GOMENDRA MULTIPLE COLLEGE

Birtamode-04, Jhapa

Affiliated to Purbanchal University

Program: BCA-IT

Semester: First

Faculty of Science and Technology

Project report on

Airline Reservation System

Submitted by :
Sandesh Bhariti
Kundawa jabegu
Sajak Basnet

Under the supervision of
Assistant lecturer:
Mr. Nabin Prasain

College chief
Dr. Rupak Khanal

Acknowledgement

I would like to express my sincere appreciation to all individuals who played a vital role in the successful completion of the Project Air Ticketing system. I am particularly grateful to my supervisor, **Mr. Nabin Prasain**, for their invaluable guidance and unwavering support.

I extend my gratitude to the esteemed faculty members of **Gomendra Multiple College** for imparting their teachings and sharing their knowledge, which greatly contributed to the development of this project. Additionally, I would like to acknowledge my friends and classmates for their assistance and encouragement throughout the process.

I would also like to express my thanks to the authors of textbooks and online resources that significantly enhanced my understanding of C programming, which proved instrumental in the project's implementation.

Lastly, I am deeply indebted to my family for their constant support and belief in me throughout this effort.

The completion of this project would not have been possible without the collective efforts of all the individuals mentioned above. I take full responsibility for any remaining errors, and I would greatly appreciate it if they were brought to my attention.

Table of Contents

Introduction of Project

The "Airline Reservation System" represents an advanced software solution created as an integral component of the BCA 1st-semester project. Its primary purpose is to streamline and enhance the process of ticket reservation, thus promoting efficient management of transportation services. This system has been meticulously crafted to encompass a wide array of ticket reservation functions, spanning user authentication to comprehensive data recording. It effectively tackles the fundamental challenges associated with reservations, guaranteeing data accuracy, preventing duplicate bookings, and safeguarding user confidentiality.

Key Features:

This project offers a range of key features, including:

1. User-Friendly Interface: Provides an easy-to-use interface for booking flights.
2. Ticket Reservation: Allows users to book tickets with personal and travel details.
3. Ticket Display: offers a convenient way to view ticket information.
4. Cancellation: Enable passengers to cancel reservations effortlessly.
5. Data Persistence: Stores passenger and flights for accuracy and retrieval.
6. Security: Implements user authentication to ensure data privacy and retrieval.
7. Date Selection: Allows users to select departure and return dates for round-trip bookings.
8. Flight Selection: Provides a range of flight options for passengers to choose from.

Objectives of the Project

Project Overview:

In collaboration with a team of three, we developed an Airline Reservation System during our first semester. This system streamlined flight booking processes, offering users a simple interface for selecting flights, managing seat availability, and storing passenger information. Our project exemplified effective teamwork and practical application of software development skills within the context of airline operations.

Objectives:

1. Simplify the ticket reservation process for passengers.
2. Enable passengers to view available flights and select preferred options.
3. Store passenger information, including personal details, flight preferences, and seat assignments.
4. Generate and display ticket details upon request.
5. Allow passengers to cancel their reservations if needed.

Importance of Airline Reservation System

The Airline Reservation System (ARS) stands as a cornerstone of the aviation industry, underpinning its operational efficiency and success. This system holds immense importance for several reasons. Firstly, it simplifies and expedites the booking process, allowing passengers to reserve flights, select preferred seats, and complete transactions seamlessly. This user-friendly interface greatly enhances the overall passenger experience. Moreover, the ARS empowers airlines with dynamic pricing capabilities, enabling them to optimize ticket costs based on factors such as demand, seat availability, and booking timing. This dynamic pricing strategy maximizes revenue potential.

Another crucial role of the ARS lies in seat management. Airlines can efficiently monitor seat inventory, accommodate passenger seat preferences, and allocate seats based on individual requests or passenger loyalty status. Additionally, the system aids in route optimization, leveraging data analytics to identify high-demand routes, adjust flight schedules, and enhance overall operational efficiency. This results in cost savings and improved customer service. Furthermore, the ARS facilitates comprehensive passenger information management, including personal details, booking history, and travel preferences, which airlines can utilize to offer personalized services, thereby fostering passenger loyalty.

In summary, the Airline Reservation System is the linchpin of modern air travel, driving efficiency, revenue generation, and customer satisfaction. Its multifaceted role in managing bookings, optimizing routes, and personalizing services positions it as an indispensable tool for the airline industry's continued growth and success.

System Requirements

Minimum System Requirements:

1. Operating System: Windows, Linux, or macOS.
2. Processor: Dual-core processor (e.g., Intel Core i3 or equivalent).
3. Memory (RAM): 2GB of RAM or higher.
4. Storage: 50MB of free disk space for program installation and data storage.
5. Input Devices: Standard keyboard and mouse for user interactions.
6. IDE (integrated Development environment) like Dev C++, Turbo C++, Code blocks.
7. Code editors like Visual studio code, sublime text with GCC with MinGW version 6+ installed.

Source Code

```
// total time wasted 34 hours

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

#define MAX_USERS 100
#define MAX_USERNAME_LENGTH 50
#define MAX_PASSWORD_LENGTH 50
#define MAX_PASSENGERS 50
#define MAX_BUFFER_SIZE 100  // for Password auth
#define MAX_STRING_LENGTH 50 // Defining maximum length of string
inputs
char name_[MAX_STRING_LENGTH];
char type_ticket;
char log_in;
char bookedBy[MAX_USERNAME_LENGTH];

// Defining  maximum length for flight names
#define MAX_FLIGHT_NAME_LENGTH 20

// Defining the maximum length for seat class
#define MAX_SEAT_CLASS_LENGTH 20

// structure of login and signup authentication
struct User
{
  char username[MAX_USERNAME_LENGTH];
  char password[MAX_PASSWORD_LENGTH];
};

typedef struct User User;
```

```c
User users[MAX_USERS];
int userCount = 0;

// structure which holds details of passenger
typedef struct passenger
{
  char name[MAX_STRING_LENGTH];
  int age;
  char gender;
  char from[MAX_STRING_LENGTH];
  char to[MAX_STRING_LENGTH];
  char flight_no[MAX_STRING_LENGTH];
  int seat_no;
  char seat_class[MAX_SEAT_CLASS_LENGTH];
  char departure_date[MAX_STRING_LENGTH];
  char return_date[MAX_STRING_LENGTH];

} Passenger;

// structure holds flight details
typedef struct flight
{
  char name[MAX_FLIGHT_NAME_LENGTH];
} Flight;

// Array of string (All flight names)
Flight flights[] = {{"Flight-1"},
                    {"Flight-2"},
                    {"Flight-3"},
                    {"Flight-4"}};

int num_flights = sizeof(flights) / sizeof(flights[0]); //
returns total size of flights[] array

void display_flights()
{
  printf("Available flights:\n");
  for (int i = 0; i < num_flights; i++)
```

```
  {
    printf("%d) %s\n", i + 1, flights[i].name);
  }
}

int get_flight_index(int option) { return option - 1; }

int get_seat_number()
{
  int seat_no;
  printf("Enter seat number: ");
  scanf("%d", &seat_no);
  return seat_no;
}

// method to save user data to a file
void saveUserDataToFile()
{
  FILE *file = fopen("userdata.txt", "w");
  if (file == NULL)
  {
    printf("Error opening file for writing.\n");
    exit(1);
  }

  for (int i = 0; i < userCount; i++)
  {
    fprintf(file, "%s %s\n", users[i].username,
users[i].password);
  }

  fclose(file);
}

// method that loads data from the file
void loadUserDataFromFile()
{
  FILE *file = fopen("userdata.txt", "r");
  if (file == NULL)
```

```
  {
    return; // user data doesnot exist
  }

  char username[MAX_USERNAME_LENGTH];
  char password[MAX_PASSWORD_LENGTH];

  while (fscanf(file, "%s %s", username, password) != EOF)
  {
    strcpy(users[userCount].username, username);
    strcpy(users[userCount].password, password);
    userCount++;
  }

  fclose(file);
}

// method to check if a username is already taken
int isUsernameTaken(const char *username)
{
  for (int i = 0; i < userCount; i++)
  {
    if (strcmp(users[i].username, username) == 0)
    {
      return 1; // Username is taken
    }
  }
  return 0; // Username is not taken
}

// Function to sign up a new user
void signup()
{
  char username[MAX_USERNAME_LENGTH];
  char password[MAX_PASSWORD_LENGTH];

  printf("Enter a new username: ");
  scanf("%s", username);
```

```
  if (isUsernameTaken(username))
  {
    printf("Username is already taken. Please choose a different
one.\n");
    return;
  }

  printf("Enter a password: ");
  scanf("%s", password);

  strcpy(users[userCount].username, username);
  strcpy(users[userCount].password, password);
  userCount++;

  printf("Signup successful! You can now log in.\n");

  // Save the updated user data to the file
  saveUserDataToFile();
}

// Function to log in a user
int login()
{
  char username[MAX_USERNAME_LENGTH];
  char password[MAX_PASSWORD_LENGTH];

  printf("Enter your username: ");
  scanf("%s", username);
  strcpy(bookedBy,username);
  printf("Enter your password: ");
  scanf("%s", password);

  for (int i = 0; i < userCount; i++)
  {
    if (strcmp(users[i].username, username) == 0 &&
strcmp(users[i].password, password) == 0)
    {
      printf("\nLogin successful! Welcome, %s!\n", username);
      return 1;
```

```
    }
  }

  printf("Login failed. Please check your username and
password.\n");
  return 0;
}

void reserve_ticket(Passenger *passengers, int *num_passengers)
{
  FILE *fp;
  Passenger p;

  printf("\nEnter Passenger Details:\n");
  printf("Name: ");
  scanf("%s", p.name);
  strcpy(name_, p.name);
  strcat(name_, ".txt");
  fp = fopen(name_, "a");
  printf("Age: ");
  scanf("%d", &p.age);
  printf("Gender (M/F): ");
  scanf(" %c", &p.gender);
  printf("From: ");
  scanf("%s", p.from);
  printf("To: ");
  scanf("%s", p.to);
  printf("Do you want a return Ticket [y/n]: ");
  scanf(" %c", &type_ticket);
  getchar(); // Consume the newline character

  if (type_ticket == 'y')
  {
    printf("Enter your departure date: ");
    fgets(p.departure_date, MAX_STRING_LENGTH, stdin);

    printf("Enter your return date: ");
    fgets(p.return_date, MAX_STRING_LENGTH, stdin);
  }
```

```
else
{
  printf("Enter your departure date: ");
  fgets(p.departure_date, MAX_STRING_LENGTH, stdin);
}

display_flights();
printf("Enter your option: ");
int option;
scanf("%d", &option);

if (option < 1 || option > num_flights)
{
  printf("Invalid flight option!\n");
  option = 0;
  printf("Enter flight options again: ");
  scanf("%d", &option);
}

strcpy(p.flight_no, flights[get_flight_index(option)].name);

printf("Enter seat class: ");
scanf("%s", p.seat_class);

if (strcmp(p.seat_class, "business") == 0)
{
  p.seat_no = get_seat_number();
}
else
{
  p.seat_no = get_seat_number();
}

if (*num_passengers >= MAX_PASSENGERS)
{
  printf("Maximum number of passengers reached!\n");
  return;
}
```

```
  passengers[*num_passengers] = p;
  *num_passengers += 1;

  // Code for writing ticket details in file
  fprintf(fp,
"================================================================
================\n");
  fprintf(fp, "                        Ticket Details of %s\n",
p.name);
  fprintf(fp,
"================================================================
================\n");
  fprintf(fp, "Name: %-20s Age: %-
2d                        Gender: %c\n", p.name, p.age,
p.gender);
  fprintf(fp, "From: %-20s To: %-20s        Flight Number: %s\n",
p.from, p.to, p.flight_no);
  fprintf(fp, "Class: %-18s  Seat Number: %-
2d                  Departure Date: %s\n", p.seat_class,
p.seat_no, p.departure_date);
  fprintf(fp, "Booked By: %-20s\n",bookedBy);

  if (type_ticket == 'y')
  {
    fprintf(fp, "Return Date: %-20s", p.return_date);
  }

  fprintf(fp,
"\n================================================================
======\n");
  fclose(fp);

  printf("\nTicket Reserved Successfully!\n");
}

void display_ticket(Passenger *passengers, int num_passengers)
{
  char name[MAX_STRING_LENGTH];
  printf("\nEnter Passenger Name: ");
```

```
  scanf("%s", name);

  int found = 0;
  for (int i = 0; i < num_passengers; i++)
  {
    if (strcmp(passengers[i].name, name) == 0)
    {
      found = 1;
    printf("============================================================
============================\n");
    printf("                          Ticket Details of %s\n",
name);
    printf("============================================================
============================\n");
      printf("Name: %-20s Age: %-
2d                            Gender: %c\n", passengers[i].name,
passengers[i].age, passengers[i].gender);
      printf("From: %-20s To: %-20s         Flight Number: %s\n",
passengers[i].from, passengers[i].to, passengers[i].flight_no);
      printf("Class: %-18s  Seat Number: %-
2d                   Departure Date: %s\n",
passengers[i].seat_class, passengers[i].seat_no,
passengers[i].departure_date);
      printf("Booked By: %-20s",bookedBy);

      if (type_ticket == 'y')
      {
        printf("Return Date: %-15s", passengers[i].return_date);
        break;
      }

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

    if (!found)
    {
      printf("\nPassenger Not Found!\n");
    }
```

```
  }
}

void cancel_reservation(Passenger *passengers, int
*num_passengers)
{
  char *file_name;
  char name[MAX_STRING_LENGTH];

  printf("\nEnter Passenger Name: ");
  scanf("%s", name);

  int found = 0; // Firstly indicating found as 0 (FALSE)
  for (int i = 0; i < *num_passengers; i++)
  {
    if (strcmp(passengers[i].name, name) == 0)
    {
      found = 1; // if passenger is found indicating 1 (TRUE)

      file_name = strcat(name, ".txt");
      if (remove(file_name) == 0)
      {
        printf("\nReservation Cancelled for %s\n",
passengers[i].name);
      }

      for (int j = i; j < *num_passengers - 1; j++)
      {
        passengers[j] = passengers[j + 1];
      }
      *num_passengers -= 1;
      break;
    }
  }

  if (!found)
  {
    printf("\nPassenger Not Found!\n");
  }
```

```
}

void displayLoadingBar()
{
  int total = 20, progress, i;
  for (i = 0; i <= total; i++)
  {
    progress = i * 100 / total;

    // Display progress bar
    printf("\rLoading: [");
    int j;
    for (j = 0; j < i; j++)
    {
      printf("=");
    }
    printf(">");
    for (j = i + 1; j <= total; j++)
    {
      printf(" ");
    }
    printf("] %d%%", progress);
    fflush(stdout);

    struct timespec sleepTime;
    sleepTime.tv_sec = 0;
    sleepTime.tv_nsec = 100000000L; // 100 milliseconds
    nanosleep(&sleepTime, NULL);
    printf("\n");
  }
}

void displayChoice()
{
  printf("\n\t\t What Would You Like To Do?");
  printf("\n\t\t_____________________________________\n\n");
  printf("\n\t\t 1. Reserve Ticket");
  printf("\n\t\t 2. Display Ticket");
  printf("\n\t\t 3. Cancel Ticket");
```

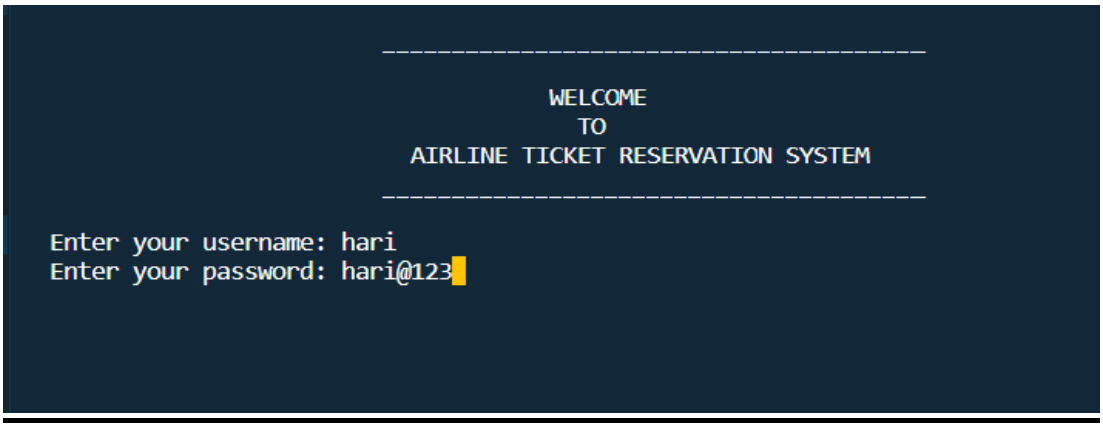
```
  printf("\n\t\t 4. Exit\n");
  printf("\n\t\t_____\n\n");
}

int isLoggedIn(const char *username, const char *password){
      for (int i = 0; i < userCount; i++) {
        if (strcmp(users[i].username, username) == 0 &&
strcmp(users[i].password, password) == 0) {
            printf("Login successful! Welcome, %s!\n", username);
            strcpy(bookedBy, username);
            return 1;
        }
    }

    return 0;

}

// will write main function to match the final requirement
int main()
{

  Passenger passengers[MAX_PASSENGERS];
  int num_passengers = 0;
  int i;
  int firstIteration = 0; // flag that checks how many iteration
has occured
  int choice, option;
  char userName[MAX_STRING_LENGTH], password[MAX_STRING_LENGTH];
  char buffer[MAX_BUFFER_SIZE];
  // char username[MAX_BUFFER_SIZE];
  // char Password_[MAX_BUFFER_SIZE];
  int loggedIn = 0;
  int flag = 0;
  // char login;

  loadUserDataFromFile(); // load user data from the file at the
start of the program
```

```
  printf("\n\t\t\t_____________________________________");
  printf("\n");
  printf("\n\t\t\t\t    WELCOME ");
  printf("\n\t\t\t\t      TO ");
  printf("\n\t\t\t  AIRLINE TICKET RESERVATION SYSTEM  ");
  printf("\n\t\t\t_____________________________________");
  printf("\n\n");

    while(1) {
  if(flag == 0){
    printf("Enter your username: ");
    scanf("%s", userName);
    printf("Enter your password: ");
    scanf("%s", password);
    flag = isLoggedIn(userName, password);

    if(flag == 0){
      printf("Login Failed. Please check your username and
password\n");
      continue; //asking for input again
    }

  }
        printf("\n");
        displayChoice(); // function that displays the choice
        printf("Enter Your Choice: ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
          reserve_ticket(passengers, &num_passengers);
          break;
        case 2:
          display_ticket(passengers, num_passengers);
          break;
        case 3:
          cancel_reservation(passengers, &num_passengers);
          break;
        case 4:
```

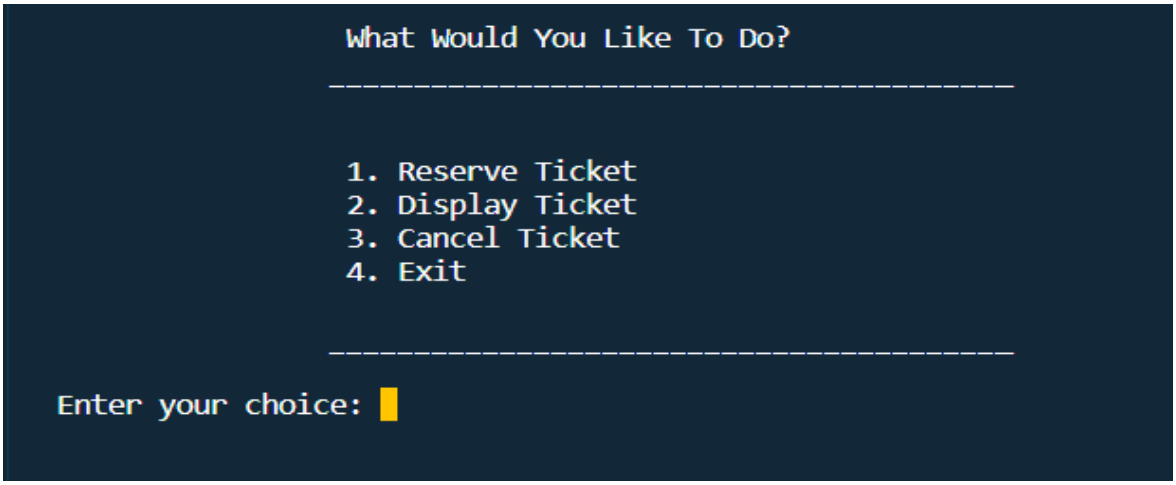
```
            printf("\nThank you for Choosing Us.\n\n\n");
            exit(0);
        default:
            printf("Invalid choice. Please try again.\n");
        }
    }
    return 0;
}
```

Output/ Application Overview Windows

Authentication



Menu section



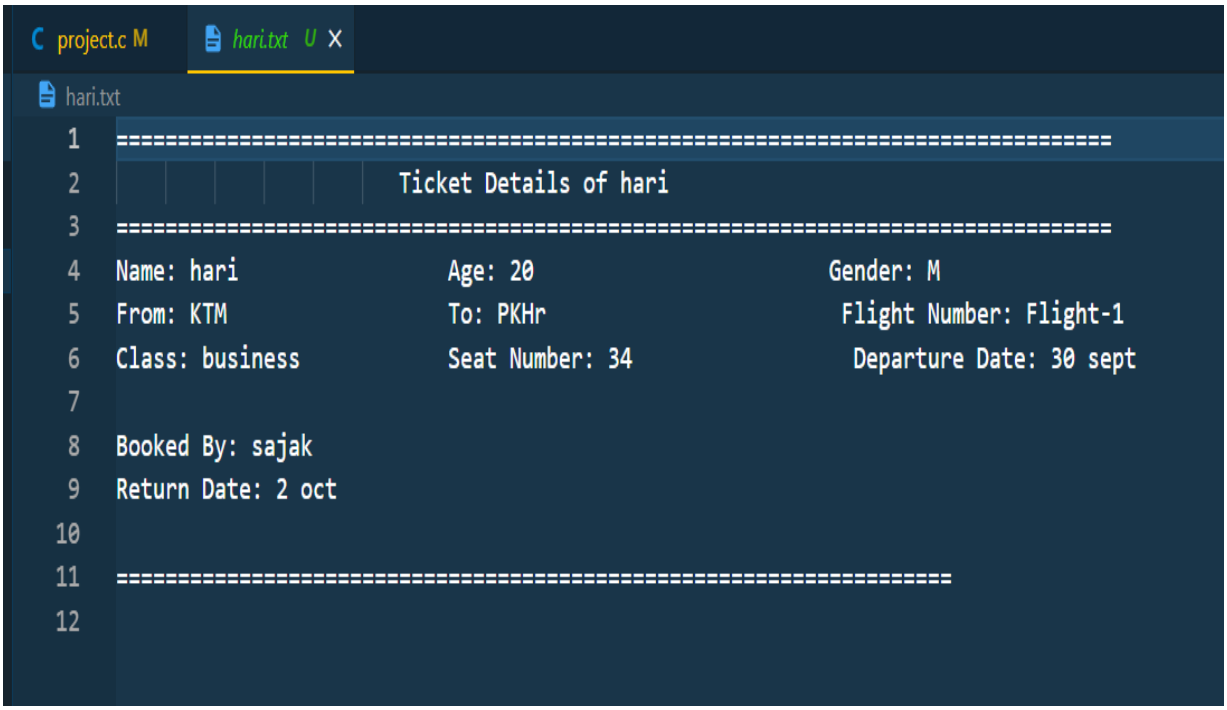
Reserving ticket by the user

```
                  What Would You Like To Do?
                 ----------------------------------------

                  1. Reserve Ticket
                  2. Display Ticket
                  3. Cancel Ticket
                  4. Exit

                 ----------------------------------------
Enter your choice: 1

Enter Passenger Details:
Name: ramesh
Age: 30
Gender (M/F): M
From: kathmandu
To: Pokhara
Do you want a return Ticket [y/n]: n
Enter your departure date: 14 september
Available flights:
1) Flight-1
2) Flight-2
3) Flight-3
4) Flight-4
Enter your option: 2
```

Displaying the Reserved Ticket

```
Enter Your Choice: 2

Enter Passenger Name: hari
================================================================================
                    Ticket Details of hari
================================================================================
Name: hari              Age: 20                 Gender: M
From: KTM               To: PKHr                Flight Number: Flight-1
Class: business         Seat Number: 34         Departure Date: 30 sept

Booked By: sajak            Return Date: 2 oct
```

Displaying Ticket Details in file



Abstract

The project under the topic “Airline Ticket Reservation System” using C-programming language provides the low-level access to the computer hardware. This project meets almost all the computerized demands required in the field of Ticket Reservation. The predominant purpose of this system is to manage the air transportation system in an effective manner.

C programming language is one of the most popular programming language due to its efficiency and modularity. It allows the user to break complex program into modular functions which makes work simpler. Our system provides lots of features like reserve, display, cancel, exit etc . C programming also provide file handling system which is the main key to develop our system. Our system is the comprehensive system consisting of integrated modules for various aspects of Ticket Reservation. Through this system, administration can easily manage customers details effortlessly. This system is made in a user-friendly interface so that users can easily access it without any difficulties.

Future Use and Implementation

The Airline Reservation System project holds substantial potential for future applications and enhancements within the airline industry. One prospective avenue is the adaptation of the system to support more sophisticated booking and ticketing features, including personalized passenger profiles, flexible fare management, and seamless integration with other travel services. Such improvements can elevate the overall travel experience for passengers and streamline airline operations. Additionally, advancements in data analytics can empower airlines to make data-driven decisions, optimizing routes, pricing, and resource allocation.

Moreover, the project can serve as a foundation for innovation in passenger safety and security. Integrating advanced technologies such as biometric authentication and real-time security checks can enhance the security measures at airports and during flights. This can contribute to safer and more efficient air travel, which is of paramount importance in the aviation industry.

Beyond airlines, the Airline Reservation System can find utility in related sectors such as hospitality and tourism. Collaborations with hotels, car rental services, and travel agencies can create a comprehensive travel ecosystem that caters to the diverse needs of travelers. By offering personalized recommendations and travel packages, the system can contribute to a more enjoyable and hassle-free travel experience.

Furthermore, the adaptability and scalability of the system make it suitable for a range of applications beyond traditional airline reservations. It can be customized to support booking and scheduling needs in various domains, including event management, conference planning, and facility reservations. As the travel and hospitality industries continue to evolve, the Airline Reservation System remains a versatile platform capable of meeting the evolving demands of travelers and organizations alike

Conclusion

In conclusion, the development of our Airline Reservation System has provided our team of novice programmers with invaluable learning experiences and insights. We are deeply grateful to our college's management for their unwavering support and resources, as well as to our friends and teachers for their guidance and suggestions. While the project was not without its challenges, it has equipped us with a deeper understanding of programming logic and the C language. We view this project as a significant milestone and look forward to refining and expanding it in the future to cater to the evolving needs of the airline industry. We extend our heartfelt thanks to everyone who supported and encouraged us throughout this rewarding journey.