

Automatic evaluation[+]

Driver.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7 using System.Collections;
8 using System.Data;
9 using System.Configuration;
10
11 namespace TicketManagement //DO NOT change the namespace name
12 {
13     public class Program //DO NOT change the class name
14     {
15
16         static void Main(string[] args) //DO NOT change the 'Main' method signature
17         {
18             //Implement the code here
19             char choice = 'y';
20             Console.WriteLine("Enter Ticket Details: ");
21             while( choice == 'y')
22             {
23                 Console.WriteLine("Enter Passenger Id:");
24                 string id = Console.ReadLine();
25                 Console.WriteLine("Enter Passenger Name:");
26                 string name = Console.ReadLine();
27                 Console.WriteLine("Enter Travel Date:");
28                 string date = Console.ReadLine();
29                 Console.WriteLine("Enter Distance Travelled:");
30                 int dist = Convert.ToInt32(Console.ReadLine());
31                 DistanceValidator dv = new DistanceValidator();
32                 while ( dv.ValidateTravelDistance(dist) == "true")
33                 {
34                     Console.WriteLine("Given distance is invalid");
35                     Console.WriteLine("Enter Distance Travelled: ");
36                     dist = Convert.ToInt32(Console.ReadLine());
37                 }
38                 TicketDetail td = new TicketDetail(id, name, date, dist);
39                 TicketBooking tb = new TicketBooking();
40                 tb.CalculateCost(td);
41                 tb.AddTicket(td);
42                 Console.WriteLine(td.PassengerId);
43                 Console.WriteLine(td.PassengerName);
44                 Console.WriteLine(td.TravelDate);
45                 Console.WriteLine(td.DistanceTravel);
46                 Console.WriteLine($"Ticket Cost : {td.TicketCost}");
47                 Console.WriteLine("Book Another Ticket (y/n): ");
48                 choice =Convert.ToChar(Console.ReadLine());
49             }
50         }
51     }
52     public class DistanceValidator
53     { //DO NOT change the class name
54
55         public String ValidateTravelDistance(int distance) //DO NOT change the method signature
56         {
57             //Implement code here
58             if(distance < 0)
59             {
60                 return "Given distance is invalid";
61             }
62             else
63             {
```

```

64         return "";
65     }
66 }
67 }
68 }
69

```

App.config

```

1 <!-- THIS IS FOR REFERENCE ONLY. YOU ARE NOT REQUIRED TO MAKE ANY CHANGES HERE -->
2
3 <?xml version="1.0" encoding="utf-8" ?>
4 <configuration>
5     <startup>
6         <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
7     </startup>
8     <connectionStrings>
9         <add name="SqlCon"
connectionString="server=localhost;database=TicketBookingDB;uid=XXXXXX;password=XXXXXX;" />
10    </connectionStrings>
11 </configuration>

```

DBHandler.cs

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Data.SqlClient;
7 using System.Configuration;
8
9 namespace TicketManagement          //DO NOT change the namespace name
10
11 {
12     public class DBHandler          //DO NOT change the class name
13     {
14         //Implement the methods as per the description
15         public DBHandler() { }
16
17         public SqlConnection GetConnection()
18         {
19             return new SqlConnection(ConfigurationManager.ConnectionStrings["SqlCon"].ConnectionString);
20         }
21     }
22 }
23
24 }
25

```

TicketBooking.cs

```

1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using System.Data;
5 using System.Data.SqlClient;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9
10 namespace TicketManagement          //DO NOT change the namespace name
11 {
12     public class TicketBooking      //DO NOT change the class name
13     {
14         //Implement the property as per the description
15         public SqlConnection Sqlcon { get; set; }
16
17         public TicketBooking() { }
18         DBHandler d = new DBHandler();
19         public void AddTicket(TicketDetail detail)

```

```

20     {
21
22         string query = "INSERT INTO TicketBooking VALUES (@id, @name, @date, @dist, @cost)";
23         using (SqlConnection con = d.GetConnection())
24         using (SqlCommand cmd = new SqlCommand(query, con))
25         {
26             cmd.Parameters.Add("@id", SqlDbType.VarChar).Value = detail.PassengerId;
27             cmd.Parameters.Add("@name", SqlDbType.VarChar).Value = detail.PassengerName;
28             cmd.Parameters.Add("@date", SqlDbType.VarChar).Value = detail.TravelDate;
29             cmd.Parameters.Add("@dist", SqlDbType.Int).Value = detail.DistanceTravel;
30             cmd.Parameters.Add("@cost", SqlDbType.Float).Value = detail.TicketCost;
31             con.Open();
32
33             try
34             {
35                 cmd.ExecuteNonQuery();
36             }
37             catch (Exception e)
38             {
39                 Console.WriteLine(e.Message);
40             }
41             finally
42             {
43                 con.Close();
44             }
45         }
46     }
47
48     //Implement the methods as per the description
49     public void CalculateCost(TicketDetail detail)
50     {
51         if(detail.DistanceTravel <= 100)
52         {
53             detail.TicketCost = detail.DistanceTravel * 1;
54         }
55         else if(detail.DistanceTravel >100 && detail.DistanceTravel <= 300)
56         {
57             detail.TicketCost = detail.DistanceTravel * 1.5;
58         }
59         else if (detail.DistanceTravel > 300 && detail.DistanceTravel <= 500)
60         {
61             detail.TicketCost = detail.DistanceTravel * 2.5;
62         }
63         else if (detail.DistanceTravel > 500)
64         {
65             detail.TicketCost = detail.DistanceTravel * 4.5;
66         }
67     }
68
69 }
70 }
71

```

TicketDetail.cs

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace TicketManagement           //DO NOT change the namespace name
8  {
9      public class TicketDetail        //DO NOT change the class name
10     {
11         //Implement the fields and properties as per description
12
13         private string passengerId;

```

```

14     private string passengerName;
15     private string travelDate;
16     private int distanceTravel;
17     private double ticketCost;
18
19     public string PassengerId
20     {
21         get { return passengerId; }
22         set { this.passengerId = value; }
23     }
24     public string PassengerName
25     {
26         get { return passengerName; }
27         set { this.passengerName = value; }
28     }
29     public string TravelDate
30     {
31         get { return travelDate; }
32         set { this.travelDate = value; }
33     }
34     public int DistanceTravel
35     {
36         get { return distanceTravel; }
37         set { this.distanceTravel = value; }
38     }
39     public double TicketCost
40     {
41         get { return ticketCost; }
42         set { this.ticketCost = value; }
43     }
44
45
46     public TicketDetail() { }
47     public TicketDetail(string passengerId, string passengerName, string travelDate, int distanceTravel)
48     {
49         this.passengerId = passengerId;
50         this.passengerName = passengerName;
51         this.travelDate = travelDate;
52         this.distanceTravel = distanceTravel;
53     }
54 }
55 }
56
57 }
58

```

Grade

Reviewed on Friday, 7 January 2022, 7:32 PM by Automatic grade

Grade 100 / 100

Assessment report

[\[+\]](#)Grading and Feedback