

**Total Marks: 10 Obtained Marks:**

# Assignment 03 (Project)

**Last date of Submission: 28 May 2024**



**Submitted To: Shakeel Ahmad**



**Student Name: Saif-ur-Rehman Awan**

**Abdul Moiz**

**Humza Zaheer**

**Fahad Hussain**



**Reg. Number: 4778-FOC/BSSE/F23 (Saif-ur-Rehman Awan)**

**4819-FOC/BSSE/F23 (Abdul Moiz)**

**4803-FOC/BSSE/F23 (Humza Zaheer)**

**4783-FOC/BSSE/F23 (Fahad Hussain)**



**Hotel Reservation System:**

**Source Code:**

#include<iostream>

#include<string>

#include<cstdlib>

using namespace std;

float ef=1000,sf=3000,lf=5000; //intilizing room fares.

int c=0, //for counting of customers.

ern=100, //intilizing room numbers.

srn=200,

lrn=300;

struct BookRoom\_structure{ //for booking rooms, the following details will be saved.

string name,

checkin\_date,

Phone\_No,

category,

cnic\_No;

int stay\_days,

room\_no;

float totalfare;

};

struct Rooms\_structure{

int er,sr,lr; //categories of rooms.

};

Rooms\_structure R={5,5,5}; //Intial Number of each category room.

int n=R.er+R.sr+R.lr;

void clearscreen(){

// cout<<"\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n";

system("cls");

}

void pausescreen(){

system("pause");

}

void Hotelname(){

cout<<"\tWelcome to CR's Hotel.\n"

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

}

void BookaRoom\_module(int c,BookRoom\_structure B[],Rooms\_structure &R){ //module for Booking a Room.

Hotelname();

cout<<"\tBook A Room."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cin.ignore(); //for ignoring the previous input integer.

cout<<"Enter Your Details.\n"

<<"-------------------\n";

cout<<"Name: \t\t";getline(cin,B[c].name);

cout<<"CNIC: \t\t";cin>>B[c].cnic\_No;

cout<<"Phone No.: \t";cin>>B[c].Phone\_No;

cout<<"Check-in Date: \t";cin>>B[c].checkin\_date;

cout<<"Days to Stay: \t";cin>>B[c].stay\_days;

int roomcat;

do //for repeation incase of invalid input.

{

cout<<"Select Category: "<<endl

<<" 1. Economy.\n"

<<" 2. Standard.\n"

<<" 3. Luxury.\n"<<endl

<<"Choice: ";

cin>>roomcat; //taking input for room category.

switch (roomcat){

case 1:{

if(R.er<=0){ //checking whether Economy rooms are available or not.

cout<<"No Economy Rooms are available currently.";

continue;

}

B[c].totalfare=B[c].stay\_days\*ef;

R.er--; //decrement of available unoccupied rooms after booking.

B[c].category="Economy.";

B[c].room\_no=++ern; //assigning room no according to category selected.

break;

}

case 2:{

if(R.sr<=0){

cout<<"No Standard Rooms are available currently.";

continue;

}

B[c].totalfare=B[c].stay\_days\*sf; //calculating total flare according to category and no. of days to stay.

R.sr--;

B[c].category="Standard."; //assigning selected category.

B[c].room\_no=++srn;

break;

}

case 3:{

if(R.lr<=0){

cout<<"No Luxury Rooms are available currently.";

continue;

}

B[c].totalfare=B[c].stay\_days\*lf;

R.lr--;

B[c].category="Luxury.";

B[c].room\_no=++lrn;

break;

}

default:{ //incase of invalid option selected.

cout<<"Invalid Option Selected.\nPlease Select the Correct Category."<<endl;

break;

}

}

}while(roomcat!=1&&roomcat!=2&&roomcat!=3); //repeation of loop incase of invalid input.

cout<<endl<<B[c].category<<" Room Booked!" //showing Category selected, Room No. and Total Fare of the Room.

<<"\n Your Room No is: "<<B[c].room\_no

<<"\n Total Price of Room will be: "<<B[c].totalfare<<" Rs"<<endl<<endl;

pausescreen();

}

void Information(); //prototype of Information Section from the Main Menu.

void EditCustomerRecord\_module(BookRoom\_structure B[]); //Prototype for the editing of customer's record.

void EditRecords\_module(Rooms\_structure &R,BookRoom\_structure B[]) //module for going in Editing section from the Main Menu.

{

int edt;

do{ //loop for going back only one step instead of directly gong back to main menu.

clearscreen(); //for clearing screen.

Hotelname();

cout<<"\tEdit Records."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"1. Edit Customer's Record.\n"

<<"2. Edit Room's Fare.\n"

<<"3. Edit Room No.s\n\n"

<<"0. Go Back to Main Menu."<<endl;

do{ //for again taking input, incase of invalid input.

cout<<"\nEnter Your Choice: ";

cin>>edt;

switch(edt){

case 1:{

cout<<"\tEdit Customer's Record."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

EditCustomerRecord\_module(B); //for calling customer record edit module

break;

}

case 2:{ //for editing of Room Fares of each category.

int edtflr;

do{

clearscreen();

Hotelname();

cout<<"\tEdit Room's Fare."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"1. Economy. (Rs."<<ef<<" per Day)."<<endl

<<"2. Standard. (Rs."<<sf<<" per Day)."<<endl

<<"3. Luxary. (Rs."<<lf<<" per Day)."<<endl

<<"0. Go back to Edit Record.\n"

<<"\nEnter Your Choice: ";

do{ //for not going back without user decision.

cin>>edtflr;

if(edtflr==1||edtflr==2||edtflr==3){ //for the valid input process.

cout<<"Change Fare to: ";

}

switch(edtflr){

case 1:{

cin>>ef;

break;

}

case 2:{

cin>>sf;

break;

}

case 3:{

cin>>lf;

break;

}

case 0:{

continue;

break;

}

default:{

cout<<"\nPlease Select Correct Option.";

break;

}

}

}while(edtflr!=1&&edtflr!=2&&edtflr!=3&&edtflr!=0); //program remain in the edit room fare section

}while(edtflr!=0); //program exits only when 0 input is given, and goes to the Edit menu, not in main menu.

break;

}

case 3:{ //editing of Number of Rooms of each Category

int edtRno;

do{

clearscreen();

Hotelname();

cout<<"\tEdit Room No.s."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"1. Economy. ("<<R.er<<" Rooms left)."<<endl

<<"2. Standard. ("<<R.sr<<" Rooms Left)."<<endl

<<"3. Luxury. ("<<R.lr<<" Rooms Left)."<<endl

<<"0. Go back to Edit Record.\n"

<<"\nEnter Your Choice: ";

do{

cin>>edtRno;

if(edtRno==1||edtRno==2||edtRno==3){ //only gets executed if valid input is given

cout<<"Change Room No.s to: ";

}

switch(edtRno){

case 1:{

cin>>R.er; //taking new input for the number of Economy Rooms

break;

}

case 2:{

cin>>R.sr;

break;

}

case 3:{

cin>>R.lr;

break;

}

case 0:{

continue;

break;

}

default:{

cout<<"\nPlease Select Correct Option.";

break;

}

}

}while(edtRno!=1&&edtRno!=2&&edtRno!=3&&edtRno!=0); //program remain in the Edit Number of Room Menu.

}while(edtRno!=0); //program goes one step back only when 0 input is given.

break;

}

case 0:{

continue; //incase of 0 input, when user wants to go back to Edit Menu Section.

break;

}

default:{

cout<<"Please Select Correct Option."<<endl;

break;

}

}

}while(edt!=1&&edt!=2&&edt!=3&&edt!=0); //program goes into loop, incase of invalid input.

}while(edt!=0); //for keeping proram in the same section,until user wants to exit.

}

void CustomerEditRecord\_submodule(BookRoom\_structure B[],int i); //sub-module prototype just for the editing of Customer's Record.

void EditCustomerRecord\_module(BookRoom\_structure B[]){ //Module for searching of specific customer to edit it's record.

int edtcus,RoomNotoedit;

string nametoedit;

do{

clearscreen();

Hotelname();

cout<<"\tSearch (Edit Customer's Record)."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"Search Customer by: \n"

<<"1. Name.\n"

<<"2. Room No.\n\n"

<<"0. Go Back to Main Menu.\n\n";

do{ //incase of invalid input.

cout<<"Enter your Choice: ";

cin>>edtcus;

switch(edtcus){

case 1:{

cin.ignore();

bool found=false; //for remaining in the loop, incase of invalid customer's name is entered.

do{

cout<<"\nEnter Customer's name to Search or 0 to Go Back.\n"

<<"Enter Customer's Name: ";

getline(cin,nametoedit); //geting string input with whitespaces.

if(nametoedit=="0"){ //checking whether the user wants to go back (to search by name/room no menu) or not

continue;

}

for(int i=0;i<c;i++){ //for checking all the customers data

if(nametoedit==B[i].name){ //will execute only if specific name customer is found.

found=true; //for exicting the loop of searching customer by name

CustomerEditRecord\_submodule(B,i); //calling module of Editing Customer Records submodule

continue; //for exiting from while loop

}

}

if(!found){ //incase of invalid customer name is entered.

cout<<"\nNo such Name Customer found.\n"

<<"Please Try Again using Different Keywords.\n";

pausescreen();

}

}while(nametoedit!="0"&&!found);

break;

}

case 2:{ //for searching customer by room no to edit its record

bool found=false;

do{

cout<<"\nEnter Customer's Room No. to Search or 0 to Go Back.\n"

<<"Enter Customer's Room No.: ";

cin>>RoomNotoedit;

if(RoomNotoedit==0){

continue;

}

for(int i=0;i<c;i++){

if(RoomNotoedit==B[i].room\_no){

found=true;

CustomerEditRecord\_submodule(B,i);

continue;

}

}

if(!found){

cout<<"\nNo such Room No.'s Customer found.\n"

<<"Please Try Again using Different Room No.\n";

pausescreen();

}

}while(RoomNotoedit!=0&&!found);

break;

}

case 0:{

continue;

break;

}

default:{

cout<<"Please Select the Correct Option.\n";

break;

}

}

}while(edtcus!=1&&edtcus!=2&&edtcus!=0);

}while(edtcus!=0);

}

void CustomerEditRecord\_submodule(BookRoom\_structure B[],int i){ //submodule jsut for editing of selected customer records.

int Edtcussub;

do{

clearscreen();

Hotelname();

cout<<"\tEdit Customer's Record."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"Customer Details:\n"

<<"-----------------\n"

<<"Enter 1-6 to Edit Details."<<endl

<<endl

<<"1. Name: \t\t"<<B[i].name<<endl

<<"2. Phone No.: \t\t"<<B[i].Phone\_No<<endl

<<"3. CNIC No.: \t\t"<<B[i].cnic\_No<<endl

<<"4. Days to stay: \t"<<B[i].stay\_days<<endl

<<"5. Category: \t\t"<<B[i].category<<endl

<<"6. Check in Date: \t"<<B[i].checkin\_date<<endl<<endl

<<" Fare: \t"<<B[i].totalfare<<endl

<<" Room No.: \t"<<B[i].room\_no<<endl<<endl

<<"0. Back to Search (Edit Customer's Record)."<<endl;

do{ //goes into loop incase of invalid input

cout<<"\nEnter Your Choice: ";

cin>>Edtcussub;

cin.ignore();

switch(Edtcussub){

case 1:{

cout<<"\nChange Customer's Name to: ";

getline(cin,B[i].name);

break;

}

case 2:{

cout<<"\nChange Customer's Phone No. to : ";

getline(cin,B[i].Phone\_No);

break;

}

case 3:{

cout<<"\nChange CNIC No. to: ";

getline(cin,B[i].cnic\_No);

break;

}

case 4:{

cout<<"\nChange Stay days to: ";

cin>>B[i].stay\_days;

if(B[i].category=="Economy."){ //will check the previously selected category. if it was economy, then following actions will be taken.

B[i].totalfare=B[i].stay\_days\*ef;

B[i].room\_no=++ern;

}

else if(B[i].category=="Standard."){

B[i].totalfare=B[i].stay\_days\*sf; //for changing fare according to latest stay days

B[i].room\_no=++srn;

}

else if(B[i].category=="Luxury."){

B[i].totalfare=B[i].stay\_days\*lf;

B[i].room\_no=++lrn; //for assigning new room no for luxary category.

}

break;

}

case 5:{ //for changing category.

int catedt;

cout<<"\nChange Category to:\n"

<<" 1. Economy.\n"

<<" 2. Standard.\n"

<<" 3. Luxury.\n"

<<"Please Enter Your Choice: ";

cin>>catedt;

switch(catedt){

case 1:{

if(R.er<=0){ //checking weather rooms are available for selected category or not.

cout<<"No Economy Rooms are available currently.";

continue;

}else{

R.er--; //if economy is room is selected, it will decrease the No. of Economy Rooms available

if(B[i].category=="Standard."){

R.sr--;

}else if(B[i].category=="Luxury."){

R.lr--;

}

B[i].category="Economy.";

B[i].totalfare=B[i].stay\_days\*ef;

B[i].room\_no=++ern; //for assigning a new Economy Class Room No.

}

break;

}

case 2:{

if(R.sr<=0){

cout<<"No Standard Rooms are available currently.";

continue;

}else{

R.sr--;

if(B[i].category=="Economy."){

R.er--;

}else if(B[i].category=="Luxury."){

R.lr--;

}

B[i].category="Standard.";

B[i].totalfare=B[i].stay\_days\*sf; //chaning fare according to category selected

B[i].room\_no=++srn;

}

break;

}

case 3:{

if(R.lr<=0){

cout<<"No Luxury Rooms are available currently.";

continue;

}else{

R.lr--;

if(B[i].category=="Economy."){

R.er--;

}else if(B[i].category=="Standard."){

R.sr--;

}

B[i].category="Luxury.";

B[i].totalfare=B[i].stay\_days\*lf;

B[i].room\_no=++lrn;

}

break;

}

default:{

cout<<"Invalid Option Selected.\n"

<<"Please Re-select the Correct Option.";

break;

}

}

break;

}

case 6:{

cout<<"\nChange Check-in Date to: ";

getline(cin,B[i].checkin\_date);

break;

}

case 0:{

continue;

break;

}

default:{

cout<<"\nInvalid Option Selected.\n"

<<"Please Re-select the correct option.\n";

break;

}

}

}while(Edtcussub!=1&&Edtcussub!=2&&Edtcussub!=3&&Edtcussub!=4&&Edtcussub!=5&&Edtcussub!=6&&Edtcussub!=0);

}while(Edtcussub!=0);

}

void checkoutofcustomer\_module(BookRoom\_structure B[],Rooms\_structure R){ //module for checking out of customer.

bool blckot=false, //for not going back to main menu, without user intension from search bar of check out menu.

blcusot2=false; //for remaining inside loop, incase of invalid input is given at the time of paymentnof checkout.

string chkot;

int chkoutint,paymentmethodchoose;

do{

clearscreen();

Hotelname();

cout<<"\tCheck-out."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"Enter Name of Customer to Check-out.\n"

<<"Enter 0 to Go back to Main Menu.\n"

<<"\nEnter: ";

getline(cin,chkot);

if(chkot=="0"){

blckot=true;

continue;

}

for(int i=0;i<c;i++){ //for searching f specific customer.

if(chkot==B[i].name){

blcusot2=true;

cout<<"Customer's Detail.\n"

<<"----------------------------\n"

<<"Name: \t\t"<<B[i].name<<endl

<<"Room No.: \t"<<B[i].room\_no<<endl<<endl

<<"Phone No.: \t"<<B[i].Phone\_No<<endl

<<"CNIC No.: \t"<<B[i].cnic\_No<<endl

<<"Days to stay: \t"<<B[i].stay\_days<<endl

<<"Category: \t"<<B[i].category<<endl

<<"Check in Date: \t"<<B[i].checkin\_date<<endl

<<"Total Fare: \t"<<B[i].totalfare<<endl

<<"---------------------------------\n\n";

do{ //for remaining inside loop incase of wrong input is given.

cout<<"Does Customer want to check-out.\n"

<<" 1) Check-out.\n"

<<" 0) Back to Search Customer(Check-out).\n\n"

<<"Enter Your Choice: ";

cin>>chkoutint;

switch(chkoutint){

case 1:{

do{ //incase user has decided to check-out, payment methods are now being shown to user.

cout<<"\nEnter Your Payment Method.\n"

<<" 1) Cash.\n"

<<" 2) Credit Card.\n"

<<" 3) Debit Card.\n"

<<"Enter Your Choice: ";

cin>>paymentmethodchoose;

if(paymentmethodchoose==1||paymentmethodchoose==2||paymentmethodchoose==3){ //will only gets executed if valid payment method is selected.

cout<<"\nPayment Accepted.\n"

<<"Customer has been Checked Out.\n";

B[i].name.clear(); //deleting customers records permanently.

B[i].room\_no=0;

B[i].checkin\_date.clear();

B[i].Phone\_No.clear();

B[i].cnic\_No.clear();

B[i].stay\_days=0;

if(B[i].category=="Economy."){

R.er++; //freeing up the book room by increasing the no of unoccupied/available rooms

}else if(B[i].category=="Standard."){

R.sr++;

}else if(B[i].category=="Luxury."){

R.lr++;

}

B[i].category.clear();

B[i].totalfare=0;

cout<<"All Data of Customer have been Deleted.\n"

<<"Going back to Main Menu.\n\n";

pausescreen();

blckot=true; //for going back to main menu

}

else{ //incase of invlaid payment method is selected.

cout<<"\nInvalid Payment Method Selected.\n"

<<"Please Select a Valid Payment Method.\n";

blckot=false;

}

}while(paymentmethodchoose!=1&&paymentmethodchoose!=2&&paymentmethodchoose!=3&&!blckot); //for making user to choose correct payment method.

break;

}

case 0:{ //incase user wants to search for another person to make him checkout, not the person he search for.

blckot=false;

// continue;

break;

}

default:{ //incase user has not selected either to checkout or go back to search bar of check out menu.

cout<<"\nInvalid Option Selected.\n"

<<"Please Select a valid Option.\n\n";

pausescreen();

break;

}

}

}while(chkoutint!=1&&chkoutint!=0); //keep user in loop if he hasn't choose to check out or go back to search menu forcing user to choose valid option.

}

}

if(!blcusot2){ //incase user input named customer is not found in records stored.

cout<<"\nNo such Customer Found.\n"

<<"Please Try Again.\n\n";

pausescreen();

}

}while(!blckot&&chkot!="0"); //if user wants to go back to main menu with intension.

}

void Credit(); //prototype for credit section from the main menu.

void Intructions(); //prototype for Information section from the man menu.

int main()

{

BookRoom\_structure B[n]; //making shortcut of roombooking structure and adding limit to no of rooms that can be booked accoring to Rooms available in each category.

for(;;)

{

Hotelname();

cout<<"\tMain Menu."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

int x;

cout<<"1. Book a room."<<endl

<<"2. Customer Search."<<endl

<<"3. Room Details.\n"

<<"4. Edit Record.\n"

<<"5. Information.\n"

<<"6. Check-out.\n"

<<"7. Credit.\n"

<<"8. Instructions.\n"

<<"0. Exit."<<endl

<<"\nPlease Enter Your Choice: ";

cin>>x;

switch(x){

case 1:{

clearscreen();

BookaRoom\_module(c,B,R); //calling of Booking A Room module.

c++; //increasing of no. of customers that have booked room

break;

}

case 2:{ //for search of specific customer by any detail available.

cin.ignore();

string srchcus;

bool bfndsrch=false;

do{

system("cls");

Hotelname();

cout<<"\nCustomer Search."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"Enter - Name."<<endl

<<" - CNIC."<<endl

<<" - Phone No."<<endl

<<"Enter any of above to Search Customer."<<endl<<endl

<<"Or 0 to Go Back to Main Menu."<<endl<<endl

<<"Enter: ";

getline(cin,srchcus);

if(srchcus=="0"){

continue;

}

cout<<"\nCustomer's Details."<<endl

<<"------------------------\n";

for(int i=0;i<c;i++){ //searching for specific detailed customer.

if(srchcus==B[i].name||srchcus==B[i].cnic\_No||srchcus==B[i].Phone\_No){

cout<<"Name: \t\t"<<B[i].name<<endl

<<"Room No.: \t"<<B[i].room\_no<<endl<<endl

<<"Phone No.: \t"<<B[i].Phone\_No<<endl

<<"CNIC No.: \t"<<B[i].cnic\_No<<endl

<<"Days to stay: \t"<<B[i].stay\_days<<endl

<<"Category: \t"<<B[i].category<<endl

<<"Check in Date: \t"<<B[i].checkin\_date<<endl

<<"---------------------------------\n\n";

pausescreen();

bfndsrch=true;

break;

}

}

if(!bfndsrch){ //incase no such avaiable detailed customer is found.

cout<<"No such Detail's Customer Found.\n"

<<"Please try Again using Different Details.\n"

<<"------------------------------------------\n";

pausescreen();

}

}while(srchcus!="0"&&!bfndsrch); //can only exit this section if user wants.

break;

}

case 3:{ //for showing no =, of available rooms and all the customers details.

clearscreen();

Hotelname();

cout<<"\tRooms Available."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"Economy: "<<R.er<<endl

<<"Standard: "<<R.sr<<endl

<<"Luxury: "<<R.lr<<endl

<<endl;

cout<<"\tCustomer's List."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

if(c==0){ //incase no customer has booked a room till now.

cout<<"No Customers record are available currently."<<endl

<<"Please Enter record to view.\n"<<endl;

}

for(int i=0;i<c;i++){

if(B[i].room\_no==0){ //incase all the customers have been checked out.

cout<<"No Customers record are available currently."<<endl

<<"Please Enter record to view.\n"<<endl;

}

}

for(int i=0;i<c;i++){

if(B[i].room\_no>0){ //showing details of customers whose rooms no are greater than 0, that is the custoemr hasn't been checkeed out yet and is available in the hotel.

cout<<"Room No: \t"<<B[i].room\_no<<endl

<<"Customer Name: \t"<<B[i].name<<endl

<<"Phone No.: \t"<<B[i].Phone\_No<<endl

<<"CNIC No.: \t"<<B[i].cnic\_No<<endl

<<"Days to stay: \t"<<B[i].stay\_days<<endl

<<"Category: \t"<<B[i].category<<endl

<<"Check in Date: \t"<<B[i].checkin\_date<<endl

<<"Fare: \t\t"<<B[i].totalfare<<" Rs."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

}

pausescreen();

break;

}

case 4:{

EditRecords\_module(R,B); //for calling of Edit Records module.

break;

}

case 5:{

clearscreen();

Information(); //for calling of Information Section Module.

break;

}

case 6:{

cin.ignore();

checkoutofcustomer\_module(B,R); //for calling Customer Checkout Module

break;

}

case 7:{

Credit(); //calling of Credit section module

pausescreen();

break;

}

case 8:{

Intructions(); //caling of Instruction section module.

pausescreen();

break;

}

case 0:{

return 0;

break;

}

default:{ //incase user has selected wrong option

cout<<"Invalid Option Selected."<<endl

<<"Please Enter Valid Option."<<endl;

break;

}

}

clearscreen();

}

}

void Information(){ //module of Information Section.

Hotelname();

cout<<"\tInformation Section."<<endl

<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"

<<"\tGeneral.\n"

<<"--------------------\n"

<<endl

<<"\tEconomy.\n"

<<"--------------------\n"

<<" \* Cozy and Compact Rooms."

<<"\n \* Essential amenities."

<<"\n \* Comfortable Bed and Pillows."

<<"\n \* Flat-screen TV with limited channels."

<<"\n \* Free Wi-Fi."

<<"\n \* Private Bathroom with shower."

<<"\n \* Hairdryer and toiletries."

<<"\n \* Ideal for budget-conscious travellers and short stays."

<<endl

<<"\n\tStandard.\n"

<<"--------------------\n"

<<"\n \* Spacious Rooms."

<<"\n \* Additional Amenities."

<<"\n \* Work Desk and Chair."

<<"\n \* Mini-fridge and Coffee maker."

<<"\n \* In-room Safe."

<<"\n \* Iron and Ironing board."

<<"\n \* Upgraded Bathroom Amenities."

<<"\n \* Access to Fitness Center and Business Center."

<<"\n \* Suitable for Business Travelers and Families."

<<endl

<<"\n\tLuxury.\n"

<<"--------------------\n"

<<"\n \* Lavish Room."

<<"\n \* Premimum Amenities."

<<"\n \* King size Bed and Plush lines."

<<"\n \* Flat-screen TV with premium channels."

<<"\n \* Comlimentary bottled water and snacks."

<<"\n \* Spacious Bathroom with separate shower and Bathtub."

<<"\n \* High-end Toiletries and robes."

<<"\n \* Personalized Room Service and Concierge."

<<"\n \* Access to Executive Lounge and VIP services."

<<"\n \* Ideal for Special Occasions, Honeymooners, and Discering Travelers."

<<endl<<endl;

pausescreen();

}

void Credit(){ //module for credit section.

clearscreen();

Hotelname();

cout<<"\tCredit Section.\n"

<<"--------------------------------\n\n"

<<"This Hotel Reservation System is Developed by:\n"

<<"\t1) Saif-ur-Rehman Awan (Senior Developer and Team Leader).\n"

<<"\t2) Abdul Moiz. (Teammate).\n"

<<"\t3) Humza Zaheer. (Co-Leader).\n"

<<"\t4) Fahad Hussain. (Teammate).\n\n"

<<"Special Thanks to:\n"

<<" 1. Mr. Shakeel Ahmed (PF Professor).\n"

<<" 2. International Islamic University, Islamabad (IIUI).\n\n"

<<" This Project is a part of Programming Fundamentals (PF) Assignment no 3.\n"

<<"Under the supervision of Mr. Shakeel Ahmed, Department of SE, IIUI.\n"

<<"\nThank you for using our Hotel Reservation System!\n"<<endl;

}

void Intructions(){ //module for Intruction Section

clearscreen();

Hotelname();

cout<<"\tInstructions Section.\n"

<<"--------------------------------\n\n"

<<"The following are some of the instructions a user should follow in order,\n"

<<"in order to run and use program effectiviely.\n\n"

<<"1) You are free to use any Name, CNIC no, Phone No and Check-in Date,\n"

<<" in any format, you can also include white-spaces in between them.\n"

<<"2) Don't use other Characters than Numeric when entering Stay Days,\n"

<<" otherwise, the program will start behiouving unusually.\n"

<<"3) You are Adviced to use only Options that are showing on the display,\n"

<<" Program will not accept any other input.\n"

<<"4) When searching, use exact words to gain expected output,\n"

<<" using non-exact words will lead to failure in search.\n"

<<" It is also advised that before searching or checking-out a customer,\n"

<<" See the exact keywords of Name before, from the Rooms Details Section.\n"

<<endl

<<"Here are some features you can use in program:\n\n"

<<"The user can:\n"

<<"\* Book Multiple Rooms at once, and save of their data.\n"

<<"\* See how many Un-Occupied Rooms are available in your Hotel.\n"

<<"\* See the Details of All the Occupied Room's Customer.\n"

<<"\* Modify the Customer's Details.\n"

<<"\* Change the fare rate of each category.\n"

<<"\* Increase or Decrease the No. of Room of each category.\n"

<<"\* Search for one specific Customer.\n"

<<"\* Make the customer check-out, at any time.\n"

<<"\* See the Information about features of each category's room.\n"

<<"\* See the Credits (Developing Team and Core Purpose.).\n"

<<"\* Exit the program at any time, with its own free will.\n\n"

<<"Written by Saif-ur-Rehman Awan.\n"

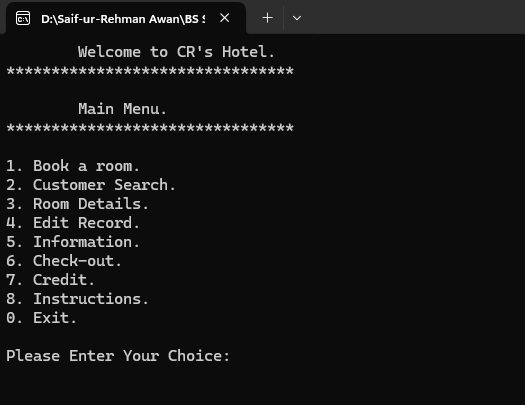
<<endl;

}

**Outputs:**

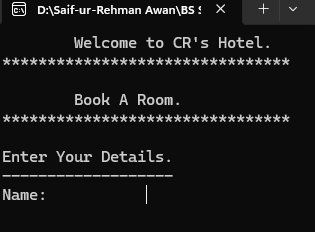
**Main Menu:**

This is the screen-shot of our Main Menu, from where the program starts.

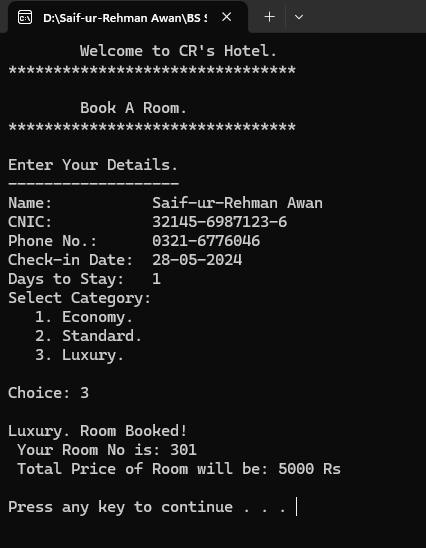
****

**(1) Book A Room (Section):**

This section opens by pressing “1” from Main Menu, in which a customer’s details are entered into the system, to be saved.

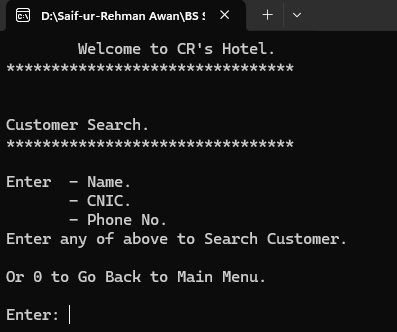


After Entering Data.

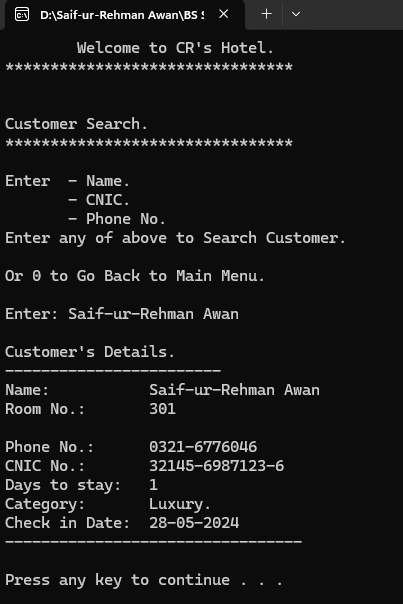


**(2) Customer Search (Section):**

This section opens by pressing “2” from the Main Menu, inside which a user can search for specific customer by entering any of its details.

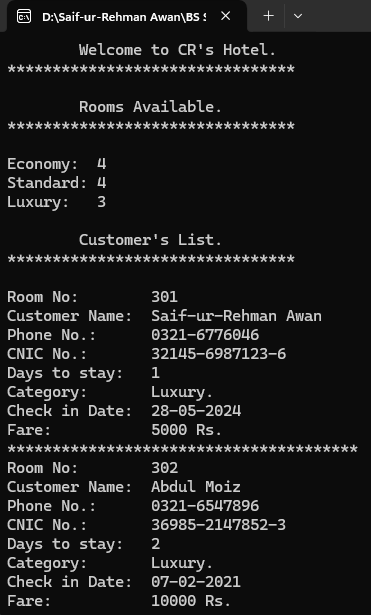


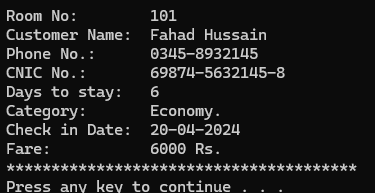
After Entering Customer’s name.



**(3) Room Details (Section):**

This section opens by pressing “3” from the Main Menu, and displays you the information about all the unoccupied rooms and the Customer’s Details, who have checked-in.



****

**(4) Edit Records (Section):**

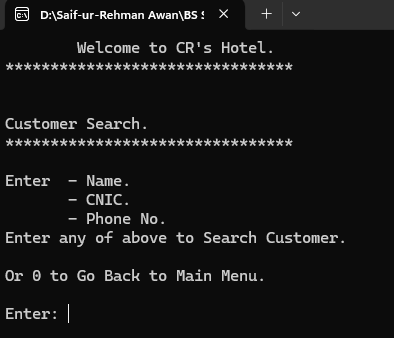
This section contains all the Editing Settings that can be done inside from the program, including editing of Customer’s record, changing amount of fare of each category, and increasing or decreasing of no. of rooms of each category.

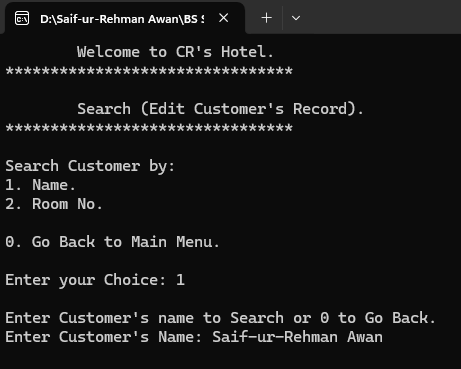


**(4.1) Edit Customer’s Records (Sub-Section of “Edit Records”):**

This sub-menu is open from inside of “Edit Records” section by pressing “1”.

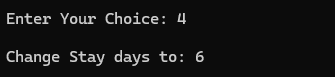
Inside this section, are the settings available to edit any of the customer’s record according to user’s will. The user will first have to search for that Customer by its name or room no. and then.







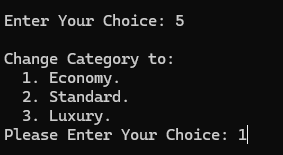
By Selecting “4”, for changing Stay Days.



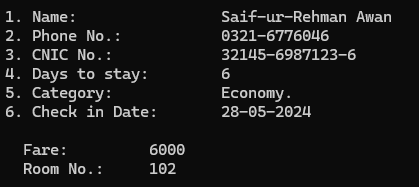
Room No. and Fare will be automatically changed according to Stay Days.



Now, by selecting “5”, to change the category, the Fare and Room No. will again change according to Category Selected.



Result:



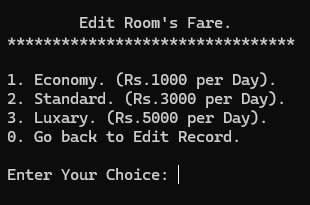
By pressing “0”, the user will be directed one step back (Search bar of Customer’s Edit Record).

By again pressing “0”, the user is prompted back to “Edit Record” menu.

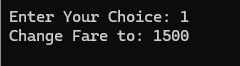
**(4.2) Edit Room’s Fare (Sub-Section of “Edit Records”):**

This section open by entering “2” inside Edit Records Menu, inside which fares of different categories are listed and user is asked whether he/she wants to change the Room’s fare or not.

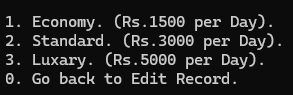
If he wants to change the fare, then he/she first have to select the category and have to enter new fare for that category.



After selecting category and new fare.



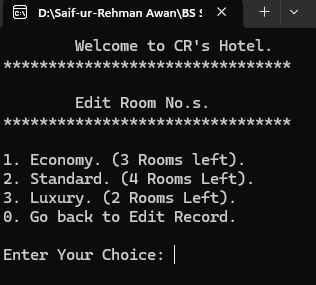
Result:



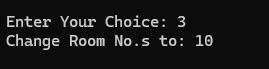
By pressing “0”, the user will be directed one step back (Search bar of Customer’s Edit Record).

**(4.3) Edit Room’s Number (Sub-Section of “Edit Records”):**

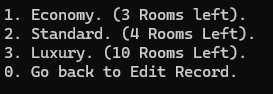
This sub-section is open by pressing “3” in the Edit Records section, and it will prompt user to the “Edit Room No’s” sub-section, where he/she can change the total numbers of rooms of each category.



After Selecting category, and number of rooms user want.



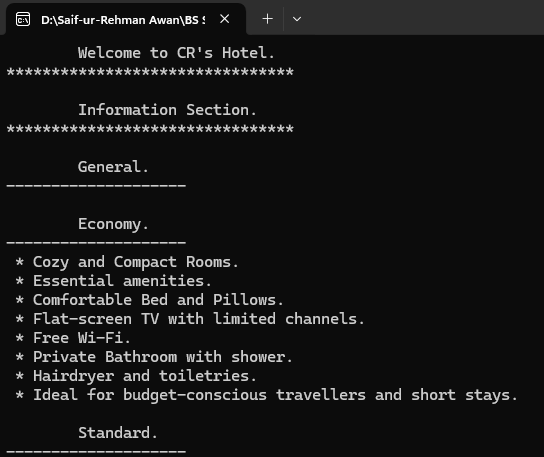
Result:

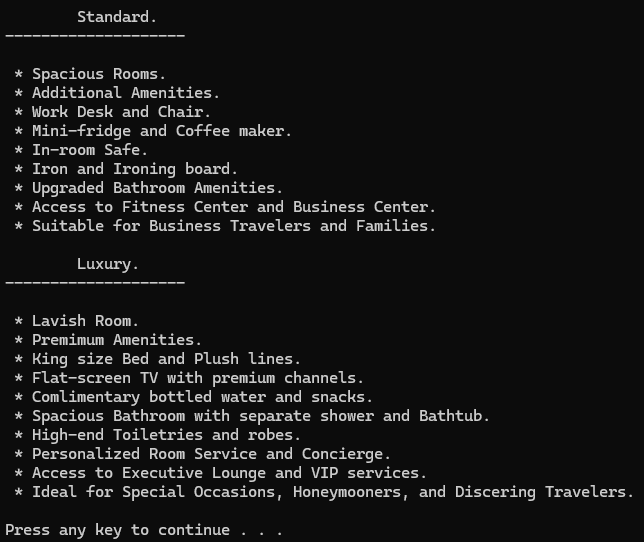


Now, if user wants to go back, he/she just simply has to press “0”, to get back to “Edit Record” menu, from where he can go back to “Main Menu” just by pressing “0” again.

**(5) Information (Section):**

This section opens by entering “5” from the Main Menu, in which information is displayed about the general features and facilities that will be provided to each customer according to their category.

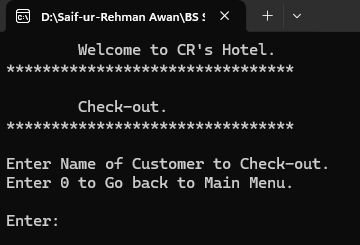




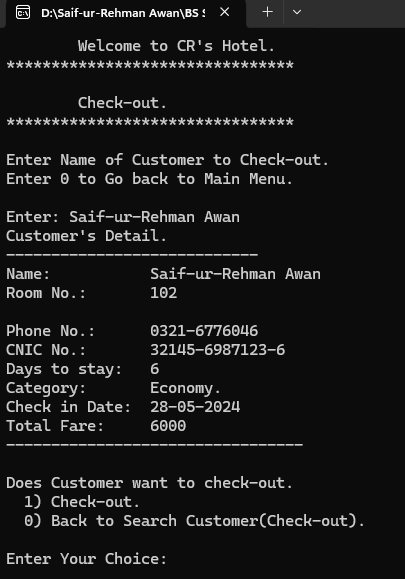
Now, press any key to go back to “Main Menu”.

**(6) Check-out (Section):**

This section contains all the checkout options and can be open from the Main Menu by entering “6”. Inside this section, user is first asked to enter the “Name of Customer” who wants to check-out, or “0” to go back to Main Menu.

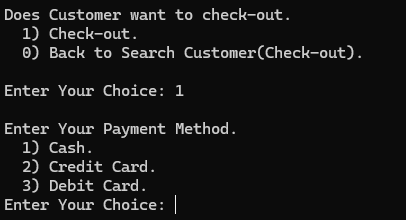


Upon entering the name of the customer, the entire detail of the customer is shown in the screen, giving user option to verify user and then taking the decision, whether he/she wants to check-out this customer or not?

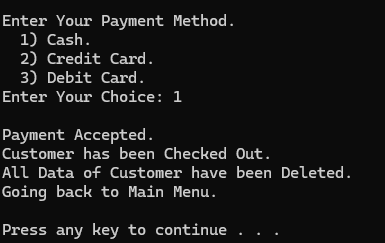


If not, he/she, can simply press the “0” to go back to the “Search bar of Check-out Menu”.

If user presses “1”, multiple payment methods will appear giving customer various payment options to pay its Bill.



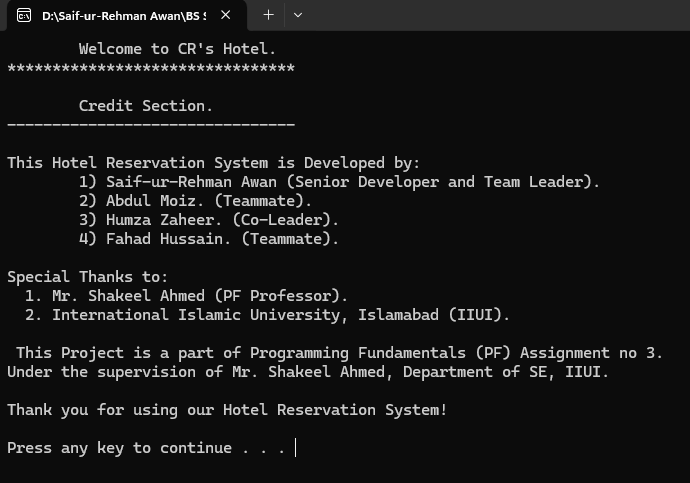
Upon entering the valid payment method option, The Customer is Checked-out from the Hotel, deleting it’s entire data from the Hotel’s Database.



After-it, press any key to go back to “Main Menu”.

**(7) Credit (Section):**

This section open from the Main Menu by entering “7”, in which the information about the developing team, project supervisor, and core purpose of developing this program is given.

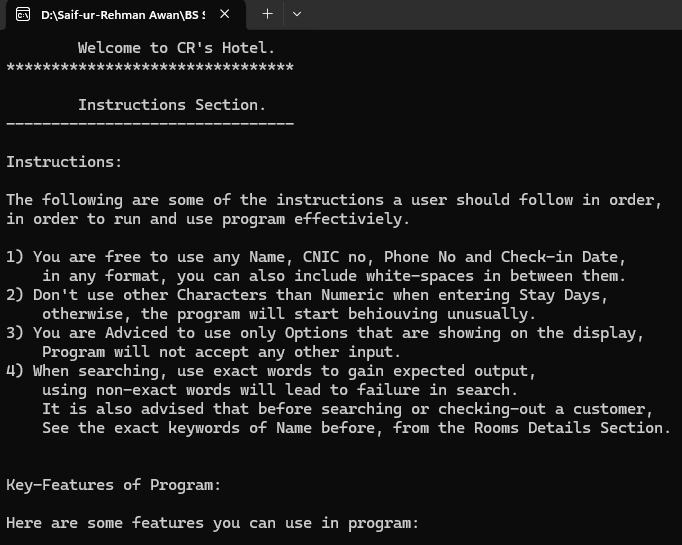


By pressing any key, you will be re-directed back to “Main Menu”.

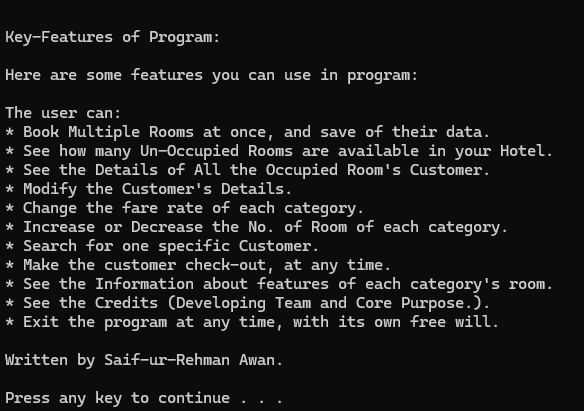
**(8) Instructions (Section):**

This section opens from the Main Menu by pressing “8” and displays “Instructions” and “Key-Features of the Program”.

At first, some instructions are being displayed about the how to use the program effectively and in a correct way.

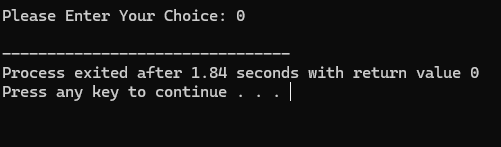


After it, some Key-features of the program are also given, telling the user what he/she can do inside the program, and what setting he can change to make it adorable and use-able.



**(9) Exit :**

This option allows user to exit program whenever they want. To use this option, user have to press “0” in the “Main Menu”, and program will be terminated.



Please keep in mind that this program doesn’t uses any external file to save customer’s data, therefore, upon terminating this program, all the entered/saved data will get lost, and can’t be retrieved again.

**The End**