# **Important Instructions:**

* Please read the document thoroughly before you code.
* Import the given skeleton code into your Visual Studio.
* Do not change the Skeleton code or the package structure, method names, variable names, return types, exception clauses, access specifiers etc.
* You can create any number of private methods inside the given class.
* You can test your code from main() method of the program

**Time: 2 hours**

**Assessment Coverage:**

* Classes and Objects
* ADO.NET

**Problem Statement: Booking Flight Ticket**

You are required to develop a C# CUI based application to book a flight Ticket and store it in DB.

**Please find below the table structures for reference:**

****

**Note:** The above table structure is for your reference. The database is already set for you to query in the platform.

**Requirement 1:** Create a public class with the following properties.

**Class Name: FlightDAL**

|  |  |
| --- | --- |
| **Method Signature** | **Description** |
|  |  |
| public static DataTable GetFlightDetails() | Create a Method called “GetFlightDetails” which returns the DataTable of Flight Details from the “FlightDetails” table of the Database. |
|  |  |
|  |  |
| public static bool ValidateFlightAvailability(int flightId, int noOfTickets) | Create another method “ValidateFlightAvailability” which accepts the flight id and the number of tickets which the user wants and checks the availability of the tickets from the database and returns a boolean value of “true” if the tickets are available. |
| public static int AddBooking(int passengerId, string passengerName, DateTime journeyDate,int noOfTickets, int flightId) | Create another method “AddBooking” which accepts the passenger id, passenger name, date of journey, number of tickets and the flight id and then calculates the total charges and finally stores the details in the “BookingDetails” table in the Database. |

Use **ADO.NET** data access technology to connect to a database, execute commands and retrieve data from the database.

Include a public property of string type and name it as **ConnectionString**.

The database connection string details is stored in the **App.config** file and the same should be returned by the ConnectionString property. The name of the connection string is “FlightBooking”.

**Requirement 2:** Create a public class with the following properties.

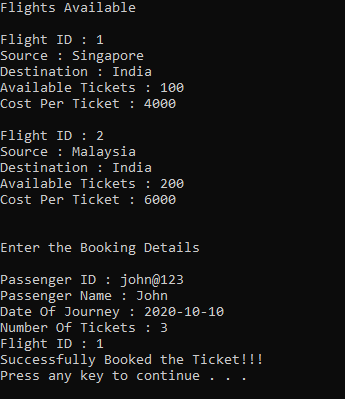
**Class Name: FlightBLL**

|  |  |
| --- | --- |
| **Method Signature** | **Description** |
|  |  |
| public static DataTable GetFlightDetails() | Create a Method called “GetFlightDetails” which returns the DataTable by invoking the FlightDAL’s GetFlightDetails method. |
|  |  |
|  |  |
|  |  |
| public static int AddBooking(string passengerId, string passengerName, DateTime journeyDate,int noOfTickets, int flightId) | Create another method “AddBooking” which accepts the passenger id, passenger name, date of journey, number of tickets and the flight id and passes the same to the AddBooking method of the FlightDAL class. |

**Requirement 3:** In the Main Program, Display the flight details using the “GetFlightDetails” method of the “FlightBLL” class. Get the following details from the User (passenger id, passenger name, date of journey, number of tickets, flight id) and calculate the total charges and insert the booking details into the database using the “FlightBLL” class’s “AddBooking” method.

The sample output is shown below:

**Sample I/O 1**



**Sample I/O 2**

