**Bus Reservation System**

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

#include <time.h>

typedef struct BinarySearchTree BST;

// if bst is empty then we have to handle the error

struct BinarySearchTree

{

int PassnNo; // busNo0SeatNo.

char name[10];

struct BinarySearchTree \*left;

struct BinarySearchTree \*right;

};

BST \*root = NULL;

int cost(BST \*r); // calculates costs

void status(); // shows bus and seats status

void busLists(); // shows buslist and do booking seat and return customer ID

void DisplaySeat(int bus[33]); // Display the seats of buses

void cancel(int x); //cancel the booking

BST \*reservationInfo(BST \*, int, int \*); // Display Reservation Info

BST \*insert(BST \*\*r, int custID); // inserting a node

int busSeat[32][9] = {0};//When called, it changes the color of the console text to red using the ANSI escape sequence "\033[1;31m". This function is likely used to highlight important messages or warnings in the console.

void redColor() /// Print the message in redcolor

{

printf("\033[1;31m");// prints an escape sequence \033[1;31m, which is a control sequence recognized by most terminal emulators

}

void resetColor() /// reset the old color of console

{

printf("\033[0m");//When called, it resets the console text color to its default using the ANSI escape sequence "\033[0m".

}

BST \*reservationInfo(BST \*r, int s, int \*custIDmatched)

{

if (r == NULL)

return NULL;

BST \*presentnode = r;

while (presentnode)

{

if (presentnode->PassnNo == s)

{

\*custIDmatched = 1;

redColor();

printf("\n-----------------------------------------------------------------");

printf("\n|| NAME: %10s ||", (presentnode->name));

printf("\n|| CUSTOMER ID: %d ||", presentnode->PassnNo);

printf("\n|| BUS NUMBER: %d ||", (presentnode->PassnNo) / 1000);

printf("\n|| SEAT NUMBER: %d ||", (presentnode->PassnNo) % 100);

printf("\n|| TICKET COST: Rs.%d ||", cost(presentnode));

printf("\n-----------------------------------------------------------------");

resetColor();

printf("\nEnter the payment amount: ");

int payment;

scanf("%d", &payment);

presentnode->PassnNo += payment; // update the booking information with payment

printf("\nPayment successful.\n");

return r;

}

else if (presentnode->PassnNo > s)

presentnode = presentnode->left;

else

presentnode = presentnode->right;

}

return NULL;

}

BST \*insert(BST \*\*r, int custId)

{

if (\*r == NULL)

{

\*r = (BST \*)malloc(sizeof(BST));

(\*r)->PassnNo = custId;

if (\*r == NULL)

{

printf("No memory…");

return NULL;

}

else

{

(\*r)->left = (\*r)->right = NULL;

printf("\n ENTER THE PERSON NAME: ");

scanf("%s", &((\*r)->name));

// write to file

FILE \*fp;

fp = fopen("bookings.csv", "a");

fprintf(fp, "Cust\_ID\n%d,No\_of\_seats\n%d,SeatNo\n%d,Name\n%s", (\*r)->PassnNo, (\*r)->PassnNo/1000, (\*r)->PassnNo%100, (\*r)->name);

fclose(fp);

}

}

else

{

if ((\*r)->PassnNo > custId)

{

(\*r)->left = insert(&((\*r)->left), custId);

}

else if ((\*r)->PassnNo < custId)

{

(\*r)->right = insert(&((\*r)->right), custId);

}

}

return \*r;

}

void DisplaySeat(int bus[33])

{

for (int i = 1; i <= 32; i++)

{

redColor();

if (i < 10 && i > 0)

{

printf("0%d .", i);

}

else

{

printf("%d .", i);

}

resetColor();

{

if (bus[i] == 0)

printf("EMPTY ");

else

printf("BOOKED"); // reserv

}

printf(" ");

if (i % 4 == 0)

printf("\n");

}

}

void login()

{

char correctUser[20] = "myusername";

char correctPass[10] = "mypassword";

char matchUser[20];

char matchPass[10];

printf("\nEnter your desired username: ");

scanf("%s", correctUser);

printf("\nEnter your desired password: ");

scanf("%s", correctPass);

redColor();

printf("\n\n=========================================================================================\n");

printf("\n\t\t\tWELCOME TO ONLINE BUS RESERVATION");

printf("\n\n=========================================================================================\n\n");

resetColor();

while (1)

{

printf("\n\nUsername: ");

scanf("%s", matchUser);

printf("\nPassword: ");

scanf("%s", matchPass);

if (strcmp(matchUser, correctUser) == 0 && strcmp(matchPass, correctPass) == 0)

{

printf("\nLogged in successfully.\n");

break;

}

else

{

redColor();

printf("\nIncorrect username or password. Please try again.\n");

resetColor();

}

}

}

int cost(BST \*r)

{

int cost, buscost;

buscost = (r->PassnNo) / 1000;

switch (buscost % 9)

{

case 1:

return 700;

break;

case 2:

return 600;

break;

case 3:

return 400;

break;

case 4:

return 700;

break;

case 5:

return 550;

break;

case 6:

return 400;

break;

case 7:

return 700;

break;

case 8:

return 1000;

break;

case 9:

return 450;

break;

default:

return 0;

break;

}

}

void status()

{

int busNum;

busLists();

busInput:

printf("\n\nENTER YOUR BUS NUMBER : ");

scanf("%d", &busNum);

if (busNum <= 0 || busNum >= 10)

{

redColor();

printf("\n PLEASE ENTER CORRECT BUS NUMBER !!\n");

resetColor();

goto busInput;

}

printf("\n");

DisplaySeat(busSeat[busNum]);

getch();

}

void busLists()

{

redColor();

printf("-----------------------------------------------------------------------------------------");

printf("\nBus.No\tName\t\t\tDestinations \t\tCharges \t\tTime\n");

printf("-----------------------------------------------------------------------------------------");

resetColor();

printf("\n1\tChanakya Travels \tPune To Malkapur \tRs.700 \t\t07:00 AM");

printf("\n2\tChintamani Travels \tJalgaon To Surat \tRs.600 \t\t01:30 PM");

printf("\n3\tSangeetam Travels \tAhmednagar To Mumbai\tRs.400 \t\t03:50 PM");

printf("\n4\tSuper Deluxe \tNagpur To Pune \tRs.700 \t\t01:00 AM");

printf("\n5\tNi3 Travels \tNashik To Pune \tRs.550 \t\t12:05 AM");

printf("\n6\tPrathamesh Travels \tKolhapur To Pune \tRs.400 \t\t09:30 AM");

printf("\n7\tPurple Travels \tPune To Chandrapur \tRs.700 \t\t11:00 PM");

printf("\n8\tMaster Travels \tPune To Banglore \tRs.1000 \t\t08:15 AM");

printf("\n9\tVolvo Travels \tMumbai To Pune \tRs.400 \t\t04:00 PM");

printf("\n");

printf("\n PRESS 'ENTER' KEY TO CONTINUE ");

getch();

}

void cancel(int randomNum)

{

int reservationNo;

int seatNumber;

int choice;

char c;

int seatCancel;

aa:

{

printf("\nENTER YOUR RESERVATION NUMBER : ");

scanf("%d", &reservationNo);

if (reservationNo == randomNum)

{

printf("\nRESERVATION NUMBER IS IT CORRECT ? %d \nENTER (Y/N) : ", reservationNo);

scanf("%s", &c);

if (c == 'y' || c == 'Y')

{

printf("\n\n============================================\n\n");

printf(" ENTER THE BUS NUMBER: ");

scanf("%d", &choice);

printf("\n HOW MANY SEATS DO WANT TO CANCEL : ");

scanf("%d", &seatCancel);

for (int i = 0; i < seatCancel; i++)

{

printf(" \nENTER THE SEAT NUMBER: ");

scanf("%d", &seatNumber);

busSeat[choice][seatNumber] = 0;

}

printf("\n\nYOUR RESERVATION HAS BEEN CANCEL !!\n\n");

printf("\n PRESS 'ENTER' KEY TO CONTINUE \n");

getch();

DisplaySeat(busSeat[choice]);

}

else if (c == 'n' || c == 'N')

{

printf("\nYOUR RESERVATION CANCELATION HAS BEEN DENIED\n");

}

}

else

{

printf("\nNOT FOUND!! ENTER THE CORRECT RESERVATION NUMBER\n");

goto aa;

}

}

}

int main()

{

srand(time(0));

int randomNum = rand();

int num, i, custID, reservationNo;

BST \*root1;

login();

main:

{

do

{

printf("\n\n====================================================================\n\n");

printf("\t\t\t\033[1;31mBUS RESERVATION\033[0m\t\t");

printf("\n\n=====================================================================\n");

printf("\n====================");

redColor();

printf(" MAIN MENU ");

resetColor();

printf("=====================\n\n");

printf(" \033[1;31m[1]\033[0m VIEW BUS LIST \n\n");

printf(" \033[1;31m[2]\033[0m BOOK TICKETS\n\n");

printf(" \033[1;31m[3]\033[0m CANCEL BOOKING\n\n");

printf(" \033[1;31m[4]\033[0m BUSES SEATS INFO\n\n");

printf(" \033[1;31m[5]\033[0m RESERVATION INFO\n\n");

printf(" \033[1;31m[6]\033[0m EXIT\n");

printf("\n=====================================================\n");

printf("\n ENTER YOUR CHOICE: ");

scanf("%d", &num);

switch (num)

{

case 1:

busLists(); // for list of bus

break;

case 2:

busLists(); // for booking the tickets

int CustId, choice, seats;

busChoice:

printf("\n\nCHOOSE YOUR BUS : ");

scanf("%d", &choice);

if (choice <= 0 || choice > 9)

{

redColor();

printf("\nENTER VALID BUS NUMBER !! \n");

resetColor();

getch();

goto busChoice;

}

printf("\n");

DisplaySeat(busSeat[choice]);

busSeatChoice:

printf("\n\nNO. OF SEATS YOU NEED TO BOOK : ");

scanf("%d", &seats);

if (seats <= 0)

{

redColor();

printf("\nENTER VALID SEAT NUMBER!!\n");

resetColor();

goto busSeatChoice;

}

else if (seats > 32)

{

redColor();

printf("\nENTER VALID SEAT NUMBER WE HAVE ONLY 32 SEATS IN A BUS !!\n");

resetColor();

goto busSeatChoice;

}

int seatNumber;

for (int i = 1; i <= seats; i++)

{

printf("\n\n==================================================================================\n\n");

seat:

printf(" ENTER THE SEAT NUMBER: ");

scanf("%d", &seatNumber);

if (seatNumber <= 0)

{

redColor();

printf("\n ENTER VALID SEAT NUMBER!!\n\n");

resetColor();

goto seat;

}

else if (seatNumber > 32)

{

redColor();

printf("\n ENTER VALID SEAT NUMBER WE HAVE ONLY 32 SEATS IN A BUS !!\n\n");

resetColor();

goto seat;

}

CustId = choice \* 1000 + seatNumber; // CustumerId

busSeat[choice][seatNumber] = 1;

root = insert(&root, CustId);

redColor();

printf("\n YOUR CUSTOMER ID IS : %d", CustId);

resetColor();

printf("\n\n==================================================================================\n\n");

}

printf("\nYOUR RESERVATION NUMBER IS : ");

redColor();

printf("%d\n", randomNum);

printf("\nPLEASE NOTE DOWN YOUR RESERVATION NUMBER FOR CANCEL BOOKING TICKETS!!\n");

resetColor();

printf("PRESS 'ENTER' KEY TO CONTINUE ");

getch();

break;

case 3:

cancel(randomNum);

break;

case 4:

status(randomNum);

break;

case 5:

takingReservationNo:

printf("\n ENTER YOUR RESERVATION NUMBER :");

scanf("%d", &reservationNo);

if (randomNum == reservationNo)

{

cust:

printf("\n ENTER YOUR CUSTOMER ID :");

scanf("%d", &custID);

int custIDmatched = 0;

root1 = reservationInfo(root, custID, &custIDmatched);

if (custIDmatched == 0)

{

redColor();

printf("\n ENTER CORRECT CUSTOMER ID!!\n");

resetColor();

goto cust;

}

}

else

{

redColor();

printf("\n INVALID RESERVATION NUMBER PLEASE ENTER CORRECT RESERVATION NUMBER !!\n");

resetColor();

goto takingReservationNo;

}

break;

default:

redColor();

printf("\n\n INVALID INPUT CHOOSE CORRECT OPTION\n");

resetColor();

break;

}

} while (num != 6);

printf("\n\n=====================================================================\n\n");

printf("THANK YOU FOR USING THIS BUS RESERVATION SYSTEM");

printf("\n\nPRESS ANY KEY TO EXIT THE END PROGRAM !! \n");

printf("\n\n");

getch();

return 0;

}

}