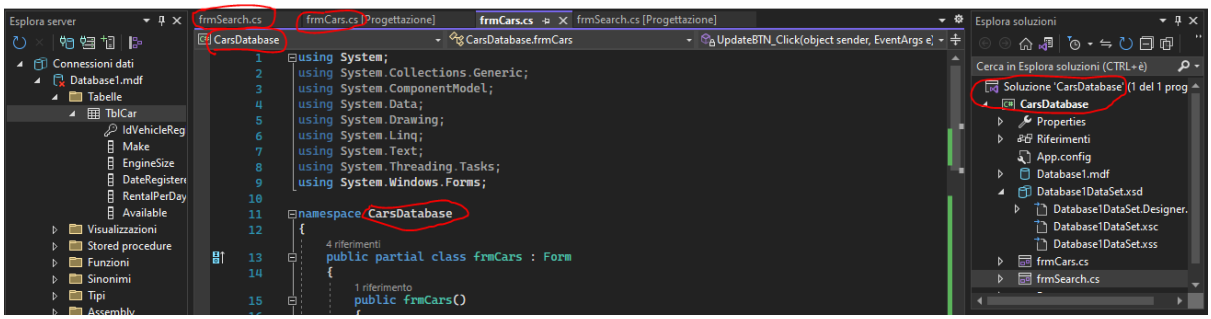
Assignment Template

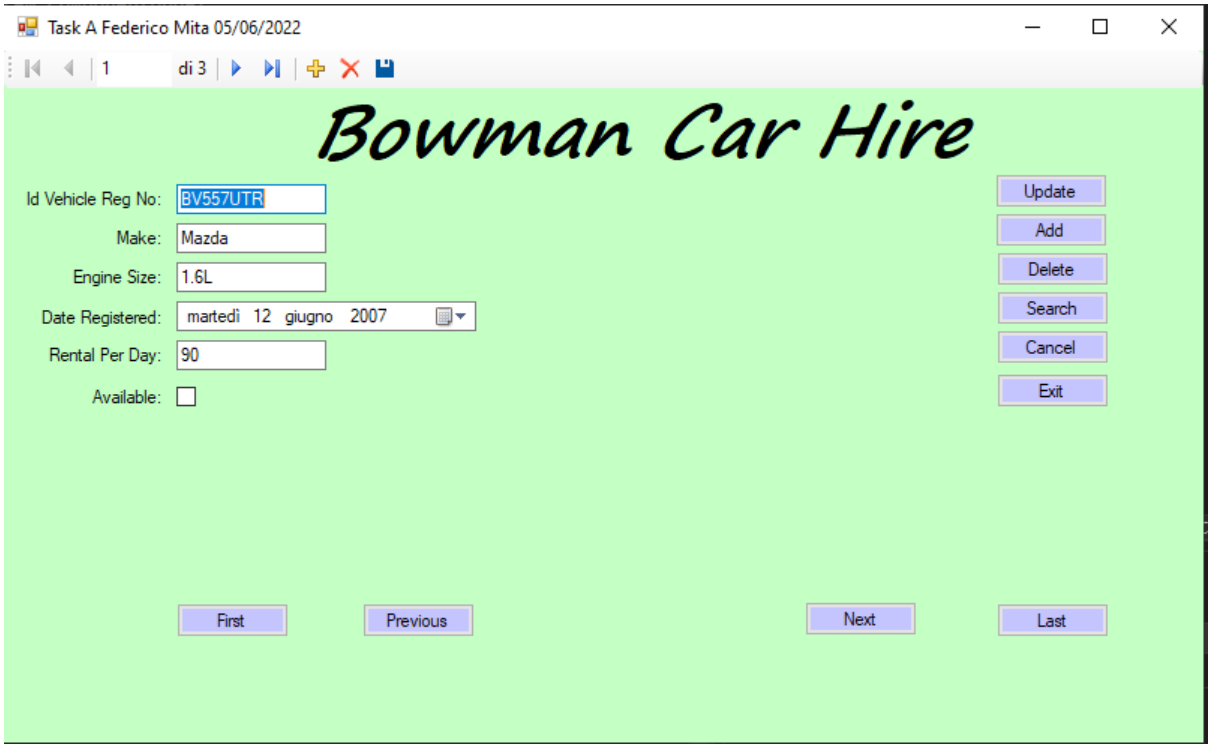
Task1: Take a screenshot of your IDE that clearly shows :

- The form saved as frmCars
- Project saved as CarsDatabse

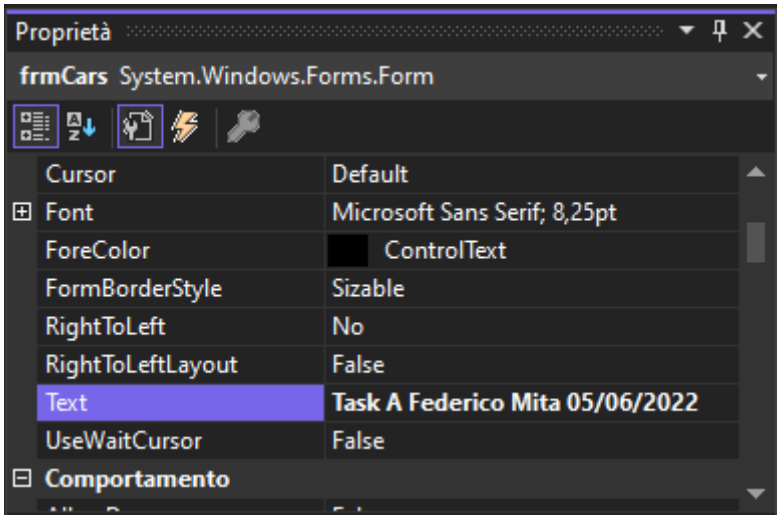


Task 2: Take a screenshot of your form running in your IDE. This must clearly show :

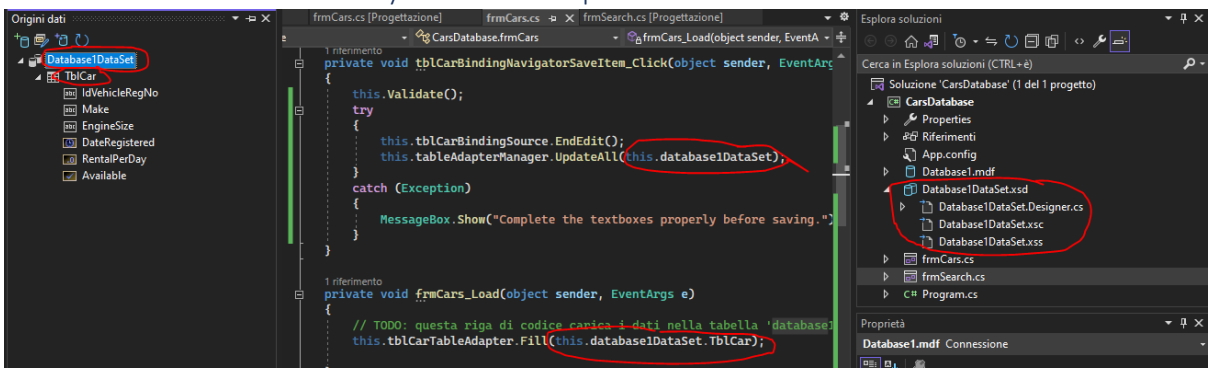
- A label for a heading in a different font, bold and large font size.
- Background set to a suitable colour
- Buttons for Add,Update,Delete ,Search,Cancel, Exit
- Controls to move to first ,previous , next and last record.
- Input and labels to display and create records.

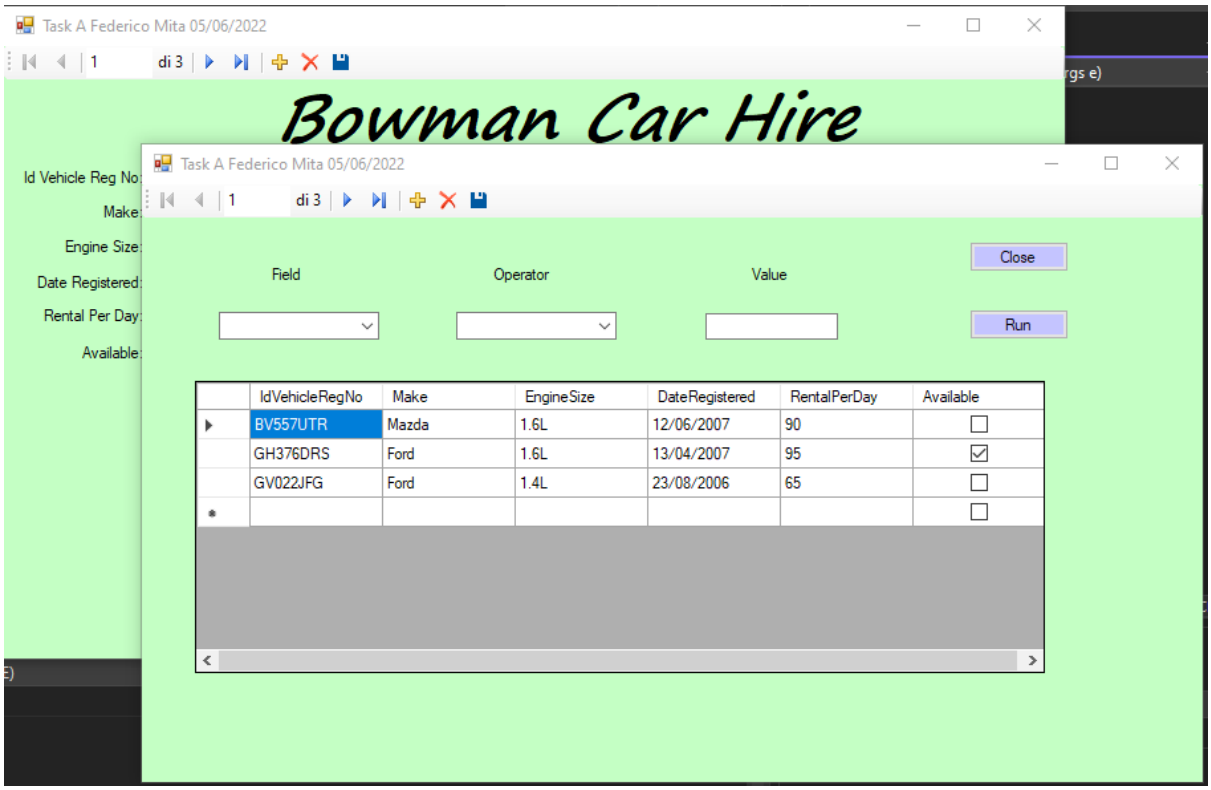


Task 3 : Take a screenshot of your form's text property set to "Your name and today's date".

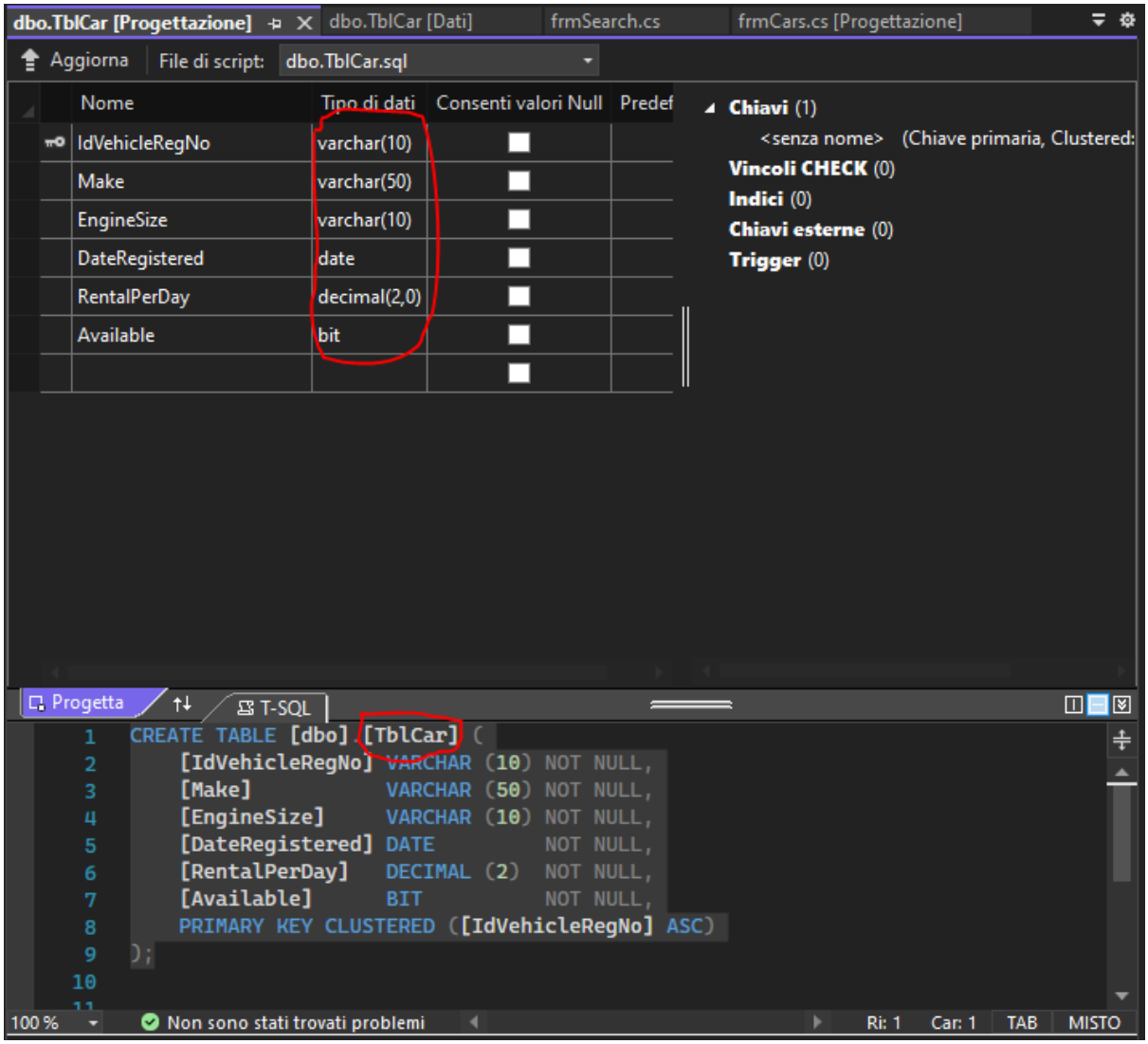


Task 4 : Take a screenshot of your form that proves it is connected to the database.





Task 5 : Take a screenshot of your database, its field and their datatypes and then take a picture showing that you have matched up the form input elements to those datatypes.



IdVehicleRegNo	Make	EngineSize	DateRegistered	RentalPerDay	Available
BV557UTR	Mazda	1.6L	12/06/2007	90	False
GH376DRS	Ford	1.6L	13/04/2007	95	True
GV022JFG	Ford	1.4L	23/08/2006	65	False
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Task 6: Take a picture of your code for the following buttons :

- Add button
- Update button
- Delete button

- Cancel button
- Code for controlling record movement
- Code for exit button
- Code for search button

```
System;
System.Collections.Generic;
System.ComponentModel;
System.Data;
System.Drawing;
System.Linq;
System.Text;
System.Threading.Tasks;
System.Windows.Forms;

pace CarsDatabase

riferimenti
ublic partial class frmCars : Form

    1 riferimento
    public frmCars()
    {
        InitializeComponent();
    }

    1 riferimento
    private void tblCarBindingNavigatorSaveItem_Click(object sender, EventArgs e)
    {
        this.Validate();
        try
        {
            this.tblCarBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.database1DataSet);
        }
        catch (Exception)
        {
            MessageBox.Show("Complete the textboxes properly before saving.");
        }
    }
```

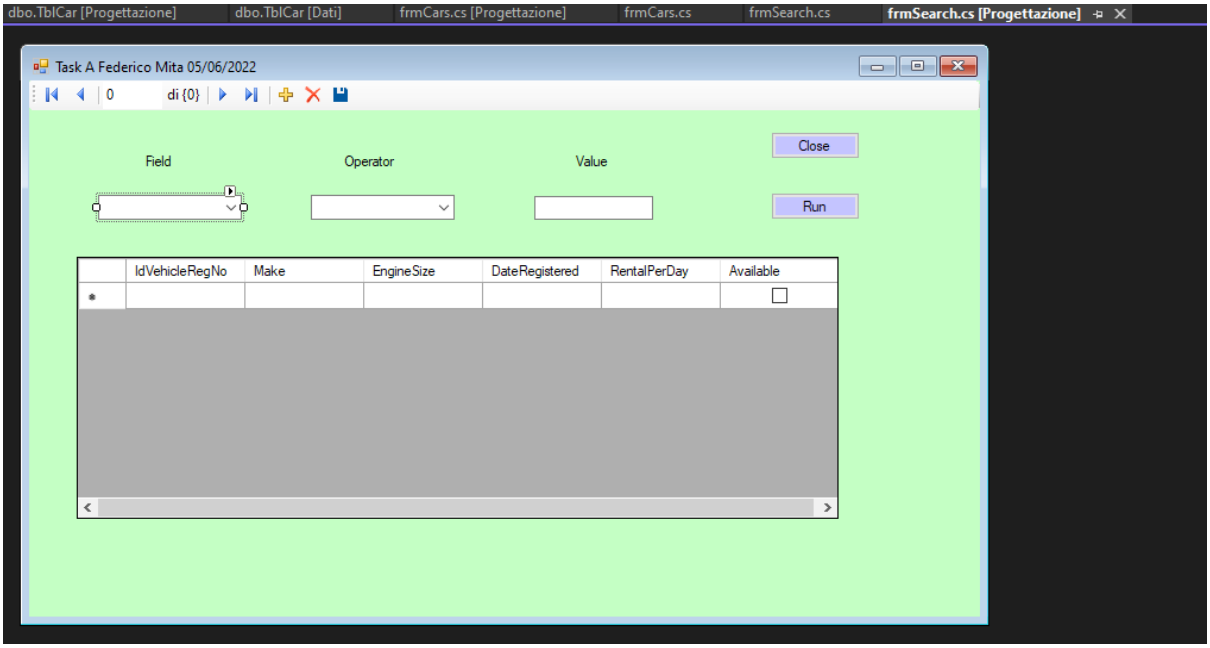
```
34      private void frmCars_Load(object sender, EventArgs e)
35      {
36          // TODO: questa riga di codice carica i dati nella tabella 'database1DataSet.TblCar'. È possibi
37          this.tblCarTableAdapter.Fill(this.database1DataSet.TblCar);
38
39      }
40
        1 riferimento
41      private void ExitBTN_Click(object sender, EventArgs e)
42      {
43          MessageBox.Show("You are quitting the program.");
44          Application.Exit();
45      }
46
        1 riferimento
47      private void FirstBTN_Click(object sender, EventArgs e)
48      {
49          try
50          {
51              bindingNavigatorMoveFirstItem.PerformClick();
52          }
53          catch (Exception)
54          {
55              MessageBox.Show("Complete the textboxes before changing page.");
56          }
57      }
58
        1 riferimento
59      private void PreviousBTN_Click(object sender, EventArgs e)
60      {
61          try
62          {
63              bindingNavigatorMovePreviousItem.PerformClick();
64          }
65          catch(Exception)
66          {
```

```
67              MessageBox.Show("Complete the textboxes before changing page.");
68          }
69      }
70
        1 riferimento
71      private void NextBTN_Click(object sender, EventArgs e)
72      {
73          //weirdly, clicking Next doesn't throw an error even with NULL fields, so it doesn't appear to need an exception
74          bindingNavigatorMoveNextItem.PerformClick();
75      }
76
        1 riferimento
77      private void LastBTN_Click(object sender, EventArgs e)
78      {
79          //weirdly, clicking Last doesn't throw an error even with NULL fields, so it doesn't appear to need an exception
80          bindingNavigatorMoveLastItem.PerformClick();
81      }
82
        1 riferimento
83      private void AddBTN_Click(object sender, EventArgs e)
84      {
85          try
86          {
87              bindingNavigatorAddNewItem.PerformClick();
88          }
89          catch(Exception)
90          {
91              MessageBox.Show("You have clicked the wrong button. Please click the button above to save.");
92          }
93      }
94
        1 riferimento
95      private void DeleteBTN_Click(object sender, EventArgs e)
96      {
97          bindingNavigatorDeleteItem.PerformClick();
98          label2.Text = "This record has been deleted.";
```

```
98          label2.Text = "This record has been deleted.";
99      }
100
        1 riferimento
101     private void UpdateBTN_Click(object sender, EventArgs e)
102     {
103         tblCarBindingNavigatorSaveItem.PerformClick();
104         label2.Text = "This record has been updated.";
105     }
106
        1 riferimento
107     private void SearchBTN_Click(object sender, EventArgs e)
108     {
109         frmSearch f2 = new frmSearch();
110         f2.Show();
111     }
112
        1 riferimento
113     private void CancelBTN_Click(object sender, EventArgs e)
114     {
115         idVehicleRegNoTextBox.Text = "";
116         makeTextBox.Text = "";
117         engineSizeTextBox.Text = "";
118         rentalPerDayTextBox.Text = "";
119
120         this.tblCarTableAdapter.Fill(this.database1DataSet.TblCar);
121
122     }
123
        1 riferimento
124     private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
125     {
126
127     }
128
129
130
```

Task 7: Take a image of your second form : frmSearch

- This should have 2 buttons – run and close
- Search textbox and it label
- A datagrid view to display the data
- The name of the form to display your name and today's date.



Task 8 : Create a Test plan and log created

Test plan (all the images are on a separate “Evidence Log” folder because of formatting problem with OpenOffice):

Functionality	Expected Result	Actual Result	Evidence in log
1(a to d). Navigate through the pages with “First”, “Previous”, “Next”, “Last” buttons.	Navigate through the logs.	Navigate through the logs. Pass.	Images from 1a to 1d.
2a. Using the button “Update” to save a filled log. 2b. Using the button “Update” to save an incomplete log. 2c. Using the button “Update to save an empty log.	2a. Saving the log with a confirmation label. 2b. Exception (all fields should be filled) or inaction. 2c. Exception (all fields should be filled) or inaction.	2a. Saving the log with a confirmation label. Pass. 2b. Saving the log with a confirmation label. Fail. 2c. Inaction. Pass.	2a. Images 2a1 and 2a2. 2b. Image 2b. 2c. Image 2c.
3a. Using the button “Add” with empty fields. 3b. Using the button “Add” with partially filled fields. 3c. Using the button “Add” with complete fields.	3a. Exception (all fields should be filled) or inaction. 3b. Exception (all fields should be filled) or inaction. 3c. Log added to TblCar.	3a. Inaction. Pass. However, all the buttons afterwords are unclickable and the program has to be forcefully closed from the IDE. 3b. Inaction. Pass. However, all the buttons afterwords are unclickable and the program has to be	3a. Image 3a. 3b. Image 3b. 3c. Images from 3c1 to 3c3.

		forcefully closed from the IDE. 3c. Inaction. Fields are deleted. Fail.	
4a. Using the button “Delete” to delete saved logs. 4b. Using the button “Delete” with incomplete fields. 4c. Using the button “Delete” with empty fields.	4a. The log should be deleted and confirmed through a label. 4b. Inaction. 4c. Inaction.	4a. The log is deleted and confirmed through a label. Pass. 4b. Inaction. Pass. However, all the buttons afterwords are unclickable and the program has to be forcefully closed from the IDE. 4c. Inaction. Pass. However, all the buttons afterwords are unclickable and the program has to be forcefully closed from the IDE.	4a. Images 4a1 and 4a2. 4b. Image 4b. 4c. Image 4c.
5. Using the button “Search” to open the frmSearch Form.	5. Opening the frmSearch Form.	5. The Form frmSearch is opened. Pass.	5. Image 5.
6a. Using the button “Cancel” to empty fields of a saved log. 6b. Using the button “Cancel” to empty fields of a modified log or return previous data. 6c. Using the button “Cancel” to empty fields already empty	6a. Inaction. 6b. The modified log's fields are emptied or previous data is returned. 6c. Inaction.	6a. Inaction. Pass. 6b. Previous data is returned. Pass. 6c. Inaction. Pass. However, all the buttons afterwords are unclickable and the program has to be forcefully closed from the IDE.	6a. Image 6a. 6b. Images 6b1 and 6b2. 6c. Image 6c.
7. Using the button “Exit” to close the program.	7. Closing the program after a message informing the user.	7. The program is closed after a message informing the user. Pass.	7. Image 7.
8. Using the button “Close” to close the frmSearch Form.	8. Closing the frmSearch Form.	8. Form frmSearch closed. Pass.	8. Impossible to provide an image.
9a. Using the button “Run” with empty fields. 9b. Using the button “Run” with partially completed fields. 9c. Using the button “Run” with completed fields (referring to	9a. Exception (all fields should be filled) or inaction. 9b. Exception (all fields should be filled) or inaction. 9c. It should search logs related to the selected data.	9a. Exception (all fields should be filled). Pass. 9b. Exception (all fields should be filled). Pass. 9c. It searches the logs related to the selected data. Pass.	9a. Image 9a. 9b. Image 9b. 9c. Image 9c. 9d. Image 9d.

existing data). 9d. Using the button “Run” with completed fields (referring to inexistent data).	9d. No logs found or exception (inexistent data).	9d. No logs found. Pass.	
10. Confirming there are the right operators for every “Field” data chosen.	10. All operators logically belonging to a specific “Field” data should be available and eventual illogical additional operator should generate an exception message.	10. All operators logically belonging to a specific “Field” data are available and all operators for “Date” (minus =) generate and exception message. Pass.	10. Images from 10.1 to 10.14.

Task 9 : Write a technical document that explains the purpose of the software and how you connect the code to the database.

CarsDatabase – A “Bowman Car Hire” Software

The CarsDatabase software serves the function of collecting all the data of the automobiles registered to Bowman Car Hire, a car-for-hire company. Through the software, it's possible to add, delete and modify any log, but it's also possible to search through the connected database sorting through the data using a chosen field.

Features

The software is basically divided into two available forms/windows. The first, dubbed frmCars, has the possibility of sorting through the already saved car logs (through the buttons “First”, “Previous”, “Next” and “Last”), but also the following:

- Text boxes to add or edit content.
- Buttons to Add or Update content (registration number, make, engine size, data registered and available rental per day).
- Buttons to Delete logs and to Cancel what has been written, getting back to the previous data.
- A Search function, which leads to the frmSearch form/window (more about this below).
- Finally, an Exit button permits to close the program.

The second form, as anticipated above, is called frmSearch and contains search tools to sort through the cars' logs saved to the “Bowman Car Hire”'s database.

- Three boxes let us choose from the aforementioned types of content and adding certain values, to filter through the data.
- A Run button executes the previous passage.

- A Close button closes this form (but not the program).

All over the various ways and combinations possible with this program, there are several exception messages that might pop up. These advise on a correct course of action without crashing the software.

Connecting the Forms to a Database

This is possible, in Visual Studio, by clicking View->Other Window->Data Sources. From there, we can choose a Data Source Type (in this case a Database), the Dataset and finally we put or select the name of our Database. Once selected, we can choose a table from it to be connected to our forms.

Task 10 : Create a program listing. This is the complete listing of a computer program, and includes source code, and all files(Database and forms) that make up the software program.

https://github.com/FedericoMita/Coding/tree/main/Bootcamps%20%26%20Online%20Courses/TDM/C%23%20tests%20and%20assignments/CarsDatabase