BUS RESERVATION SYSTEM USING BINARY SEARCH TREE

A PROJECT REPORT

Submitted by

AKANKSHA MISHRA [RA2211003011513]
ADITYA ROY [RA2211003011539]
APARIJIT CHAKRABORTY [RA2211003011540]

for the course 21CSC201J Data Structures and Algorithms

Under the Guidance of

Dr. Ashokkumar C

Assistant Professor, Department of Computing Technologies

In partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

NOVEMBER 2023

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that the 21CSC201J Data Structures and Algorithms course project report titled **"BUS RESERVATION SYSTEM USING BINARY SEARCH TREE"** is the bonafide work done by **AKANKSHA MISHRA [RA2211003011513]**, **ADITYA ROY [RA2211003011539]**, **APARIJIT CHAKRABORTY [RA2211003011540] who carried out under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.**

Dr. Ashokkumar C,

Faculty In-Charge,

Assistant Professor,

Department of Computing Technologies,

SRMIST.

Dr. M. Pushpalatha,

Professor and Head,

Department of Computing Technologies,

SRMIST.

ABSTRACT

The Bus Reservation System is a command-line application designed to facilitate the booking and management of bus tickets. It offers several features, including viewing available buses, booking tickets, canceling reservations, checking bus seat status, and retrieving reservation information. Key Features:

1. Login System: Users are required to log in with a username and password (admin/password) to access the system.
2. View Bus List: Users can view a list of available buses, each with its name, destination, ticket price, and departure time.
3. Booking Tickets: Users can select a bus, choose the number of seats to book, and assign passenger names to each seat. The system generates unique customer IDs and reservation numbers for each booking.
4. Cancellation of Bookings: Users can cancel their reservations by providing the reservation number and seat numbers. The system updates the seat status accordingly.
5. Bus Seat Status: Users can check the current seat status of a selected bus, which indicates whether seats are available or booked.
6. Reservation Information: Users can retrieve reservation information by entering their reservation number and customer ID. The system displays passenger details, including name, customer ID, bus number, seat number, and ticket cost.

This bus reservation system helps users efficiently manage their bus bookings and provides essential information about bus availability and reservations. It aims to simplify the process of reserving and canceling bus tickets.

ACKNOWLEDMENT

TABLE OF CONTENTS

1. INTRODUCTION

The Bus Reservation System stands as an innovative software solution dedicated to revolutionizing and optimizing the intricate process of reserving and overseeing bus tickets. Its primary objective is to offer a user-friendly, secure, and efficient platform for both passengers planning their bus journeys and bus operators managing their services.

In the contemporary age where accessibility and simplicity hold utmost importance, the Bus Reservation System presents an array of features catering to the needs of passengers and bus operators alike. The system places a strong emphasis on user security, data precision, and a seamless booking experience.

Key Features of the Bus Reservation System:

1. Intuitive Interface: With a user-friendly interface, the system ensures easy navigation for all users, regardless of their technical expertise.

2. Stringent Security Measures: Prioritizing security, users must log in with their credentials to safeguard personal data and booking details.

3. Extensive Bus Listings: The system furnishes a comprehensive list of available buses, complete with essential details such as bus name, destination, fare, and departure times.

4. Effortless Booking: Booking bus tickets is made simple; passengers can choose their preferred bus, select seats, and assign passenger names. Each booking generates unique customer IDs and reservation numbers for easier management.

5. Easy Cancellation: For cancellations, users can input their reservation number and seat details, facilitating hassle-free cancellations.

6. Real-Time Seat Availability: Both passengers and bus operators can access real-time information regarding available seats, enabling optimal bus occupancy.

7. Reservation Information Retrieval: Passengers can effortlessly retrieve detailed reservation information by entering their reservation number and customer ID.

The Bus Reservation System aims to simplify the booking process and elevate the overall bus travel experience, catering to travellers seeking seamless bookings and operators striving for more efficient service management. It represents a significant advancement, offering a secure, convenient, and efficient solution to modernize bus travel planning and execution.

2. REQUIREMENTS

1. Purpose:

  • Simplify booking and managing bus tickets.

  • Enhance user experience for passengers and bus operators.

2. Functional Requirements:

  • Implement secure login and user authentication.

  • Maintain a database of buses with details (name, destination, fare, departure time).

  • Allow users to select a bus, specify seats, and assign passenger names.

  • Generate unique customer IDs and reservation numbers.

  • Enable reservation cancellation using reservation numbers and seat details.

  • Provide real-time seat availability for each bus.

  • Enable retrieval of detailed reservation info using reservation number and customer ID.

  • Create a user-friendly, intuitive interface.

  • Ensure data security through encryption and access controls.

3. Non-Functional Requirements:

  • Ensure data security and encryption for user privacy.

  • Optimize usability and performance for a smooth user experience.

  • Allow scalability for increased users and buses.

  • Implement data backup and recovery mechanisms.

  • Comply with accessibility standards for diverse user needs.

4. Constraints:

  • Developed for desktop or web platforms.

5. Assumptions:

  • Users possess basic bus booking and travel knowledge.

  • The system will be accessible on desktop or web platforms.

6. Acceptance Criteria:

  • The system operates fully functional and bug-free.

  • Provide comprehensive documentation for users and administrators.

  • Ensure security of user data and sensitive information.

  • Adhere to relevant regulations and accessibility standards.

3. IMPLEMENTATION

Code

```c
#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};
void redColor() /// Print the message in redcolor
{
  printf("\033[1;31m");
}
void resetColor() /// reset the old color of console
{
  printf("\033[0m");
```

```c
}
BST *reservationInfo(BST *r, int s, int *custIDmatched) // find function
{
  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();
      getch();
      return r;
    }
    else if (presentnode->PassnNo > s)
      presentnode = presentnode->left;
    else
      presentnode = presentnode->right;
  }

  return NULL;
}
BST *insert(BST **r, int custId)
```

```c
{
    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));
        }
    }
    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])
{
```

```c
    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 userName[20] = "admin";
  char passWord[10] = "password";
  char matchPass[10];
  char matchUser[10];
  int value;
  redColor();

printf("\n\n===================================================================
```

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


printf("\n\n=================================================================
=====================\n\n");
   resetColor();
login:
{
   printf("\n\nUserName: ");
   gets(matchUser);


   printf("\nPassWord: ");
   gets(matchPass);
}

   value = strcmp(passWord, matchPass); /// string compare is function defined in headerfile i.e string.h
   if (value != 0)
   {
     redColor();
     printf("\nINVALID DETAILS TRY AGAIN...\n");
     resetColor();
     goto login;
   }
   else
   {
     printf("\nLOGED IN SUCCESFULLY...\n");
   }
}
int cost(BST *r)
{
   int cost, buscost;
   buscost = (r->PassnNo) / 1000;
   switch (buscost % 3)
   {
```

```c
    case 1:
      return 70;
      break;
    case 2:
      return 55;
      break;
    case 0:
      return 40;
      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()
{
```

```c
    redColor();
    printf("--------------------------------------------------------------------------------------");
    printf("\nBus.No\tName\t\t\tDestinations \t\tCharges \t\tTime\n");
    printf("--------------------------------------------------------------------------------------");
    resetColor();
    printf("\n1\tGangaTravels        \tDharan to Kavre       \tRs.70  \t\t07:00  AM");
    printf("\n2\tPhaphara Travels     \tKavre To Dharan       \tRs.55   \t\t01:30  PM");
    printf("\n3\tShiv Ganga Travels  \tAllahabad To Gorakhpur\tRs.40   \t\t03:50  PM");
    printf("\n4\tSuper Deluxe        \tPokhara To Benigha    \tRs.70   \t\t01:00  AM");
    printf("\n5\tSai Baba Travels     \tMaitidevi To Janakpur \tRs.55   \t\t12:05  AM");
    printf("\n6\tShine On Travels     \tMadhubani to Patna    \tRs.40   \t\t09:30  AM");
    printf("\n7\tMayur Travels        \tPatna To Gaya         \tRs.70  \t\t11:00  PM");
    printf("\n8\tRajjo Travels       \tBegusarai To Patna    \tRs.55   \t\t08:15  AM");
    printf("\n9\tShree Travels        \tGaya To Chhapra       \tRs.40   \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);
```

```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;
    }
}
}
```

```c
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)
    {
```

```c
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();
```
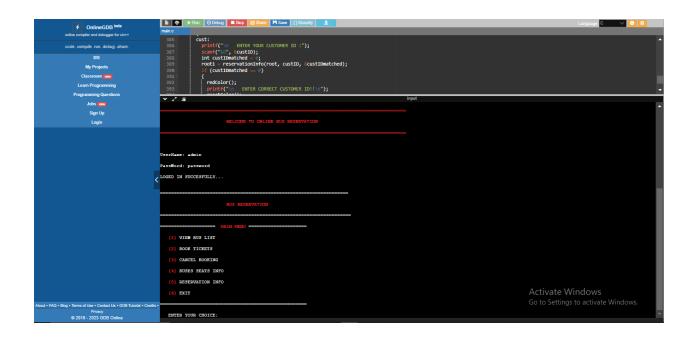
```c
        printf("\nENTER VALID SEAT NUMBER WE HAVE ONLY 32 SEATS IN A BUS !!\n");
        resetColor();
        goto busSeatChoice;
      }
      int seatNumber;
      for (int i = 1; i <= seats; i++)
      {

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();
```
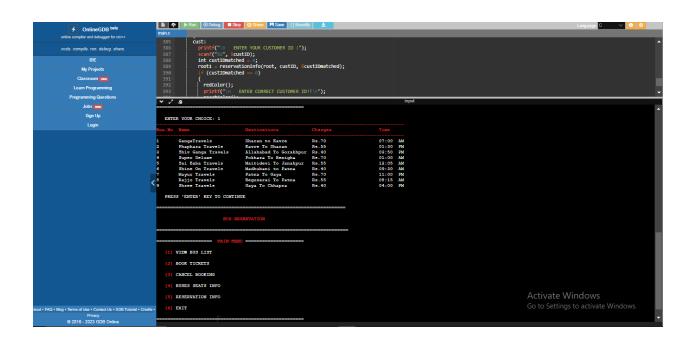
```c
printf("\n\n==================================================================
===============\n\n");
    }
    printf("\nYOUR RESERVATION NUMBER IS : ");
    redColor();
    printf("%d\n", randomNum);
    printf("\nPLEASE NOTE DOWN YOUR RESERVATION NUMBER FOR CANCEL BOOKING
TICKETS!!\n");
    resetColor();
    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();
```

```c
        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;
}
}
```

4. EXPERIMENT RESULTS & ANALYSIS

Output:

5. CONCLUSION

The Bus Reservation System represents a significant step forward in the realm of bus ticket booking and management. With a focus on convenience, security, and efficiency, this software application aims to improve the overall bus travel experience for both passengers and bus operators.

This system offers a range of features that cater to the diverse needs of its users. From a secure login system that protects sensitive information to comprehensive bus listings and an intuitive booking process, it simplifies the complexities of bus travel. Users can effortlessly book their tickets, cancel reservations when necessary, and obtain real-time information on seat availability.

Data security and privacy are paramount, with robust encryption and access controls in place. Scalability and performance have been considered to ensure that the system can accommodate a growing user base and a larger number of buses. In addition, the system complies with accessibility standards to make it inclusive for all users.

In conclusion, the Bus Reservation System is a valuable tool that not only enhances the bus travel experience but also offers bus operators the means to manage their services more efficiently. By prioritizing security, usability, and efficiency, this system sets a new standard for the industry. It empowers travelers to plan their journeys with ease and offers bus operators a streamlined way to serve their passengers. The Bus Reservation System represents a significant leap forward in the world of bus travel, and it holds the potential to revolutionize the way we book and manage our bus tickets.

6. REFERENCES

https://www.studytonight.com/c-projects/bus-reservation-system-project-using-c-language

https://www.geeksforgeeks.org/bus-reservation-system-in-c/

https://projectsgeek.com/2022/09/simple-bus-reservation-system-in-c-program.html