Air Travel Project

Group 1
Amit Talmale
Amit Saraswat
Neelesh Rana
Keshav Pareek

Data:-

Airport we used ;

IATA Code	Airport Name	Location	Latitude	Longitude
BOM	Chhatrapati Shivaji Maharaj International Airport	Mumbai	19.0896	72.8656
BLR	Kempegowda International Airport	Bangalore	13.1986	77.7066
DEL	Indira Gandhi International Airport	New Delhi	28.5681	77.1126
MAA	Chennai International Airport	Chennai	12.9900	80.1690
CCU	Netaji Subhas Chandra Bose International Airport	Kolkata	22.6522	88.4466
HYD	Rajiv Gandhi International Airport	Hyderabad	17.2316	78.4292
COK	Cochin International Airport	Kochi	10.1520	76.4017
PNQ	Pune Airport	Pune	18.5822	73.9197
GOI	Goa International Airport	Dabolim, Goa	15.3801	73.8319
AMD	Sardar Vallabhbhai Patel International Airport	Ahmedabad	23.0778	72.6348
JAI	Jaipur International Airport	Jaipur	26.8241	75.8129
GAU	Lokpriya Gopinath Bordoloi International Airport	Guwahati	26.1060	91.5859
TRV	Trivandrum International Airport	Thiruvananthapuram	8.4821	76.9209
CCJ	Calicut International Airport	Kozhikode	11.1395	75.9553
BBI	Biju Patnaik International Airport	Bhubaneswar	20.2520	85.8173
VTZ	Visakhapatnam International Airport	Visakhapatnam	17.7215	83.2245
IXB	Bagdogra Airport	Bagdogra	26.6841	88.3247
IXC	Chandigarh International Airport	Chandigarh	30.6737	76.7894
IDR	Devi Ahilya Bai Holkar Airport	Indore	22.7215	75.8011
PAT	Jai Prakash Narayan International Airport	Patna	25.5941	85.0908
IXR	Birsa Munda Airport	Ranchi	23.3171	85.3215

Here's the flight information ;

Flight No.	Departure	Destination	Departure Time	Departure Date	Aircraft
SG501	BOM	MAA	03:30 PM	22nd March 2024	aircraft1
9W601	CCU	BLR	04:00 PM	23rd March 2024	aircraft2
AI401	DEL	HYD	05:30 PM	24th March 2024	aircraft3
6E501	BOM	GOI	06:00 PM	25th March 2024	aircraft4

SG601	MAA	CCU	07:30 PM	26th March 2024	aircraft5
AI501	DEL	COK	08:00 PM	27th March 2024	aircraft6
6E601	BOM	PNQ	09:30 PM	28th March 2024	aircraft1
SG701	BLR	GOI	10:00 PM	29th March 2024	aircraft2
AI801	DEL	AMD	11:30 PM	30th March 2024	aircraft3
6E901	MAA	JAI	12:00 AM	31st March 2024	aircraft4
SG901	BLR	GAU	01:30 AM	1st April 2024	aircraft5
AI1001	BOM	TRV	02:00 AM	2nd April 2024	aircraft6
6E1101	DEL	CCJ	03:30 AM	3rd April 2024	aircraft1
SG1101	MAA	BBI	04:00 AM	4th April 2024	aircraft2

Here's the aircraft details.

Aircraft	Capacity	Economy Seats	Business Seats	First Class Seats
Boeing 747	150	50	20	20
Airbus A320	120	30	15	15
Embraer E190	100	20	10	10
Bombardier Q400	80	15	5	5
ATR 72	70	10	5	5
Boeing 737	130	40	15	15

Explanation which algorithm and data structure we used

1. Unordered Map: The unordered_map containes to store mappings between strings (such as flight numbers, airport codes) and corresponding objects (such as Flight, Airport).

2. Graph Representation: The flight connections between airports are represented using an adjacency list, stored in flight Graph, which is **a unordered_map<string, vector<string>>**. So, we can easily traverse.

3. Haversine Formula: The calculateDistance method computes the distance between two airports using the Haversine formula.

4. seats are initialized and stored in the "availableSeats" vector.

5. OOPs:

a. Classes: The code defines several classes, each representing a real-world entity:

- Seat: Represents a seat on an aircraft.
- Aircraft: Represents an aircraft with its model and seat counts.
- Flight: Represents a flight with details such as flight number, departure airport, destination airport, departure time, date, aircraft information, and available seats.
- Airport: Represents an airport with details such as airport code, name, location, latitude, and longitude.

- Airline: Represents an airline and manages flights and airports.

b. Encapsulation: Each class encapsulates its data members and provides methods to interact with them. Like:

The Flight class encapsulates details about a flight and provides methods to initialize seats, display seat details, etc.The Airport class encapsulates details about an airport and provides methods to access its information.

c. Abstraction: The classes abstract away complex implementation details. Users interact with objects through well-defined interfaces without needing to know the internal workings.

d. Inheritance: While not explicitly demonstrated in the provided code, inheritance could be used to model relationships such as different types of aircraft sharing common attributes and behaviours.

CODE:-

```
#include <iostream>
#include <unordered_map>
#include <vector>
#include <string>
#include <cmath>

using namespace std;

// Enum for seat types  // enum used for epresent a fixed set of constants.
enum class SeatType {
    ECONOMY,
    BUSINESS,
    FIRST_CLASS
};

// Class to represent seat details
class Seat {
public:
    SeatType seatType;
    string seatNumber;

    // Constructor to initialize seat type and seat number
    Seat(SeatType type, const string& number) : seatType(type),
seatNumber(number) {}
};

// Class to represent an aircraft
class Aircraft {
public:
    string model;
```

```
    size_t economySeats;
    size_t businessSeats;
    size_t firstClassSeats;

    // Constructor to initialize aircraft model and seat counts
    Aircraft(const string& mdl, size_t economy, size_t business, size_t
firstClass)
        : model(mdl), economySeats(economy), businessSeats(business),
firstClassSeats(firstClass) {}
};

// Class to represent a flight
class Flight {
public:
    string flightNumber;
    string departureAirport;
    string destinationAirport;
    string departureTime;
    string date;
    Aircraft aircraft;  // Include an instance of the Aircraft class
    vector<Seat> availableSeats;

    // Constructor to initialize flight details
    Flight(const string& number, const string& departure, const string&
destination,
           const string& time, const string& d, const Aircraft& ac)
        : flightNumber(number), departureAirport(departure),
destinationAirport(destination),
          departureTime(time), date(d), aircraft(ac) {
              // Initialize available seats
              initializeSeats();
          }

    // Add any additional flight details or methods as needed

    // Display seat details for the flight
    void displaySeatDetails() const {
        cout << "Seat Details for " << flightNumber << ":" << endl;

        displaySeatDetailsByClass(SeatType::ECONOMY, "Economy Class");
        displaySeatDetailsByClass(SeatType::BUSINESS, "Business Class");
        displaySeatDetailsByClass(SeatType::FIRST_CLASS, "First Class");

        cout << endl;
    }

private:
```

```
    void displaySeatDetailsByClass(SeatType seatClass, const string&
className) const {
        cout << "    " << className << ":" << endl;
        for (const auto& seat : availableSeats) {
            if (seat.seatType == seatClass) {
                cout << "      " << seat.seatNumber;
            }
        }
        cout << endl;
    }


    // In this function  we initialize the seat and push in the dynamic array
    void initializeSeats() {
        // Economic Class
        for (size_t i = 1; i <= aircraft.economySeats; ++i) {
            availableSeats.push_back(Seat(SeatType::ECONOMY, "E" +
to_string(i))); //it creates a vector of Seat objects named availableSeats
        }

        for (size_t i = 1; i <= aircraft.businessSeats; ++i) {
            availableSeats.push_back(Seat(SeatType::BUSINESS, "B" +
to_string(i)));
        }

        for (size_t i = 1; i <= aircraft.firstClassSeats; ++i) {
            availableSeats.push_back(Seat(SeatType::FIRST_CLASS, "F" +
to_string(i)));
        }
    }
};

// Class to represent particular airports
class Airport {
public:
    string airportCode;
    string airportName;
    string location;
    double latitude;
    double longitude;

    // Constructor to initialize airport details including latitude and
longitude
    Airport(const string& code, const string& name, const string& loc, double
lat, double lon)
        : airportCode(code), airportName(name), location(loc), latitude(lat),
longitude(lon) {}

};
```

```
// Class to represent the airline and manage flights and airports
class Airline {
private:
    unordered_map<string, Airport> airports;
    unordered_map<string, Flight> flights;
    unordered_map<string, vector<string>> flightGraph; // Adjacency list
representation

public:
    void addAirport(const string& code, const string& name, const string&
location, double lat, double lon) {
        airports.emplace(code, Airport(code, name, location, lat, lon));
    }

    void addFlight(const string& number, const string& departure, const
string& destination,
                   const string& time, const string& date, const Aircraft&
aircraft) {
        flights.emplace(number, Flight(number, departure, destination, time,
date, aircraft));
        flightGraph[departure].push_back(destination); // Add directed edge
    }

    // Display flight schedule
    void displayFlightSchedule() const {
        cout << "Flight Schedule:" << endl;
        for (const auto& flight : flights) {
            string airlineName = getAirlineName(flight.first); // Get airline
name from flight number
            cout << "Airline: " << airlineName << endl; // Display airline
name
            cout << "Flight Number: " << flight.second.flightNumber << endl;
            cout << "   Departure Airport: " << flight.second.departureAirport
<< endl;
            cout << "   Destination Airport: " <<
flight.second.destinationAirport << endl;
            cout << "   Departure Time: " << flight.second.departureTime <<
endl;
            cout << "   Date: " << flight.second.date << endl;
            cout << "   Aircraft Model: " << flight.second.aircraft.model <<
endl;
            cout << "   Available Seats: " <<
flight.second.availableSeats.size() << endl;

            cout << endl;
        }
    }
```

```
    // Display airport details
    void displayAirportDetails() const {
        cout << "Airport Details:" << endl;
        for (const auto& airport : airports) {
            cout << "Airport Code: " << airport.second.airportCode << endl;
            cout << "   Airport Name: " << airport.second.airportName << endl;
            cout << "   Location: " << airport.second.location << endl;
            cout << "   Latitude: " << airport.second.latitude << endl;
            cout << "   Longitude: " << airport.second.longitude << endl;
            cout << endl;
        }
    }

    // Display flight graph (connectivity between airports)  -------- it show
the our directed graph
    void displayFlightGraph() const {
        cout << "Flight Graph (Directed):" << endl;
        for (const auto& entry : flightGraph) {
            cout << entry.first << " -> ";
            for (const auto& destination : entry.second) {
                cout << destination << " ";
            }
            cout << endl;
        }
    }

    // Display seat details for all flights
    void displaySeatDetailsForAllFlights() const {
        cout << "Seat Details for All Flights:" << endl;
        for (const auto& flight : flights) {
            flight.second.displaySeatDetails();
        }
    }

private:
    // this function to get airline name from flight number
    string getAirlineName(const string& flightNumber) const {
        if (flightNumber.substr(0, 2) == "6E") {
            return "Indigo";
        } else if (flightNumber.substr(0, 2) == "SG") {
            return "SpiceJet";
        } else if (flightNumber.substr(0, 2) == "AI") {
            return "Air India";
        } else {
            return "Indigo"; // Assign any unknown flights to Indigo
        }
    }
```

```
    // This is function to calculate distance between two airports using
Haversine formula
    double calculateDistance(const Airport& source, const Airport&
destination) const {
        // here we convert latitude and longitude from degrees to radians
        double lat1 = source.latitude * M_PI / 180.0;
        double lon1 = source.longitude * M_PI / 180.0;
        double lat2 = destination.latitude * M_PI / 180.0;
        double lon2 = destination.longitude * M_PI / 180.0;

        // Earth radius in kilometers
        const double R = 6371.0;

        // Haversine formula
        // The haversine formula is a very accurate way of computing distances
between
        // two points on the surface of a sphere using the latitude and
longitude of the two points

        double dlon = lon2 - lon1;  // distance of longitude
        double dlat = lat2 - lat1;  // distance of latitude
        double a = sin(dlat / 2) * sin(dlat / 2) + cos(lat1) * cos(lat2) *
sin(dlon / 2) * sin(dlon / 2);
        double c = 2 * atan2(sqrt(a), sqrt(1 - a));
        double distance = R * c;

        return distance;
    }
};

int main() {
    Airline airline;

    //Declare the airport
    //In input we gives 1.Airport code  2. name of airport  3. Location/city
4.langitude 5.latitude
    airline.addAirport("BOM", "Chhatrapati Shivaji Maharaj International
Airport", "Mumbai", 19.0896, 72.8656);
    airline.addAirport("BLR", "Kempegowda International Airport", "Bangalore",
13.1986, 77.7066);
    airline.addAirport("DEL", "Indira Gandhi International Airport", "New
Delhi", 28.5681, 77.1126);
    airline.addAirport("MAA", "Chennai International Airport", "Chennai",
12.9900, 80.1690);
    airline.addAirport("CCU", "Netaji Subhas Chandra Bose International
Airport", "Kolkata", 22.6522, 88.4466);
```

```
    airline.addAirport("HYD", "Rajiv Gandhi International Airport",
"Hyderabad", 17.2316, 78.4292);
    airline.addAirport("COK", "Cochin International Airport", "Kochi",
10.1520, 76.4017);
    airline.addAirport("PNQ", "Pune Airport", "Pune", 18.5822, 73.9197);
    airline.addAirport("GOI", "Goa International Airport", "Dabolim, Goa",
15.3801, 73.8319);
    airline.addAirport("AMD", "Sardar Vallabhbhai Patel International
Airport", "Ahmedabad", 23.0778, 72.6348);
    airline.addAirport("JAI", "Jaipur International Airport", "Jaipur",
26.8241, 75.8129);
    airline.addAirport("GAU", "Lokpriya Gopinath Bordoloi International
Airport", "Guwahati", 26.1060, 91.5859);
    airline.addAirport("TRV", "Trivandrum International Airport",
"Thiruvananthapuram", 8.4821, 76.9209);
    airline.addAirport("CCJ", "Calicut International Airport", "Kozhikode",
11.1395, 75.9553);
    airline.addAirport("BBI", "Biju Patnaik International Airport",
"Bhubaneswar", 20.2520, 85.8173);
    airline.addAirport("VTZ", "Visakhapatnam International Airport",
"Visakhapatnam", 17.7215, 83.2245);
    airline.addAirport("IXB", "Bagdogra Airport", "Bagdogra", 26.6841,
88.3247);
    airline.addAirport("IXC", "Chandigarh International Airport",
"Chandigarh", 30.6737, 76.7894);
    airline.addAirport("IDR", "Devi Ahilya Bai Holkar Airport", "Indore",
22.7215, 75.8011);
    airline.addAirport("PAT", "Jai Prakash Narayan International Airport",
"Patna", 25.5941, 85.0908);
    airline.addAirport("IXR", "Birsa Munda Airport", "Ranchi", 23.3171,
85.3215);


    // Type of aircraft use in India

    Aircraft aircraft1("Boeing 747", 150, 50, 20);
    Aircraft aircraft2("Airbus A320", 120, 30, 15);
    Aircraft aircraft3("Embraer E190", 100, 20, 10);
    Aircraft aircraft4("Bombardier Q400", 80, 15, 5);
    Aircraft aircraft5("ATR 72", 70, 10, 5);
    Aircraft aircraft6("Boeing 737", 130, 40, 15);



    // Available Flight
    // in this airlinr represent by this code
    // "SG"= SpiceJet  ------- "AI"=Air India ------- "6E" = Indigo  <-  else
other code also come Airline Indigo
    // then com Airport code and then destination airport come
```

```
    // then time and date of departure
    //  then which aircraft we used

    airline.addFlight("SG501", "BOM", "MAA", "03:30 PM", "22nd March 2024",
aircraft1);
    airline.addFlight("9W601", "CCU", "BLR", "04:00 PM", "23rd March 2024",
aircraft2);
    airline.addFlight("AI401", "DEL", "HYD", "05:30 PM", "24th March 2024",
aircraft3);
    airline.addFlight("6E501", "BOM", "GOI", "06:00 PM", "25th March 2024",
aircraft4);
    airline.addFlight("SG601", "MAA", "CCU", "07:30 PM", "26th March 2024",
aircraft5);
    airline.addFlight("AI501", "DEL", "COK", "08:00 PM", "27th March 2024",
aircraft6);
    airline.addFlight("6E601", "BOM", "PNQ", "09:30 PM", "28th March 2024",
aircraft1);
    airline.addFlight("SG701", "BLR", "GOI", "10:00 PM", "29th March 2024",
aircraft2);
    airline.addFlight("AI801", "DEL", "AMD", "11:30 PM", "30th March 2024",
aircraft3);
    airline.addFlight("6E901", "MAA", "JAI", "12:00 AM", "31st March 2024",
aircraft4);
    airline.addFlight("SG901", "BLR", "GAU", "01:30 AM", "1st April 2024",
aircraft5);
    airline.addFlight("AI1001", "BOM", "TRV", "02:00 AM", "2nd April 2024",
aircraft6);
    airline.addFlight("6E1101", "DEL", "CCJ", "03:30 AM", "3rd April 2024",
aircraft1);
    airline.addFlight("SG1101", "MAA", "BBI", "04:00 AM", "4th April 2024",
aircraft2);




    airline.displayFlightSchedule();
    airline.displayAirportDetails();
    airline.displayFlightGraph();

    // Displaying seat details for all flights
    airline.displaySeatDetailsForAllFlights();

    return 0;
}
```

Output of the code:

Output of the code:
Flight Schedule:
Airline: Indigo
Flight Number: 9W601
  Departure Airport: CCU
  Destination Airport: BLR
  Departure Time: 04:00 PM
  Date: 23rd March 2024
  Aircraft Model: Airbus A320
  Available Seats: 165

Airline: Air India
Flight Number: AI501
  Departure Airport: DEL
  Destination Airport: COK
  Departure Time: 08:00 PM
  Date: 27th March 2024
  Aircraft Model: Boeing 737
  Available Seats: 185

Airline: Indigo
Flight Number: 6E501
  Departure Airport: BOM
  Destination Airport: GOI
  Departure Time: 06:00 PM
  Date: 25th March 2024
  Aircraft Model: Bombardier Q400
  Available Seats: 100

Airline: SpiceJet
Flight Number: SG601
  Departure Airport: MAA
  Destination Airport: CCU
  Departure Time: 07:30 PM
  Date: 26th March 2024
  Aircraft Model: ATR 72
  Available Seats: 85

Airline: SpiceJet
Flight Number: SG701
  Departure Airport: BLR
  Destination Airport: GOI
  Departure Time: 10:00 PM
  Date: 29th March 2024
  Aircraft Model: Airbus A320
  Available Seats: 165

Airline: Indigo
Flight Number: 6E601
  Departure Airport: BOM
  Destination Airport: PNQ
  Departure Time: 09:30 PM
  Date: 28th March 2024
  Aircraft Model: Boeing 747
  Available Seats: 220

Airline: Air India
Flight Number: AI801
  Departure Airport: DEL
  Destination Airport: AMD
  Departure Time: 11:30 PM
  Date: 30th March 2024

Aircraft Model: Embraer E190
Available Seats: 130

Airline: Air India
Flight Number: AI401
Departure Airport: DEL
Destination Airport: HYD
Departure Time: 05:30 PM
Date: 24th March 2024
Aircraft Model: Embraer E190
Available Seats: 130

Airline: Indigo
Flight Number: 6E901
Departure Airport: MAA
Destination Airport: JAI
Departure Time: 12:00 AM
Date: 31st March 2024
Aircraft Model: Bombardier Q400
Available Seats: 100

Airline: Indigo
Flight Number: 6E1101
Departure Airport: DEL
Destination Airport: CCJ
Departure Time: 03:30 AM
Date: 3rd April 2024
Aircraft Model: Boeing 747
Available Seats: 220

Airline: SpiceJet
Flight Number: SG1101
Departure Airport: MAA
Destination Airport: BBI
Departure Time: 04:00 AM
Date: 4th April 2024
Aircraft Model: Airbus A320
Available Seats: 165

Airline: SpiceJet
Flight Number: SG901
Departure Airport: BLR
Destination Airport: GAU
Departure Time: 01:30 AM
Date: 1st April 2024
Aircraft Model: ATR 72
Available Seats: 85

Airline: SpiceJet
Flight Number: SG501
Departure Airport: BOM
Destination Airport: MAA
Departure Time: 03:30 PM
Date: 22nd March 2024
Aircraft Model: Boeing 747
Available Seats: 220

Airline: Air India
Flight Number: AI1001
Departure Airport: BOM
Destination Airport: TRV
Departure Time: 02:00 AM
Date: 2nd April 2024
Aircraft Model: Boeing 737
Available Seats: 185

Airport Details:
Airport Code: PAT
Airport Name: Jai Prakash Narayan International Airport
Location: Patna
Latitude: 25.5941
Longitude: 85.0908

Airport Code: IXC
Airport Name: Chandigarh International Airport

Location: Chandigarh
Latitude: 30.6737
Longitude: 76.7894

Airport Code: BBI
Airport Name: Biju Patnaik International Airport
Location: Bhubaneswar
Latitude: 20.252
Longitude: 85.8173

Airport Code: IXB
Airport Name: Bagdogra Airport
Location: Bagdogra
Latitude: 26.6841
Longitude: 88.3247

Airport Code: CCJ
Airport Name: Calicut International Airport
Location: Kozhikode
Latitude: 11.1395
Longitude: 75.9553

Airport Code: BOM
Airport Name: Chhatrapati Shivaji Maharaj International Airport
Location: Mumbai
Latitude: 19.0896
Longitude: 72.8656

Airport Code: BLR
Airport Name: Kempegowda International Airport
Location: Bangalore
Latitude: 13.1986
Longitude: 77.7066

Airport Code: CCU
Airport Name: Netaji Subhas Chandra Bose International Airport
Location: Kolkata
Latitude: 22.6522
Longitude: 88.4466

Airport Code: MAA
Airport Name: Chennai International Airport
Location: Chennai
Latitude: 12.99
Longitude: 80.169

Airport Code: AMD
Airport Name: Sardar Vallabhbhai Patel International Airport
Location: Ahmedabad
Latitude: 23.0778
Longitude: 72.6348

Airport Code: VTZ
Airport Name: Visakhapatnam International Airport
Location: Visakhapatnam
Latitude: 17.7215
Longitude: 83.2245

Airport Code: HYD
Airport Name: Rajiv Gandhi International Airport
Location: Hyderabad
Latitude: 17.2316
Longitude: 78.4292

Airport Code: PNQ
Airport Name: Pune Airport
Location: Pune
Latitude: 18.5822
Longitude: 73.9197

Airport Code: COK
Airport Name: Cochin International Airport
Location: Kochi
Latitude: 10.152
Longitude: 76.4017

Airport Code: IDR
  Airport Name: Devi Ahilya Bai Holkar Airport
  Location: Indore
  Latitude: 22.7215
  Longitude: 75.8011

Airport Code: JAI
  Airport Name: Jaipur International Airport
  Location: Jaipur
  Latitude: 26.8241
  Longitude: 75.8129

Airport Code: IXR
  Airport Name: Birsa Munda Airport
  Location: Ranchi
  Latitude: 23.3171
  Longitude: 85.3215

Airport Code: DEL
  Airport Name: Indira Gandhi International Airport
  Location: New Delhi
  Latitude: 28.5681
  Longitude: 77.1126

Airport Code: GOI
  Airport Name: Goa International Airport
  Location: Dabolim, Goa
  Latitude: 15.3801
  Longitude: 73.8319

Airport Code: GAU
  Airport Name: Lokpriya Gopinath Bordoloi International Airport
  Location: Guwahati
  Latitude: 26.106
  Longitude: 91.5859

Airport Code: TRV
  Airport Name: Trivandrum International Airport
  Location: Thiruvananthapuram
  Latitude: 8.4821
  Longitude: 76.9209

Flight Graph (Directed):
MAA -> CCU JAI BBI
DEL -> HYD COK AMD CCJ
BLR -> GOI GAU
CCU -> BLR
BOM -> MAA GOI PNQ TRV
Seat Details for All Flights:
Seat Details for 9W601:
  Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120
  Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30
  First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

Seat Details for AI501:
  Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111

E112 E113 E114 E115 E116 E117 E118 E119 E120 E121 E122 E123 E124 E125
E126 E127 E128 E129 E130
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30 B31 B32 B33
B34 B35 B36 B37 B38 B39 B40
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

Seat Details for 6E501:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15
First Class:
F1 F2 F3 F4 F5

Seat Details for SG601:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36
E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49 E50 E51 E52
E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65 E66 E67 E68
E69 E70
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10
First Class:
F1 F2 F3 F4 F5

Seat Details for SG701:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

Seat Details for 6E601:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120 E121 E122 E123 E124 E125
E126 E127 E128 E129 E130 E131 E132 E133 E134 E135 E136 E137 E138 E139
E140 E141 E142 E143 E144 E145 E146 E147 E148 E149 E150
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30 B31 B32 B33
B34 B35 B36 B37 B38 B39 B40 B41 B42 B43 B44 B45 B46 B47 B48 B49
B50
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18
F19 F20

Seat Details for AI801:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33

E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10

Seat Details for AI401:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10

Seat Details for 6E901:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15
First Class:
F1 F2 F3 F4 F5

Seat Details for 6E1101:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120 E121 E122 E123 E124 E125
E126 E127 E128 E129 E130 E131 E132 E133 E134 E135 E136 E137 E138 E139
E140 E141 E142 E143 E144 E145 E146 E147 E148 E149 E150
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30 B31 B32 B33
B34 B35 B36 B37 B38 B39 B40 B41 B42 B43 B44 B45 B46 B47 B48 B49
B50
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18
F19 F20

Seat Details for SG1101:
Economy Class:
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120
Business Class:
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30
First Class:
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

Seat Details for SG901:

Economy Class:

E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70

Business Class:

B1 B2 B3 B4 B5 B6 B7 B8 B9 B10

First Class:

F1 F2 F3 F4 F5

Seat Details for SG501:

Economy Class:

E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120 E121 E122 E123 E124 E125
E126 E127 E128 E129 E130 E131 E132 E133 E134 E135 E136 E137 E138 E139
E140 E141 E142 E143 E144 E145 E146 E147 E148 E149 E150

Business Class:

B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30 B31 B32 B33
B34 B35 B36 B37 B38 B39 B40 B41 B42 B43 B44 B45 B46 B47 B48 B49
B50

First Class:

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18
F19 F20

Seat Details for AI1001:

Economy Class:

E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17
E18 E19 E20 E21 E22 E23 E24 E25 E26 E27 E28 E29 E30 E31 E32 E33
E34 E35 E36 E37 E38 E39 E40 E41 E42 E43 E44 E45 E46 E47 E48 E49
E50 E51 E52 E53 E54 E55 E56 E57 E58 E59 E60 E61 E62 E63 E64 E65
E66 E67 E68 E69 E70 E71 E72 E73 E74 E75 E76 E77 E78 E79 E80 E81
E82 E83 E84 E85 E86 E87 E88 E89 E90 E91 E92 E93 E94 E95 E96 E97
E98 E99 E100 E101 E102 E103 E104 E105 E106 E107 E108 E109 E110 E111
E112 E113 E114 E115 E116 E117 E118 E119 E120 E121 E122 E123 E124 E125
E126 E127 E128 E129 E130

Business Class:

B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15 B16 B17
B18 B19 B20 B21 B22 B23 B24 B25 B26 B27 B28 B29 B30 B31 B32 B33
B34 B35 B36 B37 B38 B39 B40

First Class:

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15