**Airline Management System**

**Submitted By**:

* Muhammad Ibrahim *[24K-0649]*
* Muhammad Hasnain Siddiqui *[24K-0516]*
* Muhammad Obaid Ullah *[24K-0793]*

**Course**: Object Oriented Programming [OOP]  
**Instructor**: Sir Shafique Rehman  
**Submission Date**: 11/05/2025

**1. Executive Summary:**  
This project is a menu-driven **Airline Management System** built in C++ using **OOP principles**. It supports two user roles—**Admin** and **Customer**—with distinct functionalities. Admins can manage flights (add, remove, edit) and view customer data, while customers can book, view, and cancel flights. The system features **custom console manipulation functions** for an organized terminal interface.

**2. Introduction**

**Background**:  
Airline systems require structured data management and user-specific access controls. This project was chosen to **practice core C++ OOP concepts** (classes, inheritance, file handling) while simulating a real-world scenario.

**Objectives**:

* Practice real-world application of C++ through terminal-based airline reservation simulation.
* Implement **user authentication** (login/registration) for Admins and Customers.
* Provide **role-based access**:
  + **Admin**: Flight management (add/remove/edit), customer list review.
  + **Customer**: Flight booking, cancellation, and booking history.

**3. System Description**

**Features**:

* **Role-Based Access Control:** Distinct functionalities and interfaces for Admin and Customer roles.
* **Terminal-Based UI:** Structured output using custom print() and display() functions to enhance readability.
* **Persistent Storage:** Use of file handling for storing and retrieving user credentials and flight data.

**4. Implementation**

**Development Process**:

* Designed class hierarchy with inheritance: User (base), Admin, Customer, and Flight.
* Implemented file I/O operations for reading/writing user and flight information.
* Developed menu-driven interface using switch-case logic for role-specific navigation.

**Tools**:

* **Language**: C++
* **Version Control**: Git/GitHub
* **IDE**: VSCode

**Challenges**:

* Managing **console output formatting** (solved with custom display functions).
* Ensuring **data consistency** in file operations (handled via error-checking).

**5. Results**

* Successfully demonstrated **OOP principles** (encapsulation, polymorphism).
* Achieved **persistent storage** for users/flights via text files.
* Terminal UI improved readability with **aligned columns and borders**.

**6. References**

* <https://www.learncpp.com/> [To learn basics of C++ programming and OOP principles]
* Lab Manuals [To learn basics of C++ programming and OOP principles]
* https://www.geeksforgeeks.org/ [Syntax]
* <https://youtu.be/RXzzE2wnnlo?si=z5d5dkUFA2xGnmDh> [Vectors in C++]