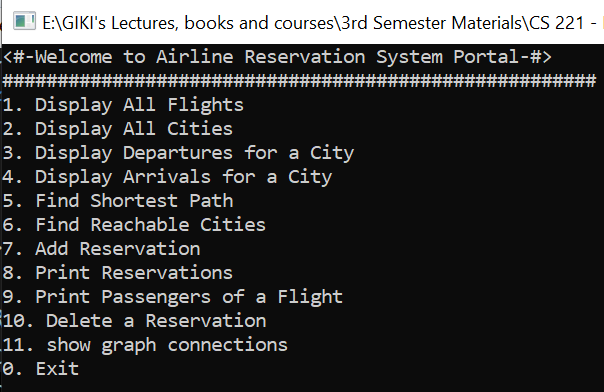


*Project by:***SAJJAD AKBAR KHAN (2023634) M. HAMMAD (2023420) REHMAT ALI (2023609)**

*Submitted to Dr. Ahmer Rashid*

**Airline Reservation System Project rEPORT**

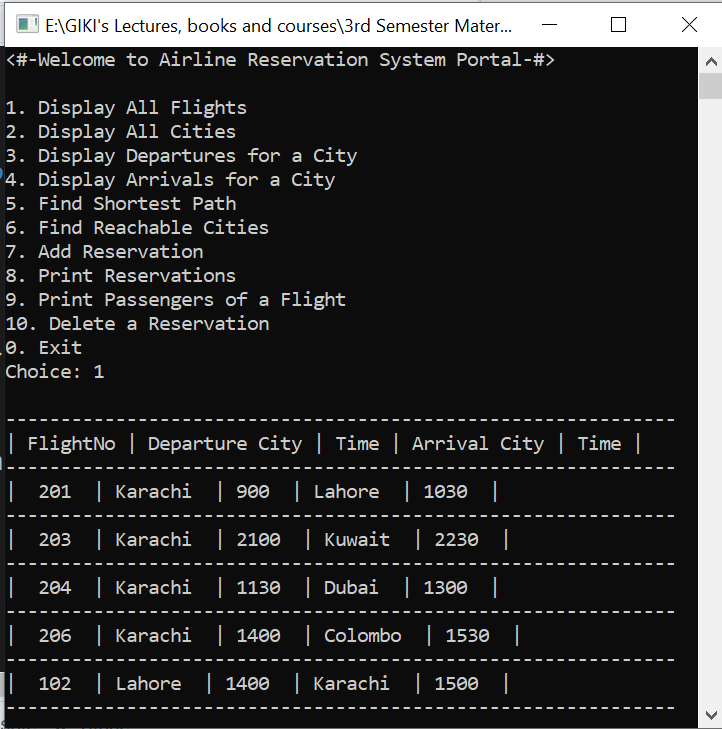
DSA - CS 221

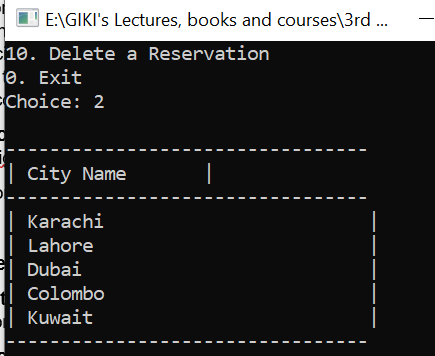
**Introduction:**

This program implements a comprehensive **Flight Management System**, enabling efficient management and querying of flights, cities, and passenger reservations. The system incorporates various functionalities, including data display, graph-based traversal, and reservation management.

***Features***

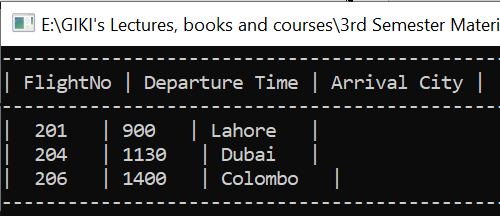
**1. Display All Flights**

* **Functionality**: Lists all flights in the system with detailed information such as flight number, departure city, arrival city, departure time, arrival time, and passenger count.
* **Key Function**: void DisplayAllFlightsData()
* **Output**: A formatted table of flight details.

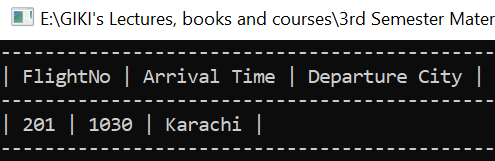
**2. Display All Cities**

* **Functionality**: Lists all cities in the system that have at least one flight (departure or arrival).
* **Key Function**: void DisplayAllCities()
* **Output**: A simple list of city names.

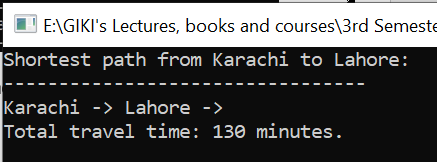
**3. Display Departures for a City**

* **Functionality**: Displays all flights departing from a specified city. Provides flight details sorted by departure time.
* **Key Function**: void DisplayDeparturesForCity(const string& cityName)
* **Error Handling**: Validates whether the city exists before displaying flights.
* **Output**: A list of departing flights or a message indicating no flights found.

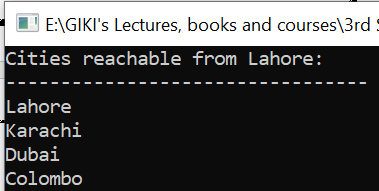
**4. Display Arrivals for a City**

* **Functionality**: Lists all flights arriving at a specified city. Details are sorted by arrival time.
* **Key Function**: void DisplayArrivalsForCity(const string& cityName)
* **Error Handling**: Checks for city existence before processing.
* **Output**: A list of arriving flights or an error message if none exist.

**5. Find Shortest Path**

* **Functionality**: Computes the shortest path between two cities based on flight travel time using **Dijkstra's Algorithm**.
* **Key Function**: void DijkstraShortestPath(const string& startCity, const string& endCity)
* **Output**: Displays the path between cities and the total travel time or a message if no path exists.

**6. Find Reachable Cities**

* **Functionality**: Uses **Breadth-First Search (BFS)** to determine all cities that can be reached from a specified city.
* **Key Function**: void BFSReachableCities(const string& cityName)
* **Output**: A list of reachable cities or an error message if no cities are found.

**7. Add Reservation**

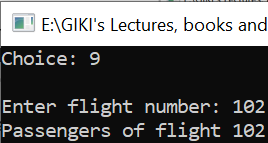
* A screen shot of a computer

  Description automatically generated**Functionality**: Adds a new passenger reservation for one or more flights. Updates passenger counts on the respective flights.
* **Key Function**: void AddReservation(const string& name, const vector<int>& flightNos)
* **Validation**: Checks the availability of flights and updates passenger details.
* **Output**: Confirmation of the reservation or an error message if flights are invalid.

**8. Print Reservations**

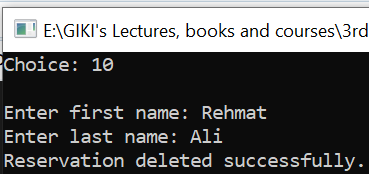
* A screen shot of a computer

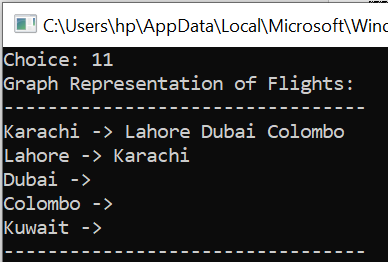
  Description automatically generated**Functionality**: Displays all passenger reservations in the system, including their names and the flights they are booked on.
* **Key Function**: void PrintReservations()
* **Output**: A detailed list of all reservations or a message indicating no reservations exist.

**9. Print Passengers of a Flight**

* **Functionality**: Lists all passengers on a specific flight, identified by the flight number.
* **Key Function**: void PrintPassengersOfFlight(int flightNo)
* **Validation**: Verifies the existence of the flight before processing.
* **Output**: A list of passenger names or a message if no passengers are found.

**10. Delete a Reservation**

* **Functionality**: Removes a passenger's reservation by name and updates the flight's passenger count accordingly.
* **Key Function**: void DeleteReservation(const string& name)
* **Output**: Confirmation of the deletion or an error message if the reservation does not exist.

1. **Show Graph Connections**  
   • **Functionality**: Displays the flight routes between cities, illustrating the cities each city is directly connected to via flights.  
   • **Key Function**: void DisplayGraphConnections()  
   • **Output**: A list of cities and their directly connected cities, visualizing the flight network.

**Summary**

This system provides a robust solution for flight and passenger management. It combines efficient algorithms like *sorting, hashing,* and *graph traversal* with dynamic data structures for flexible data handling.

The modular design ensures ease of use, while the feature set caters to essential needs for managing flights and reservations. Further enhancements, such as a graphical user interface or real-time data integration, could make the system even more versatile.

*<The End>*