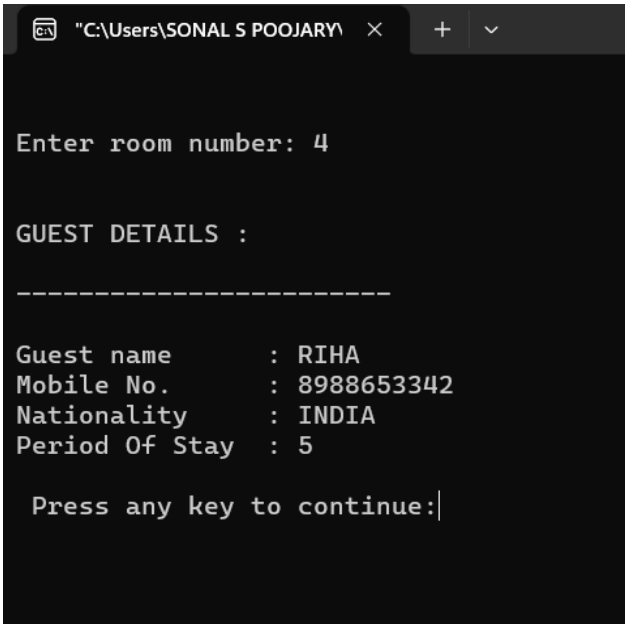


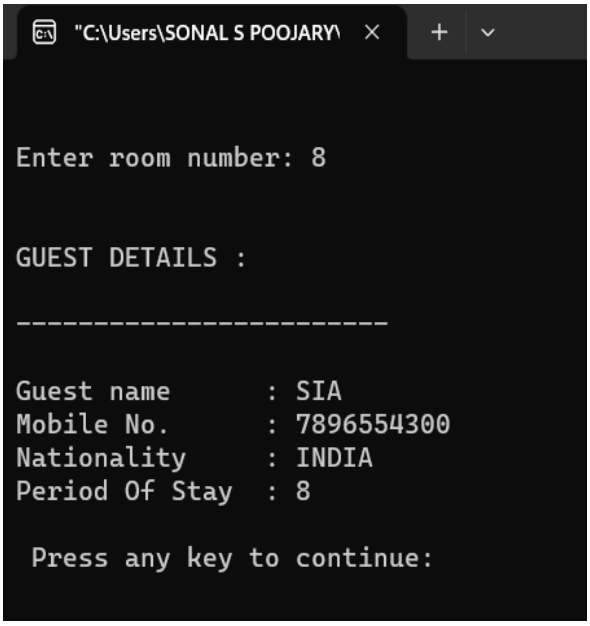
2. Book a room



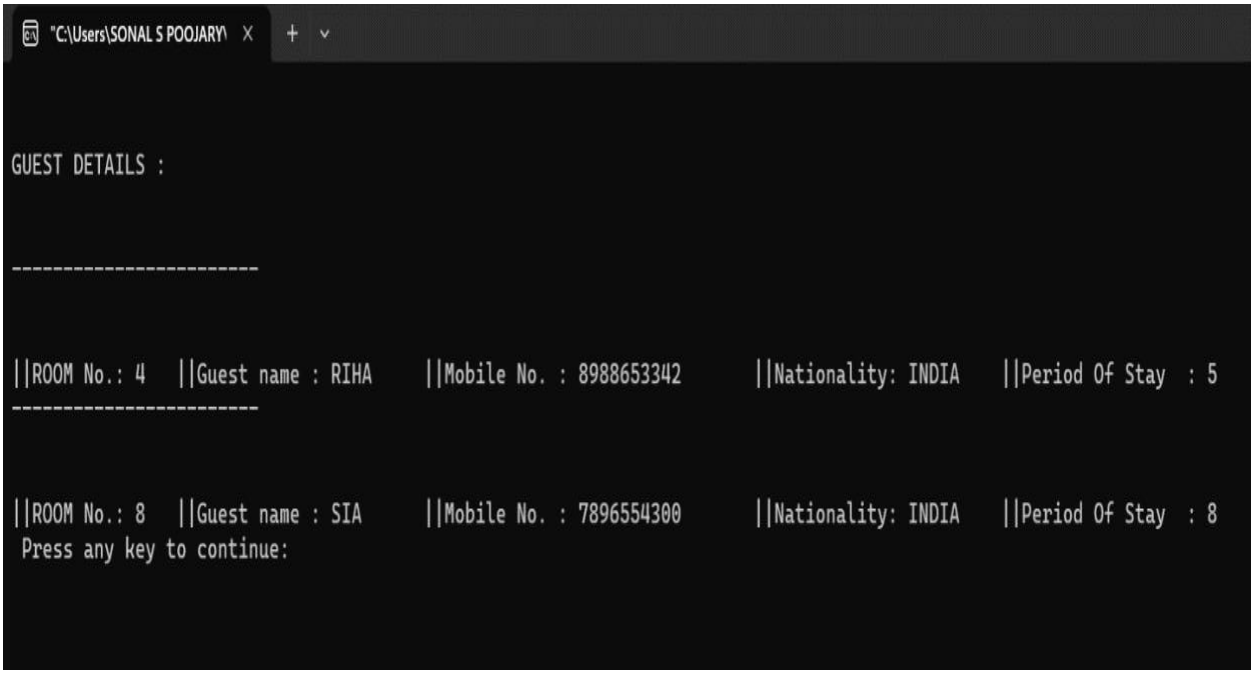


3. Check a Guest

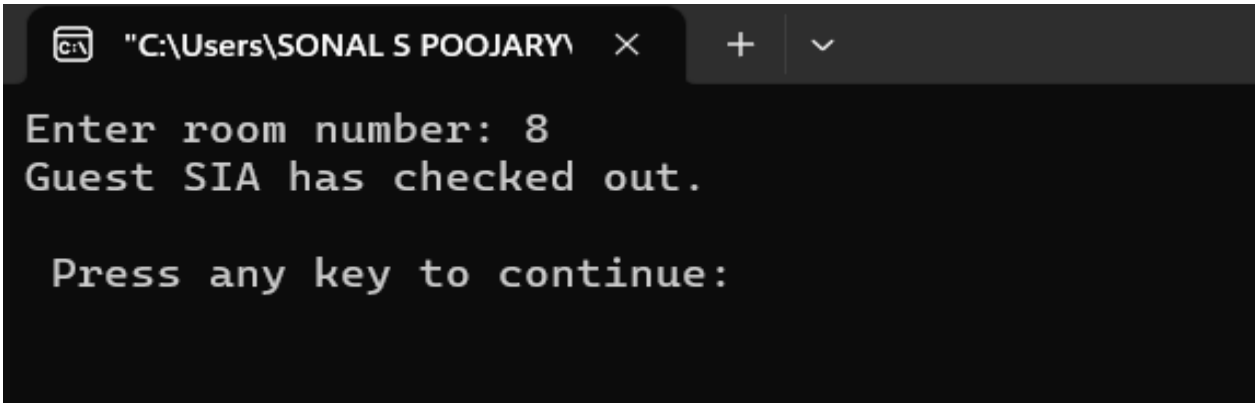




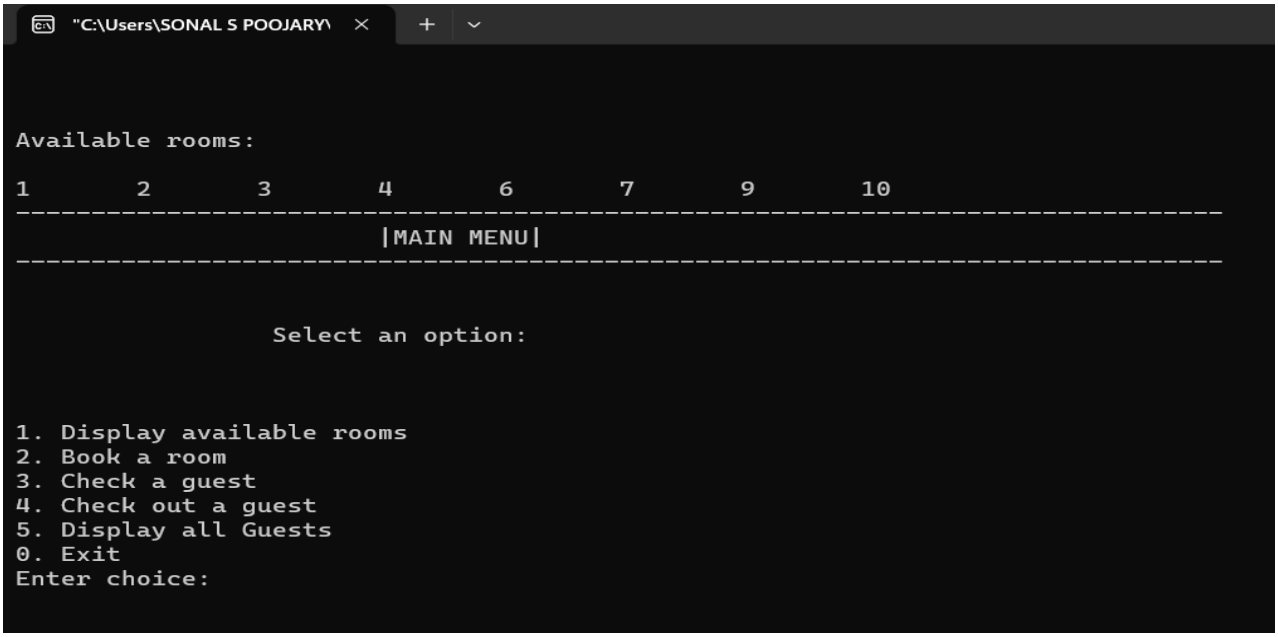
4. Display all the guest



5. Check-out a Guest



6. Available rooms



CONCLUSION

The program is designed to facilitate the management of room bookings, guest check-in, and check-out. The program employs a struct called Room to store relevant information about each room, including the room number, guest name, status, mobile number, nationality, and period of stay.

The use of a struct to store this information is a practical and efficient approach, as it allows for easy access to all the relevant data about each room. The inclusion of multiple functions within the program suggests that the program is intended to be comprehensive, addressing various aspects of room management.

Overall, the program appears to be a useful tool for managing a hotel or other type of lodging establishment. By centralizing information about room bookings, guest check-ins, and check-outs, the program can help streamline operations and improve overall efficiency.

LIMITATIONS AND FUTURE SCOPE

Limitations:

- Limited functionality: The program only manages simple tasks like reserving rooms and checking guests in and out. It lacks more sophisticated functions like staff scheduling, inventory management, and payment processing.

- Limited scalability: The program might have trouble expanding to handle a lot of rooms, visitors, and employees. It might get harder to run the system efficiently as the number of visitors and employees rises.

- Limited security: The program might not be secure enough to prevent unauthorized access to private information about guests and personnel. The privacy and security of visitors and staff members could be jeopardized by hacking or data breaches.

Future Aims:

- Payment processing: By integrating payment processing capabilities into the program, users will be able to pay for their accommodations and other services right from the interface.

- Inventory management elements, such as tracking supplies and overseeing housekeeping activities, can be added to the program.

- Staff scheduling: By enabling managers to set and manage staff schedules, we can guarantee that there is always enough staff coverage.

- Mobile App: The program will soon have a mobile application that will let users make reservations, view them, and manage their stay from their cellphones.

- Reporting and Analytics: The program's addition of reporting and analytics features will allow managers to create reports and examine information on revenue, occupancy rates, and other important indicators. By doing so, they will be able to make data-driven decisions and raise the hotel's general performance.

REFERENCES

https://www.lovelycoding.org/hotel-management-systemproject-in-c/

https://code-projects.org/hotel-management-system-c-programming-source-code/

https://itsourcecode.com/free-projects/c-projects/hotel-management-system-project-in-c-with-source-code/