Date :- 04 / 12 / 2024

Project Name :- Airline Management System

Description :-

The Airline Management System is a software tool created to make managing airline operations easier and more efficient. It handles key tasks like scheduling flights, booking tickets, and managing passenger reservations. By automating these processes, the system reduces errors and improves productivity, all while offering a simple and user-friendly interface. Built with Object-Oriented Programming (OOP) principles in C++, it's designed to be flexible, scalable, and easy to maintain—making it a practical solution for real-world airline needs.

Components Used :-

1. **Programming Language : C++(Object-Oriented Programming)**

2. **Classes and Objects :**
   # Flight Class :- Manages flight details (e.g., ID, source, destination, schedule, seats).
   # Passenger Class :- Passengers information (e.g., Name , Contact Info , Passenger Id etc.)
   # Reservation Class :- Links passengers to flights, managing bookings and Cancellations.
   # AirlineSystem Class :- Control hole system (e.g. , Add Flight , Add Passengers , Passengers Tickets , Reservations , Tickets Cancellation etc).

3. **System Features:**
   # Flight Management: Add, view, and manage flights.
   # Passenger Management: Add and maintain passenger records.
   # Ticket Booking: Real-time seat reservation for flights.
   # Reservation Management: View and cancel reservations

4. **Data Structures:**
   # Vectors: Used for dynamic storage and management of flights, passengers, and reservations.
5. **User Interface:**
   # Console-based menu-driven system for input/output operations.

6. **Error Handling:**
   # Input validation for managing invalid or duplicate entries.

Objectives :-

1. **Automation - Eliminate manual processes for managing flights, bookings, and reservations, thereby improving operational efficiency.**

2. **Accuracy and Reliability - Reduce errors in seat allocation, passenger details, and flight schedules.**

3. **Scalability - Allow future expansion to support advanced features like payment integration, flight tracking, and analytics.**

4. **Learning and Implementation - Demonstrate the practical application of OOP principles in solving real-world problems. Provide a functional prototype that highlights the power and flexibility of C++.**