Documentation (User Manual) for Flight Reservation System

Group Name: CodeBhairav
Group Members:-

1. Priyanshu Mahato:
    - Roll No.: 21MS002
    - Email ID: pm21ms002@iiserkol.ac.in
2. Soham Roy Chowdhury:
    - Roll No.: 21MS055
    - Email ID: src21ms055@iiserkol.ac.in

Following is the documentation (or the user manual) accompanying Flight Reservation System.c, a very basic simulation of an actual Flight Ticket Reservation System, explaining what different parts of the code do, system requirements, and how to run it.

Some preliminary information: In this code, we have tried to exploit the concepts and usage of structures and pointers. We have taken the risk of using concepts that haven't been discussed in class and are probably outside the scope of the syllabus covered in the course. To learn about these concepts, various online resources were used (including GPT). We also encountered several errors and bugs during the process, and the internet was used profusely to figure out how to rectify them. Our main focus was on structures and pointers, so figuring everything out and rectifying the errors (you might get an idea about the effort we have put into learning these concepts in depth when you see the code) took most of our time, and that prevented us from using file handling in our code. We actually tried using file handling, but it wasn't working according to our expectations, and we did not have enough time to debug the relevant sections (you can still view it as a commented section of code as we haven't removed it). But this can be interpreted as the following disclaimer:

"Our flight ticket reservation system has the utmost user privacy!" Once you close the program, all existing user and flight data is erased, as we don't store any user data. This prevents any breach of data from our side. So, you must exercise extreme caution while choosing from the options on the main interface, especially the "Exit" option. Additionally, the users are requested to print the tickets as soon as they are issued, and the operators are required to enter the user details in an offline register for future reference (in case you plan to close the program after some time).

System Requirements:-

1. **OS**: Pop! OS, a Debian-based GNU/Linux operating system, was used to write the code
2. **Execution**: The code can easily be executed in any Linux terminal (tested on gnome/zsh) and on WindowsPowerShell.
3. **Compiler:** gcc was used to compile the C code
4. **PDF Reader:** This document was created using Google Docs and saved as a pdf. It opens fine using Okular and shouldd be supported by most other pdf readers.

How to use the code:-

First things first, the code doesn't use any external database to extract/read user, admin or flight data. To differentiate between a user and an admin, we have used a unique "Employee ID". Only employees have unique employee IDs which are required to log in every time, whereas users do not have anything similar and can log in only using their username and password. The admins can't log in as users, and users can't log in as admins. So, the first thing to be done after running the code is to sign up as an admin and enter the required flight details (it displays an error message if you try to use "Admin Utilities" without signing up as an admin first). After doing that, if the admin wants, he/she can edit the flight details if required; otherwise, not (if required, the admin can edit the flight details at any point in time). Then, if the admin wants, he/she can view all the flight details; again, this can be done at any point of time the admin wishes to do so. Next, sign up as a user if you haven't signed up yet (it shows an error message if you try to log in without signing up first). Now use the "Book Tickets" option to log in as a user and book tickets by entering the desired source and destination. Now that you have booked the tickets, you may print them and proceed with packing. In case you want to view the tickets you have booked so far, you may use the View Booked Tickets option and have a look. The admin can also view "all" the booked tickets by logging in as an admin within the View Booked Tickets option.

Unfortunately, there is no cancellation or refund policy (due to lack of time), and you may just not show up instead of cancelling your ticket. (Well, if you are someone we know, something can be done!)

Contributions by group members:

Although almost every aspect of the code was brainstormed, written, and debugged by both of us, the broad breakdown of functions written by each member is as follows:

Priyanshu Mahato: signUp(), bookTickets(), adminUtilities(), saveDataToFile(), loadDataToFile(), viewBookedTickets(), Interface Design

Soham Roy Chowdhury: enterFlightDetails(), editFlightDetails(), viewAllFlightDetails(), displayDetails(), freePerson(), freeDatabase()

Apart from contributing to these modules, Soham also helped a lot in testing the code after every step.

Example Runs: Since the run is quite long, the complete information and sample run are provided in the video sent along with this file. (https://drive.google.com/file/d/1uPr73h9cqX12shqAPq2mgNtBmmnaZPu4/view?usp=drive_link)