

A PROJECT REPORT ON

**AIRLINE REGESTRATION SYSTEM**

SUBMITTED BY:

# Miss. Warule Divya Raosaheb

(2124UCEF1016)

SUBJECT: **C++** PROGRAMMING

Under the guidance of

**Prof. Ishwari Tirse**

Department of **Computer Science and Engineering**

Sanjivani Rural Education Society’s SANJIVANI UNIVERSITY

KOPARGAON – 423603 (2024-2025)

# INDEX

|  |  |  |
| --- | --- | --- |
| Sr No. | Content | Page No. |
| 1 | INTRODUCTION | 3 |
| 2 | CODE | 4 |
| 3 | OUTPUT | 9 |
| 4 | CONCLUSION | 12 |

# INTRODUCTION

The Airline Reservation System is a simple console-based application written in C++ that facilitates the booking and management of airline seats. Designed for ease of use, the program allows users to book available seats, cancel existing bookings, and view both available and booked seats. With a user-friendly menu interface, it operates in a loop, enabling users to perform multiple actions until they choose to exit. This system serves as a foundational example of how to manage reservations efficiently, making it an ideal project for beginners looking to understand basic programming concepts and data management in C++.

# CODE

# #include <iostream>

# #include <vector>

# #include <string>

# using namespace std;

# 

# class AirlineReservationSystem {

# private:

# int totalSeats;

# vector<bool> seats; // true means booked, false means available

# public:

# AirlineReservationSystem(int total) : totalSeats(total), seats(total, false) {}

# void bookSeat(int seatNumber) {

# if (seatNumber < 1 || seatNumber > totalSeats) {

# cout << "Invalid seat number." << endl;

# return;

# }

# if (seats[seatNumber - 1]) {

# cout << "Seat " << seatNumber << " is already booked." << endl;

# } else {

# seats[seatNumber - 1] = true;

# cout << "Seat " << seatNumber << " has been successfully booked." << endl;

# }

# }

# void cancelSeat(int seatNumber) {

# if (seatNumber < 1 || seatNumber > totalSeats) {

# cout << "Invalid seat number." << endl;

# return;

# }

# if (!seats[seatNumber - 1]) {

# cout << "Seat " << seatNumber << " is not booked." << endl;

# } else {

# seats[seatNumber - 1] = false;

# cout << "Booking for seat " << seatNumber << " has been cancelled." << endl;

# }

# }

# void viewAvailableSeats() {

# cout << "Available seats: ";

# for (int i = 0; i < totalSeats; ++i) {

# if (!seats[i]) {

# cout << (i + 1) << " ";

# }

# }

# cout << endl;

# }

# void viewBookedSeats() {

# cout << "Booked seats: ";

# for (int i = 0; i < totalSeats; ++i) {

# if (seats[i]) {

# cout << (i + 1) << " ";

# }

# }

# cout << endl;

# }

# };

# int main() {

# AirlineReservationSystem ars(10); // Let's assume there are 10 seats

# int choice, seatNumber;

# do {

# cout << "\nAirline Reservation System Menu:\n";

# cout << "1. Book Seat\n";

# cout << "2. Cancel Seat\n";

# cout << "3. View Available Seats\n";

# cout << "4. View Booked Seats\n";

# cout << "5. Exit\n";

# cout << "Enter your choice: ";

# cin >> choice;

# switch (choice) {

# case 1:

# cout << "Enter seat number to book: ";

# cin >> seatNumber;

# ars.bookSeat(seatNumber);

# break;

# case 2:

# cout << "Enter seat number to cancel: ";

# cin >> seatNumber;

# ars.cancelSeat(seatNumber);

# break;

# case 3:

# ars.viewAvailableSeats();

# break;

# case 4:

# ars.viewBookedSeats();

# break;

# case 5:

# cout << "Exiting the system." << endl;

# break;

# default:

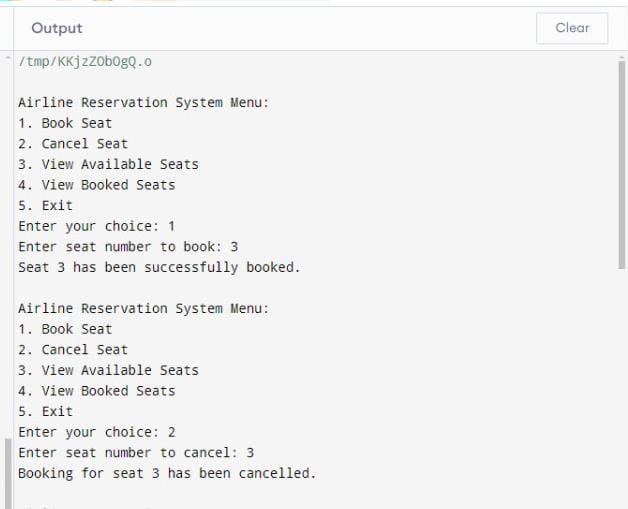
# cout << "Invalid choice. Please try again." << endl;

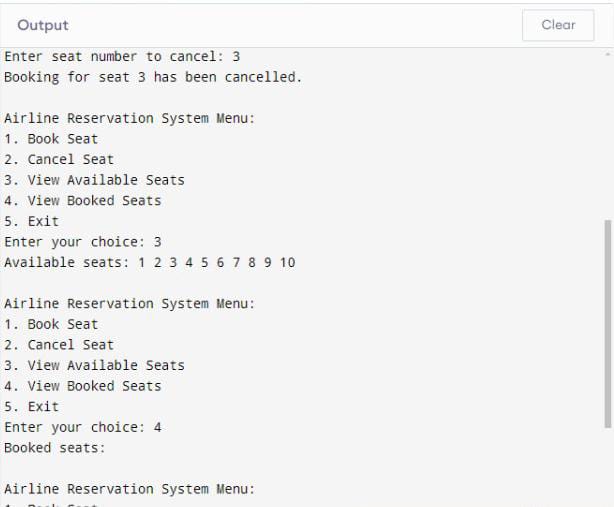
# }

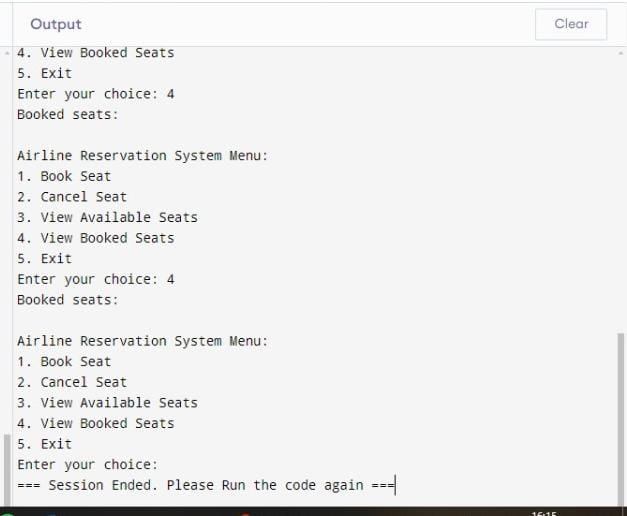
# } while (choice != 5);

# return 0;

# }







CONCLUSION

In conclusion, the Airline Reservation System exemplifies a straightforward yet effective approach to managing seat bookings in a simulated airline environment. By allowing users to easily book, cancel, and view seat availability, the program demonstrates fundamental programming principles such as data handling, user interaction, and control structures. This project not only provides practical experience in C++ programming but also lays the groundwork for more advanced features and improvements, such as integrating a graphical user interface, handling multiple flights, or implementing a database for persistent storage. Overall, this system serves as a valuable learning tool for aspiring developers in the field of software engineering.