Cario

Definition

Cario is car system program designed to serve car exhibition and garages owners . Using our car system program you can add , filter , edit and delete cars whenever you want . Our program elegant with the adding pictures feature , you can add pictures to the car you want so you can recognize the car by just seeing its picture , color selection is one of the features that we are proud of .

In this report I am going to demonstrate how our program works , what obstacles we faced and how we managed to solve them properly .

Design

We put an effort in choosing the appropriate colors that can give the user the best possible experience , our prevalent color was gray , gray was the color that had the honor to be chosen as background color for our program , red was the color we used for distinguishing errors so the user get noticed if an input has been entered in a wrong way .
Our program has responsive design , it is neat whatever its situation is , maximized , minimized or whatever , it is always neat and can maintain its beautiful design.

Insertion

For adding a car in your database , you are supposed to input all the values of the attributes of the car ; model , horsepower , max speed , color and the image of the car .
The input is tested before any operation is done to it , we make sure that your input will be suitable to be inserted in its vessel .
Every input has its corresponding private data member in a class called car . We faced a problem by entering the color of the car into its private data member , so we

had to think in a way that can make it possible , after a lot of brainstorming , the way we chose was segmentation the color into its basic components ; blue , red , green and obesity . We used two bitwise operations called shifting right and left , using these operations, we could assemble the segments into an integer variable by shifting the values of the segment :

int icol = (col.Color.A << 24) | (col.Color.R << 16) | (col.Color.G << 8) | col.Color.B;

We have shifted the first segment 24 digit to the left , as well as the second component we shifted it 16 digits to the left ... etc , Then we did an OR operation to all components and zipped them into one single variable , we stored this variable in a private data member in our class .
For uploading image , after the user has chosen the picture he/she wants we save its address in an invisible textbox then we took the text and save it in a private data member in our car class .
Our program has the ability to read the data in the textboxes and write it into a file so you can save your input data and retrieve them whenever you want using the read from file button , which copies the data in the file and puts it in the input fields , so you can continue your insertion .
To add car into your collection , you must press “Add” button which adds your car to the database after checking that every input is suitable to be inserted (valid) , if not then an error message pops up to the user letting him/her know that he/she must validate his/her input data .

Presentation

After you have inserted your cars successfully , now you can browse them by pressing the "Show available cars" button . By moving to the next page , you can see a table containing the existed cars , every car has its unique ID , you can browse them and you can update (edit) any property you want in any record you want , remove queries is one of the available features , filtering the shown cars feature helps you find cars depending on specific attribute(s) that you choose yourself .

How are the queries shown ?

After we have inserted every car property in the database , we can now present the database info into our DataGridViewer .

But here comes the obstacles ;

Presenting ID , car model , max speed and price is not a hard task to do , due to that all of them is a text input (string) , which is fine , but this is not the case for pictures and colors .

For presenting a picture we had to break it down to its basic components (blue , red , green , obesity) , if you can recall , we have assembled them into one integer variable , now it is the time to disassemble this variable so we can extract every single component alone , then we use these components to build a color , then set this color as a background color for a cell belonging to the corresponding car .

The pictures had a special treat! , can you recall when we saved their address ? We used this address to access the picture , then saves it in a file in our project , so that our pictures are existed in the project whoever the user is and whatever the host device is !

After storing them locally , we use the address that is in the database to access these pictures and show them correctly in their suitable cells .

Conclusion

Facing those obstacles was really a challenge I will never forget about !

From choosing the idea , to dividing roles , to searching for the best methods and apply them ! It was a long enjoyable journey !

- Issam Abdoh .
- Raneem Halawani .
- Raghad Abd-Alhadi
- Yazan Akkad