AIRLINE MANAGEMENT SYSTEM

OVERWIEW

- A simple console-based flight booking system written in C++
- Allows admins to manage flights and passengers to book tickets
- **FEATURES: add, update, and delete flights, booking tickets, and cancelling bookings**

BENIFITS:

- Easy to use
- provides a simple and effective way to book flights
- Helps reduce manual errors and increase productivity

SYSTEM STRUCTURE

- Admin Panel
- Passenger Panel
- Flight Management (add, update, delete)
- Booking Management (book, cancel)

Benefits:

- Modular design makes it easy to modify and maintain
- Clear separation of concerns between admin and passenger panels

LIBRARIES AND FUNCTIONS

#include <iostream>: Input/Output Stream Library **(e.g., cout, cin)**

#include <string>: String Library (e.g., string data type)

#include<sstream>: String Stream Library (e.g., ostringstream)

#include<Limits>: Limits Library (e.g., numric_limits)

#include<cctye>: character testing\manipulation

#include<stdexcept>: handle the error through standard exceptions (**like** runtime_error, invalid_arrgument)

Functions:

adminPanel(), passengerPanel(), addFlight()

ARRAYS AND DO-WHILE LOOP

Arrays: flight flights[MAX_FLIGHTS]; , booking bookings[MAX_BOOKINGS];

Do-While Loop: do { … } while (choice != 3);

Why use it?

- **Arrays** provide efficient storage and access to multiple values
- **Do-While Loop** ensures code execution at least once before checking the condition

Benefits:

- Improves code efficiency
- Ensures code execution at least once

SWITCH STATEMENT

Purpose: Execute different blocks of code based on a variable's value

Example:

```
switch (choice) { ... }
```

Why use it?

- More readable and efficient than multiple if-else statements
- Improves code maintainability

Benefits:

- Improves code readability
- Reduces code duplication

EXAMPLE USE CASE

- Admin logs in and adds a new flight (uses **adminPanel()** and **addFlight()**)
- Passenger views available flights and books a ticket (uses **passengerPanel()** and **bookTicket()**)
- Passenger cancels a booking using their booking ID (uses **cancelTicket()**)

Benefits:

- Demonstrates the system's functionality
- Shows how the system can be used in real-world scenarios

OSTRINGSTREAM

ostringstream is a class in C++ that allows you to insert data into a **string stream,** which is a sequence of characters. It is similar to **cout**, but instead of printing the data to the console, it stores it in a string.

intToString

In the **intToString** function, **ostringstream** is used to convert an integer to a string.

Benefits:

- Allows easy conversion of data types to strings
- Flexibility string building
- Efficient performance

EXCEPTION HANDLIG

Mechanism to handle **runtime errors** without **crashing** the program.

Keywords:

try, catch, throw

How it works:

1. **try block** → Code that may cause an error
2. **throw** → Raise an exception when an error occurs
3. **catch block** → Handle the exception

Benefits:

- Prevents program crashes
- Makes code robust and maintainable
- Separates normal logic from error-handling logic

THANK YOU