



Hotel Management System

PAVAN KISHORE N [PES2UG22CS385]

PRAVEEN KUMAR M [PES2UG22CS413]

P PUNEETH S S [PES2UG22CS380]



AGENDA

- ▶ Adding Customer and Room Details.
- ▶ Performing the functions of Check in and Check out.
- ▶ Implementation of cancellation of Room bookings.
- ▶ Implementations of Bill calculations.
- ▶ Handling the errors if the user enter the wrong details.
- ▶ Looping system.

SYNOPSIS

- ▶ This code is an implementation of Hotel Management System in C language using Data structures. It allows the customer to enter the his/her details in order to book the rooms. The code defines many functions such as to add rooms, to cancel bookings, to checkout, to display the rooms status, and to display the Bills. This code handles the errors if the customer enters the incorrect details, like invalid date entering, and cancelling the rooms after checking out etc. And this code also tracks the total earnings from the customers.



Technical Aspects

Programming Language : C

Data Structures: Structures and Linked Lists.

Abstract Data Types definitions

- ▶ **Structures** are used in the code to organize and encapsulate data related to rooms and customers, enhancing readability and maintainability. They provide a clear and modular way to manage information, improving code structure for hotel room reservations and customer details.
- ▶ **Linked lists** are used in the hotel management code for dynamic and efficient storage of room and customer details. They allow flexible allocation and deallocation of memory, facilitating seamless updates and modifications in response to changing occupancy and reservation status in the hotel.

project > C n.c > isDateValid(const char *)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <stdbool.h>
5  #include <time.h>
6  #include <conio.h>
7  //structure containing customer details
8  typedef struct customerdetails {
9      int roomNo;           // allocating room in ascending order
10     char name[50];
11     char address[100];
12     char checkInDate[20];
13     char checkOutDate[20];
14     float bill;
15     float originalBill;
16     bool isCancelled;
17     bool isCheckedout;
18     struct customerdetails* next;
19 } customerdetails;
20 // structure containing the room properties
21 typedef struct roomdetails {
22     int roomNo;
23     int status;
24     int customerCount;
25     struct roomdetails* next;
26 } roomdetails;
27
28 roomdetails* reserveroom = NULL;
29 customerdetails* customerentry = NULL;
30
31 // Function declarations
32 void greetings();
```

```
31 // Function declarations
32 void greetings();
33 roomdetails* addroom(roomdetails* start, int roomNo);
34 customerdetails* addcustomer(customerdetails* start, int roomNo, char* name, char* address, char* checkInDate, char* checkOutDate);
35 void displaycustomer(customerdetails* start);
36 void checkoutcustomer(int roomNo);
37 int isempty();
38 float calculatebill(char* checkInDate, char* checkOutDate, bool isCancelled);
39 void displayroomstatus(roomdetails* startR);
40 void allocateinitialrooms(int totalRooms);
41 void cancelbooking(int roomNo);
42 void displaytotalearnings();
43 bool isDateValid(const char* date);
44
45 int main() {
46     int choice, roomNo;
47     char name[50], address[100], checkInDate[20], checkOutDate[20];
48     int cancelRoomNo;
49
50     customerdetails* temp_cus;
51     greetings();
52
53     int totalRooms = 20;
54     allocateinitialrooms(totalRooms); // calling a function to initialize 20 rooms
55
56     do {
57         printf("\n");
58
59         printf("\n\n");
60         printf("1. Check in customer\n");
61         printf("2. Check out customer\n");
```

```
62 printf("3. Cancel booking\n");
63 printf("4. Display customer details\n");
64 printf("5. Display room status\n");
65 printf("6. Total revenue details\n");
66 printf("7. Exit\n");
67 printf("Enter your choice: ");
68 scanf("%d", &choice);
69
70 switch (choice) {
71     case 1:
72         printf("Enter customer details:\n");
73         printf("Name: ");
74         scanf("%[^\\n]", name);
75         printf("Address: ");
76         scanf("%[^\\n]", address);
77
78         do {
79             printf("Check-in date (yyyy-mm-dd): ");
80             scanf("%s", checkInDate);
81         } while (!isDateValid(checkInDate));
82         while (1) {
83             printf("Check-out date (yyyy-mm-dd): ");
84             scanf("%s", checkOutDate);
85             if (!isDateValid(checkOutDate)) {
86                 printf("Invalid date format. Please use the yyyy-mm-dd format.\n");
87             } else if (strcmp(checkOutDate, checkInDate) <= 0) {
88                 printf("Invalid check-out date. Please enter a valid date later than the check-in date.\n");
89             } else {
90                 break;
91             }
92         }
```



```
93
94
95     int roomAllocated = 0;
96
97     roomdetails* temp_room = reserveroom;
98     while (temp_room != NULL) {
99         if (temp_room->status == 0) {
100             customerentry = addcustomer(customerentry, temp_room->roomNo, name, address, checkInDate, checkOutDate);
101             temp_room->status = 1;
102             temp_room->customerCount++;
103             customerentry->bill = calculatebill(checkInDate, checkOutDate, customerentry->isCancelled);
104             printf("Customer allocated to Room %d.\n", temp_room->roomNo);
105             roomAllocated = 1;
106             break;
107         }
108         temp_room = temp_room->next;
109     }
110     if (!roomAllocated) {
111         printf("No rooms available. Please check out customers to make rooms available.\n");
112     }
113     break;
114 case 2:
115     printf("Enter room number: ");
116     scanf("%d", &roomNo);
117     checkoutcustomer(roomNo);
118     break;
119 case 3:
120     printf("Enter room number to cancel booking: ");
121     scanf("%d", &cancelRoomNo);
122     cancelbooking(cancelRoomNo);
123     break;
```

```

124         case 4:
125             displaycustomer(customerentry);
126             break;
127         case 5:
128             displayroomstatus(reserveroom);
129             break;
130         case 6:
131             displaytotalearnings();
132             break;
133         case 7:
134             printf("Exiting...\n");
135             break;
136         default:
137             printf("Invalid choice. Please choose again.\n");
138     }
139 } while (choice != 7);
140
141 return 0;
142 }
143
144 void greetings()
145 {
146     printf("\n\n");
147     printf("\t\t\t\t *****\n");
148     printf("\t\t\t\t * * * * *\n");
149     printf("\t\t\t\t * * * * *\n");
150     printf("\t\t\t\t * ----- * \n");
151     printf("\t\t\t\t *  WELCOME TO PESU LODGE  * \n");
152     printf("\t\t\t\t * ----- * \n");
153     printf("\t\t\t\t * * * * *\n");
154     printf("\t\t\t\t * PES UNIVERSITY,BENGALURU,India * \n");

```

```

155     printf("\t\t\t\t\t *      Email: pesulodgePPP@gmail.com      *\n");
156     printf("\t\t\t\t\t *      Contact:98364723461      *\n");
157     printf("\t\t\t\t\t *      *\n");
158     printf("\t\t\t\t\t *****\n");
159     printf("\t\t\t\t\t Press any key to continue: ");
160     getch();
161 }
162
163 void allocateinitialrooms(int totalRooms) {
164     for (int roomNo = 1; roomNo <= totalRooms; roomNo++) {
165         reserveroom = addroom(reserveroom, roomNo);
166     }
167 }
168
169 roomdetails* addroom(roomdetails* start, int roomNo) {
170     roomdetails* new_room = (roomdetails*)malloc(sizeof(roomdetails));
171     new_room->roomNo = roomNo;
172     new_room->status = 0;
173     new_room->customerCount = 0;
174     new_room->next = NULL;
175     if (start == NULL) {
176         return new_room;
177     }
178     roomdetails* temp = start;
179     while (temp->next != NULL) {
180         temp = temp->next;
181     }
182     temp->next = new_room;
183     return start;
184 }
185

```



```
217         temp_room->customerCount--;
218         if (temp_room->customerCount == 0) {
219             temp_room->status = 0;
220         }
221         break;
222     }
223     temp_room = temp_room->next;
224 }
225 temp_cus->isCheckedOut = true;
226 printf("Customer in Room %d has been checked out.\n", roomNo);
227 return;
228 }
229 else {
230     printf("Room %d has already been cancelled. Cannot check out.\n", roomNo);
231 }
232 break;
233 }
234 temp_cus = temp_cus->next;
235 }
236 if (!roomFound) {
237     printf("Room %d is not allocated.\n", roomNo);
238 }
239 }
240
241 int isempty() {
242     roomdetails* temp_room = reserveroom;
243     while (temp_room != NULL) {
244         if (temp_room->status == 0) {
245             return 0;
246         }
247         temp_room = temp_room->next;
```



```
}  
return 1;  
}  
  
void cancelbooking(int roomNo) {  
    customerdetails* temp_cus = customerentry;  
    int roomFound = 0;  
  
    while (temp_cus != NULL) {  
        if (temp_cus->roomNo == roomNo) {  
            roomFound = 1;  
  
            if (temp_cus->isCheckedOut) {  
                printf("Room %d is currently empty. Cannot cancel booking.\n", roomNo);  
                return;  
            }  
  
            if (temp_cus->isCancelled) {  
                printf("Booking for Room %d has already been cancelled.\n", roomNo);  
                return;  
            }  
  
            temp_cus->isCancelled = true;  
            temp_cus->bill += 200.0;  
            temp_cus->originalBill = temp_cus->bill;  
            roomdetails* temp_room = reserveroom;  
            while (temp_room != NULL) {  
                if (temp_room->roomNo == roomNo) {  
                    temp_room->status = 0;  
                    temp_room->customerCount = 0;  
                    break;  
                }  
            }  
        }  
    }  
}
```

```

279         temp_room = temp_room->next;
280     }
281     printf("Booking for Room %d has been cancelled. Fine of $200 applied.\n", roomNo);
282     return;
283 }
284 temp_cus = temp_cus->next;
285 }
286 printf("No booking found for Room %d.\n", roomNo);
287 }
288
289 void displaycustomer(customerdetails* start) {
290     customerdetails* temp_cus = start;
291     printf("\nCustomer Details:\n");
292     printf("Room No    Name                Address                Check-in Date    Check-out Date    Bill\n");
293     while (temp_cus != NULL) {
294         if (temp_cus->isCancelled) {
295             printf("%-9d%-30s%-30s%-15s%-16s200.00 (Fine)\n", temp_cus->roomNo, temp_cus->name, temp_cus->address, temp_cus->checkInDate,
296                 temp_cus->checkOutDate);
297         } else {
298             printf("%-9d%-30s%-30s%-15s%-16s%.2f\n", temp_cus->roomNo, temp_cus->name, temp_cus->address, temp_cus->checkInDate,
299                 temp_cus->checkOutDate, temp_cus->bill);
300         }
301         temp_cus = temp_cus->next;
302     }
303 }
304
305 float calculatebill(char* checkInDate, char* checkOutDate, bool isCancelled) {
306     struct tm tm1 = {0};
307     struct tm tm2 = {0};
308     if (sscanf(checkInDate, "%d-%d-%d", &tm1.tm_year, &tm1.tm_mon, &tm1.tm_mday) != 3 ||
309         sscanf(checkOutDate, "%d-%d-%d", &tm2.tm_year, &tm2.tm_mon, &tm2.tm_mday) != 3 ||

```

```

308     tm1.tm_year < 0 || tm2.tm_year < 0 || tm1.tm_mon < 1 || tm1.tm_mon > 12 || tm2.tm_mon < 1 || tm2.tm_mon > 12 ||
309     tm1.tm_mday < 1 || tm1.tm_mday > 31 || tm2.tm_mday < 1 || tm2.tm_mday > 31) {
310     printf("Invalid date format or out-of-range values. Please use the yyyy-mm-dd format and ensure valid date components.\n");
311     return -1.0;
312 }
313 tm1.tm_year -= 1900;
314 tm1.tm_mon -= 1;
315 tm2.tm_year -= 1900;
316 tm2.tm_mon -= 1;
317
318 time_t t1 = mktime(&tm1);
319 time_t t2 = mktime(&tm2);
320
321 if (t1 == -1 || t2 == -1 || t2 <= t1) {
322     printf("Invalid dates. Please make sure the check-out date is later than the check-in date.\n");
323     return -1.0;
324 }
325 double diff = difftime(t2, t1);
326 int totalDays = (int)(diff / (60 * 60 * 24));
327 return totalDays * 750.0;
328 }
329
330 void displayroomstatus(roomdetails* startR) {
331     int totalRooms = 0;
332     int occupiedRooms = 0;
333     int vacantRooms = 0;
334     roomdetails* temp_room = startR;
335     while (temp_room != NULL) {
336         totalRooms++;
337         if (temp_room->status == 1) {
338             occupiedRooms++;

```



```

339     } else {
340         vacantRooms++;
341     }
342     temp_room = temp_room->next;
343 }
344 printf("\nHotel Room Status Report:\n");
345 printf("Total Rooms: %d\n", totalRooms);
346 printf("Occupied Rooms: %d\n", occupiedRooms);
347 printf("Vacant Rooms: %d\n", vacantRooms);
348 printf("\nRoom Status Visualization:\n");
349 temp_room = startR;
350 while (temp_room != NULL) {
351     printf("Room %d: ", temp_room->roomNo);
352     if (temp_room->status == 1) {
353         printf("[Occupied] - ");
354         for (int i = 0; i < temp_room->customerCount; i++) {
355             printf("C%d ", i + 1);
356         }
357     } else {
358         printf("[Vacant]");
359     }
360     printf("\n");
361     temp_room = temp_room->next;
362 }
363 }
364
365 ~
366 void displaytotalearnings() {
367     customerdetails* temp_cus = customerentry;
368     float totalEarnings = 0.0;
369     while (temp_cus != NULL) {

```

```

370     if (!temp_cus->isCancelled) {
371         totalEarnings += temp_cus->bill;
372     } else {
373         totalEarnings += 200.0;
374     }
375     temp_cus = temp_cus->next;
376 }
377 printf("\nTotal Earnings: $%.2f\n", totalEarnings);
378 }
379
380 bool isDateValid(const char* date) {
381     struct tm tm = {0};
382     if (sscanf(date, "%d-%d-%d", &tm.tm_year, &tm.tm_mon, &tm.tm_mday) == 3) {
383         tm.tm_year -= 1900;
384         tm.tm_mon -= 1;
385         if ((tm.tm_year >= 0) && (tm.tm_mon >= 0) && (tm.tm_mon < 12) && (tm.tm_mday > 0)) {
386             int maxDaysInMonth = 31;
387             if (tm.tm_mon == 3 || tm.tm_mon == 5 || tm.tm_mon == 8 || tm.tm_mon == 10) {
388                 maxDaysInMonth = 30;
389             } else if (tm.tm_mon == 1) {
390                 maxDaysInMonth = ((tm.tm_year % 4 == 0 && tm.tm_year % 100 != 0) || (tm.tm_year % 400 == 0)) ? 29 : 28;
391             }
392             if (tm.tm_mday <= maxDaysInMonth) {
393                 time_t t = mktime(&tm);
394                 if (t != -1) {
395                     return true;
396                 }
397             }
398         }
399     }
400     printf("Invalid date format or out-of-range values. Please use the yyyy-mm-dd format and ensure valid date components.\n");

```



```
return false;
}
/*
1.the difference between bill and original bill is the original bill stores the cost of the room based on no of days.bill is used to store for
cancellation charges and then add it to original bill.
2.the isCancelled and isCheckedOut (stdbool) bool function is used to check for cancellation and checkout .if cancellation is true it will
display fine of 200 and add it to revenue details. if checked out is true then no cancellation charges can be applied if we enter by mistake .
3. reserverrom and customerentry are two pointers which is initialised to null to the structure it always point to first node of respective
structures.
*/
```

OUTPUT

```
*****
*                                     *
*                                     *
*      -----                      *
*      WELCOME TO PESU LODGE        *
*      -----                      *
*                                     *
*      PES UNIVERSITY,BENGALURU,India *
*      Email: pesulodgePPP@gmail.com  *
*      Contact:98364723461           *
*                                     *
*****
      Press any key to continue:
```

1. Check in customer
 2. Check out customer
 3. Cancel booking
 4. Display customer details
 5. Display room status
 6. Total revenue details
 7. Exit
- Enter your choice: █

Enter customer details:

Name: prakash

Address: banglore

Check-in date (yyyy-mm-dd): 2023-11-23

Check-out date (yyyy-mm-dd): 2023-11-29


Customer allocated to Room 1.

1. Check in customer
2. Check out customer
3. Cancel booking
4. Display customer details
5. Display room status
6. Total revenue details
7. Exit

Enter your choice: 4

Customer Details:


Room No	Name	Address	Check-in Date	Check-out Date	Bill
1	prakash	banglore	2023-11-23	2023-11-29	4500.00



```
1. Check in customer
2. Check out customer
3. Cancel booking
4. Display customer details
5. Display room status
6. Total revenue details
7. Exit
Enter your choice: 5
```

```
Hotel Room Status Report:
Total Rooms: 20
Occupied Rooms: 1
Vacant Rooms: 19
```

```
Room Status Visualization:
Room 1: [Occupied] - C1
Room 2: [Vacant]
Room 3: [Vacant]
Room 4: [Vacant]
Room 5: [Vacant]
Room 6: [Vacant]
Room 7: [Vacant]
Room 8: [Vacant]
Room 9: [Vacant]
Room 10: [Vacant]
Room 11: [Vacant]
Room 12: [Vacant]
Room 13: [Vacant]
Room 14: [Vacant]
Room 15: [Vacant]
Room 16: [Vacant]
Room 17: [Vacant]
```



```
1. Check in customer
2. Check out customer
3. Cancel booking
4. Display customer details
5. Display room status
6. Total revenue details
7. Exit
Enter your choice: 3
Enter room number to cancel booking: 1
Booking for Room 1 has been cancelled. Fine of $200 applied.
```

```
1. Check in customer
2. Check out customer
3. Cancel booking
4. Display customer details
5. Display room status
6. Total revenue details
7. Exit
Enter your choice: 2
Enter room number: 1
Customer in Room 1 has been checked out.
```



Thank You