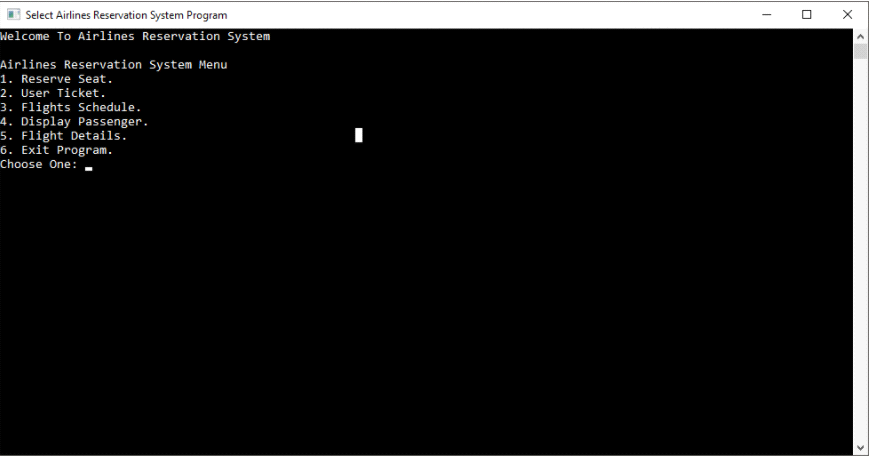
**The program features:**

* Reserve Seat.
* User Ticket.
* Flights Schedule.
* Display Passenger.
* Flight Details.

1. Add Flight.
2. Edit Flight.
3. Delete Flight.
4. Flight Leave and Arrive.
5. Back To Menu.



**The switch case is used for switching between these functions:**

* ReserveSeat();
* UserTicket();
* FlightSchedule();
* DisplayPassenger();
* FlightDetails();

**All of this comes under the FUNCTION FlightDetail()s:**

**OUR STRUCT() WILL INCLUDE ONE MORE VALUE THAT IS TOTAL SEATS ON THE FLIGHT. This is for when we reserve seats, we don't go over the number of seats available on a flight and pretty obviously, one plane cannot carry unlimited people.**

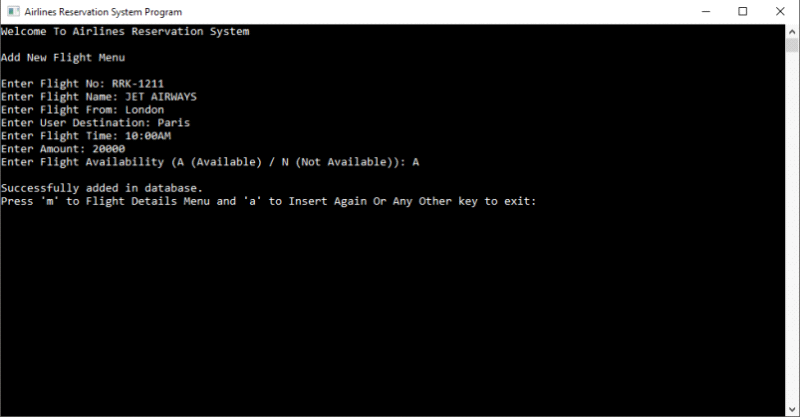
\* Based on my mood i might or might not add a security pin system to access flight details because only authorized personnel should be able to edit flight details.

This will take os to a new screen using clrscr() (clear screen) and give you the current flights recorded (pre written/hard-coded). We will use our own struct() for this. Next the user will be given another menu to do call functions like;

* AddNewFlight();
* EditFlightRecord();
* DeleteFlight();
* FlightLeaveArrive();
* Back to Main Menu;

**1- AddNewFlight**()

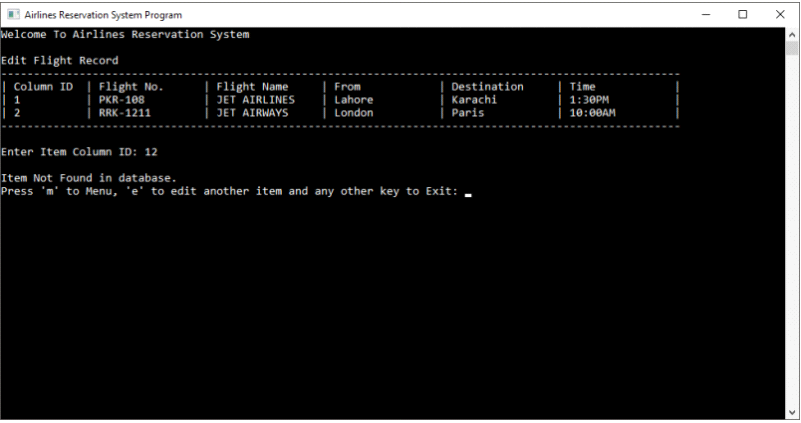
Here we will add new flight so that customers can reserve (customers are also us) seats for this flight.

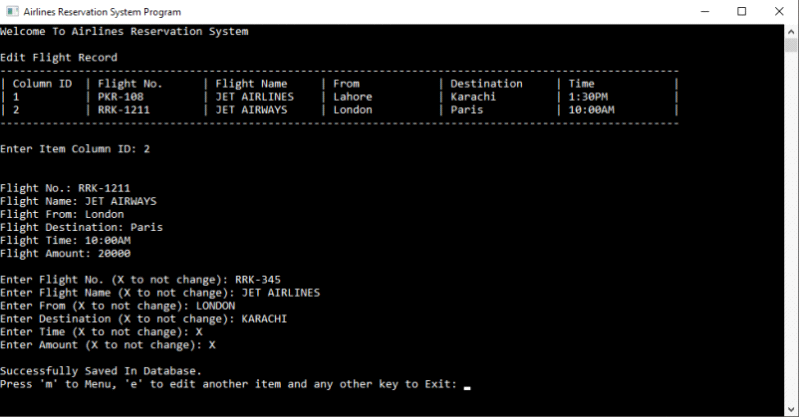


After succesfully entering flight, ask the user to either RETURN TO THE MAIN MENU or CONTINUE ADDING.

**2- EditFlightRecord()**

This function will list out all the current flight we have entered so that we can choose and edit them.

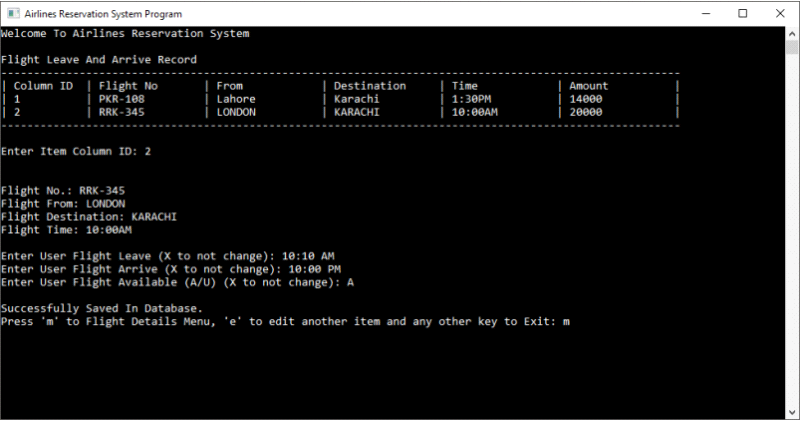




After successfully editing flight, ask the user to either RETURN TO THE MAIN MENU or CONTINUE EDITING.

**3- FlightLeaveArrive()**

Here we can change set and change the flight arrival and departure timings because the program needs to be close to reality and delays happen in real-life.



**4- DeleteFlight()**

Pretty simple this one. Quite literally does what it says, deletes flight records.

**ReserveSeats()**

Inputs are:

* departure location
* destination

based on these inputs we will display the customer the available flights for them, the timings, amount, seats available etc. Basically, all the details of the available flights on according to their departure and destination location.

them as the user which flight and how many seats they want to reserve according to the available flights and seats.

After that is done, Display the ticket information (seats and flight details)

RETURN TO MAIN MENU.

**FlightSchedule()**

Input: **Flight Number**

Show flight details, pakhrat customers like langra and tuuti ungli check their details multiple times a day to see if there's been a change.

RETURN TO MAIN MENU.

**DisplayPassengers()**

INPUT: Passenger name and code

OUTPUT: all the flights details of the, current/future/previous flights and seats reserved by that user.

**What to use?**

* switch cases, a lot of switch cases
* functions (duh!)
* structure for flight details
* structure for passengers.
* nested structure technique to connect flight details to passenger details.
* clrscr() command to clear the screen everytime we go to the main menu and when a function is called.
* And other normal ones you already know about.