EAST WEST UNIVERSITY

CSE103: Structured Programming
[Fall 2022]

Project Report
Vehicle Management System

Course Code	: CSE103
Course Title	: Structured Programming
Section	: 12
Group Number	: 3

Submitted by:

Student ID	Student Name	Contribution Percentage
2022-3-60-135	Manoshi Ghosh Pushpita	5%
2022-3-60-142	Aysha Ferdous Anika	15%
2022-3-60-123	M. Nura Alam Naim	80%

Introduction: A vehicle management system is a software program that helps individuals and organizations manage and track vehicles within their fleet. It allows users to input and store information about vehicles, including the vehicle identification number (ID), the vehicle's make and model, the manufacturer's name, and the manufacturing date. The system also typically allows users to delete, search for, and update vehicle details as needed.

For the security purpose, there is a system to access this program by entering username and password only. The user can also change the username and password if it is necessary.

Overall, a vehicle management system can help organizations streamline their vehicle operations and improve efficiency by providing a central location for all vehicle-related information and tasks. It can also help reduce the risk of errors and improve the accuracy of vehicle data.

The functions used in this program are given below with a description:

2.1 main():

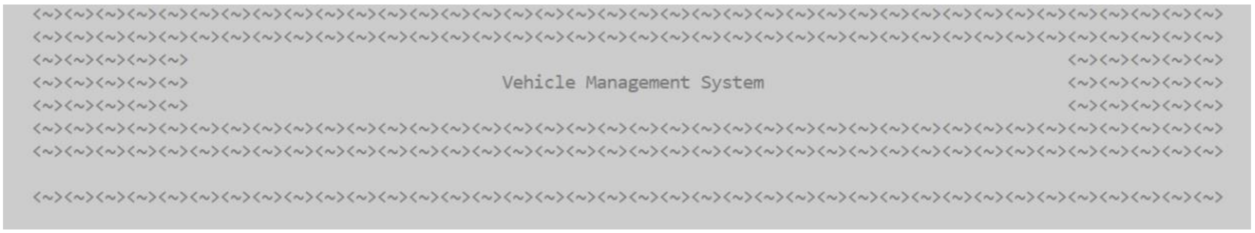
Description: The main function is the base of every function. The first interface is also in the main function. It consists of a switch function to call other functions according to the choice.

The menu function is called inside a do-while loop in the main function. When the menu function returns a value, the switch statement takes the value and the works are done inside the case of the switch according to the choice. There are 6 cases and a default case. If the chosen number is not between 1-6, the default case will print a message and take the input again.

2.2 head_Message():

Description: This is a common interface function. It has no parameter and its return type is void. It shows only the “Vehicle Management System” in the middle of the additional design.

Output:

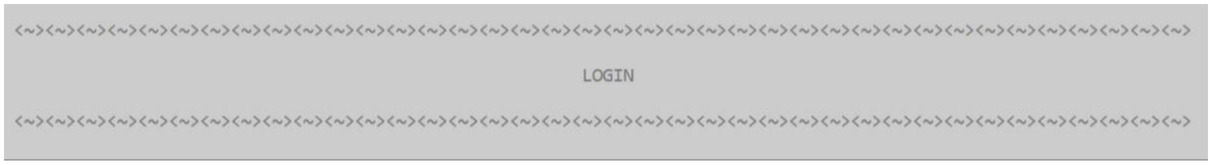


```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~>                              Vehicle Management System                             <~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
```

2.3 print_Message_in_Center():

Description: This is also a common interface function. It takes a character pointer as a parameter and returns nothing. This function is called when the name of the task is needed to show in the middle of the screen. When this function is called by passing a string, this function prints the string in the center of the line with an additional design.
Functionality: This function divides the passed string into 3 portions. First, it prints half of the string except the center index of the string on the left of the center point of the line. Then print the other half without the center index to the right. In the end, it prints the center index in the center of the line.

Output:

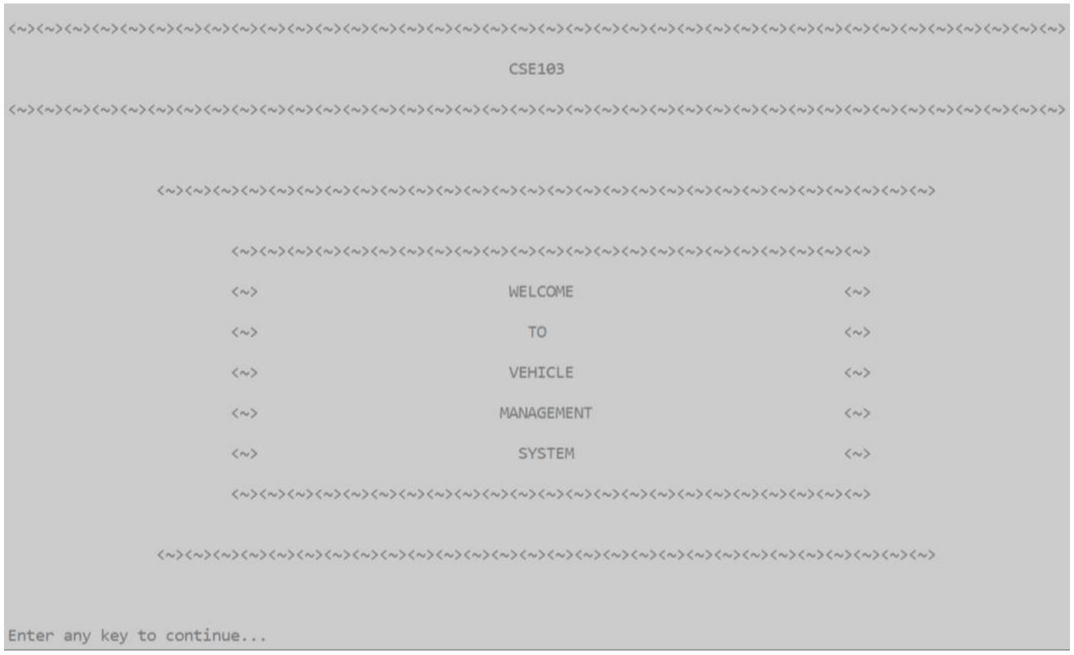


```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                            LOGIN

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
```

2.4 welcome_Message():

Description: The **print_Message_in_Center()** is called by passing “CSE103” as an argument. This function displays “WELCOME TO VEHICLE MANAGEMENT SYSTEM” in the middle of the screen with an additional design. This interface has no parameters and returns nothing. This function is used only once on the first page of the program.

Output:



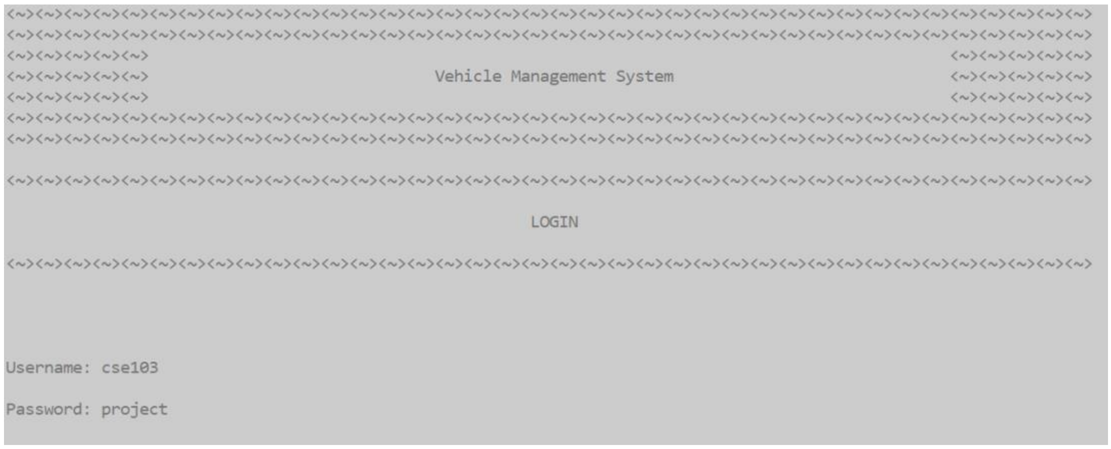
```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                      CSE103

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>


                <~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                        <~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
                        <~>                           WELCOME                             <~>
                        <~>                             TO                                <~>
                        <~>                           VEHICLE                             <~>
                        <~>                          MANAGEMENT                           <~>
                        <~>                            SYSTEM                             <~>
                        <~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                <~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>


Enter any key to continue...
```

2.5 log():

Description: This function consists **head_Message()**. Here **print_Message_in_Center()** is called by passing “LOGIN” as the argument. It will be called in the main function inside a do-while loop. It does not have any parameters and it returns an integer value after the work. This function also consists of the default username and password in crede.user and crede.pass.

Functionality: This function takes username and password from the user as input and compares them using the **strcpy()** function with the default password and username. If the password and username match, it returns integer ‘1’ which breaks the do-while loop and takes the user to the menu. If the comparison results are false, the function returns ‘0’ which leads the user to input username and password again. This process is continuous because of the do-while loop, until the correct username and password, is being entered.

Output:



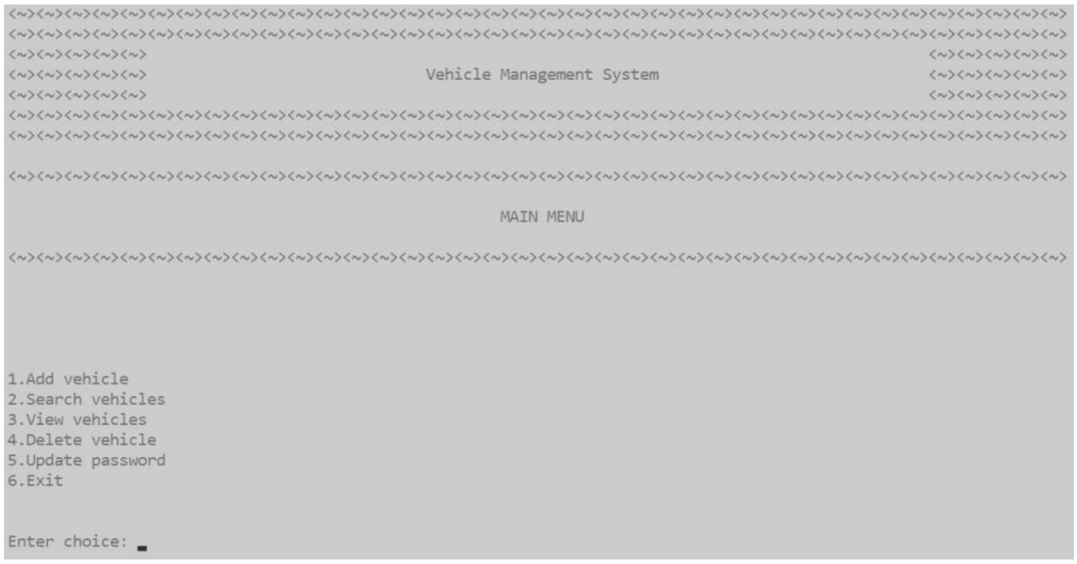
```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                     <~><~><~><~><~>
<~><~><~><~><~>                              Vehicle Management System                              <~><~><~><~><~>
<~><~><~><~><~>                                                                                     <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                       LOGIN

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>



Username: cse103

Password: project
```

2.6 menu():

Description: This function consists **head_Message()**. Here **print_Message_in_Center()** is called by passing “MAIN MENU” as an argument. This contains the list of tasks that can be done by the program. It takes nothing as a parameter but returns an integer value.

Functionality: It asks the user to select an option. This function will take the choice as a string and place the sting in a character-type array called temp. Then converts the entered number into an integer value by using the **atoi()** library function. The function works like this method because the **scanf()** is incompatible with the **gets()** function.

Output:



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~>                            Vehicle Management System                               <~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                      MAIN MENU

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>





1.Add vehicle
2.Search vehicles
3.View vehicles
4.Delete vehicle
5.Update password
6.Exit


Enter choice: _
```

2.7 addVehicleInDataBase():

Description: This function is called when the user enters choice 1 in the menu. It takes vehicle details from the user such as Vehicle ID NO, Vehicle Name, Vehicle Manufacturer Name, and Vehicle issued to date by the Manufacturer. It sends this input into different functions and checks if they are correct or not. If these details are correct then this function stores the details.

Functionality:
This function first clears the screen and then the **head_message()** function is called. After that, the **print_Message_in_Center()** function is called by passing "ADD NEW VEHICLE" as the argument.
For loop: Every work will be done inside a for-loop. The loop's initial value is the count which has been declared in the main function. After entering every detail of the vehicle, the increment will happen. After that user will be asked to press enter and then the loop will end.

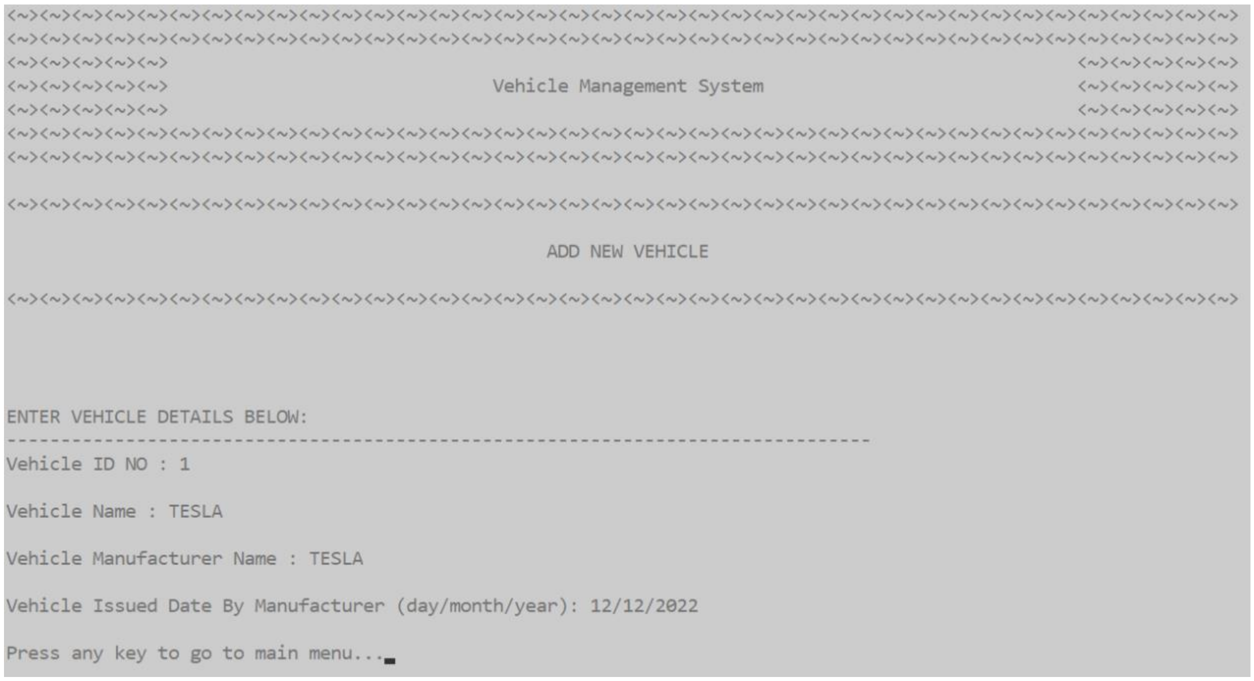
Id: This works inside a do-while loop. Entered id number will be kept inside a character array called temp as a string. Then the string will be passed to **the s_id_valid()** function to check the id is unique and consists of only digits. If the function returns a true value, the loop will end and the user will be proceeding to the next task. If the function returns false, the message "Invalid! Characters are used or the same id is given. Try again" will be printed and the loop will be continued until the valid id is entered.

Vehicle Name: This works inside a do-while loop. Entered vehicle name will be kept in the vehicle_name structure element. Then that will be copied to a character array called temp as a string. Then the string will be passed the **is_name_valid()** function to check whether the name is valid or not. If the function returns a true value, the loop will end and the user will be proceeding to the next task. If the function returns false, the message "Invalid Name. Please try again" will be printed and the loop will be continued until the valid id is entered.

Vehicle Manufacturer Name: This also works inside a do-while loop. Entered vehicle name will be kept in the manu_name structure element. Then that will be copied to a character array called temp as a string. Then the string will be passed to the **is_name_valid()** function to check whether the name is valid or not. If the function returns a true value, the loop will end and the user will be proceeding to next the task. If the function returns false, the message "Invalid Name. Please try again" will be printed and the loop will be continued until the valid id is entered.

Manufacturer date: This also works inside a do-while loop. It takes the date, month, and year inside the d,m, and y variables by using the **scanf()** library function. Without using the **scanf()**, the input data will take an extra line which will not be compatible with the interface. This process will be producing an extra buffer. To skip that buffer, the **gets()** function will place the buffer inside the temp character array. Then this function will send the inputted d, m, and y in the **is_valid_date()** function as an argument. If the **is_date_valid()** function returns ‘1’ after checking the dates are valid, the do-while loop will end otherwise the function will take the date again until the valid date is entered. After taking the valid date, the function will place the values inside the day structure elements accordingly to dd, mm and yyyy. This process is continuous until the valid date is entered.

After all the entries, the function will place the incremented loop control variable value inside the count variable and will return the count value to the main function.



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                      <~><~><~><~><~>
<~><~><~><~><~>                           Vehicle Management System                                  <~><~><~><~><~>
<~><~><~><~><~>                                                                                      <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                 ADD NEW VEHICLE

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>



ENTER VEHICLE DETAILS BELOW:
--------------------------------------------------------------------------------
Vehicle ID NO : 1

Vehicle Name : TESLA

Vehicle Manufacturer Name : TESLA

Vehicle Issued Date By Manufacturer (day/month/year): 12/12/2022

Press any key to go to main menu...
```

2.8 is_Name_Valid():

Description: This function checks if the vehicle name and vehicle manufacturer name are valid or not. Its parameter is a character pointer and the return type is an integer.

Functionality: First it checks the length of the passed string. If the string's length is more than 30 characters long, the result of the function will be '0'. Then it checks every string element if there is anything except alphabets, digits, space, underscore (_), and dash (-). If it finds anything except them, the result will be '0'. If everything is fine the result will be '1'. In the end, the function will return the result.

2.9 is_id_Valid():

Description: This function checks if the vehicle id no is unique and it has only digits or not. This function takes and character pointer and an integer count as parameters. It also returns the result as an integer.

Functionality: This function checks every string index if there are any digits or not. If it finds a character, it will consider the result as '0' otherwise '1'. Then it searches in the everybody structure index if there is the same value in the id element. If it finds any similar value in any other index, it will consider the result as '0' otherwise '1'. In the end, the function will return the result.

2.10 is_Valid_Date():

Description: This function has 3 integer parameters and its return type is an integer. It checks the inputted integer under some conditions. It checks if the combined numbers are valid or not.

Functionality: Firstly, it checks the year. If the inputted year is between 1900 to 2022, it will give the result ‘1’ otherwise ‘0’. Then it checks the month if it is between 1 to 12. If it is true, it will give the result ‘1’ otherwise ‘0’. Then checks the day. If the date is between 1 to 31 when the month is 1,3,5,7,8,10 or 12, the result will be ‘1’ otherwise ‘0’. If the date is between 1 to 30 when the month is 4,6,9 or 11 the result will be ‘1’ otherwise ‘0’. For month 2, the day must be between 1 to 28. But, for the leap year, the day can be 29. So the function checks the year when the month is 2 if the year is a leap year or not. After all the checks, the function returns an integer as the result. If any check result is false, the function with ‘l’ returns ‘0’. To do so, the or gate is used.

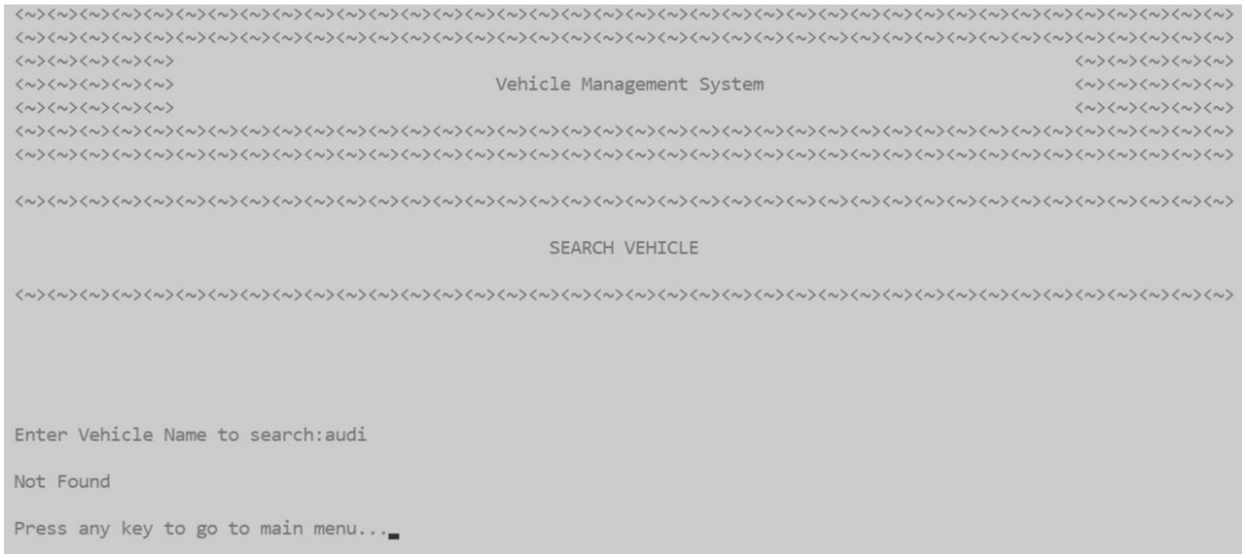
2.11 searchVehicles():

Description: When the user enters Choice 2, this function will be called. This function takes the input of the vehicle name from the user. Then this function searches the vehicle within the stored vehicle details. If the function finds a match it displays all the details of the searched vehicle and if it doesn’t find a match it simply displays not found. The count number is passed to this function as the parameter.

Functionality: This function first clears the screen and then the **head_message()** function is called. After that the **print_Message_in_Center()** function is called by passing “SEARCH VEHICLE” as the argument. Then, this function takes the vehicle name from the user and places the inputted string in temp character array. By using for loop, it searches every structure index to find where the name is. If it finds the index, it sends the index number to the **display()** to print the details of entered vehicle name. Also, it assigns ‘1’ to ‘found’ to use it. If it does not find the name of the vehicle in the structure index, found will be false and this condition will print “Not Found” in the display.

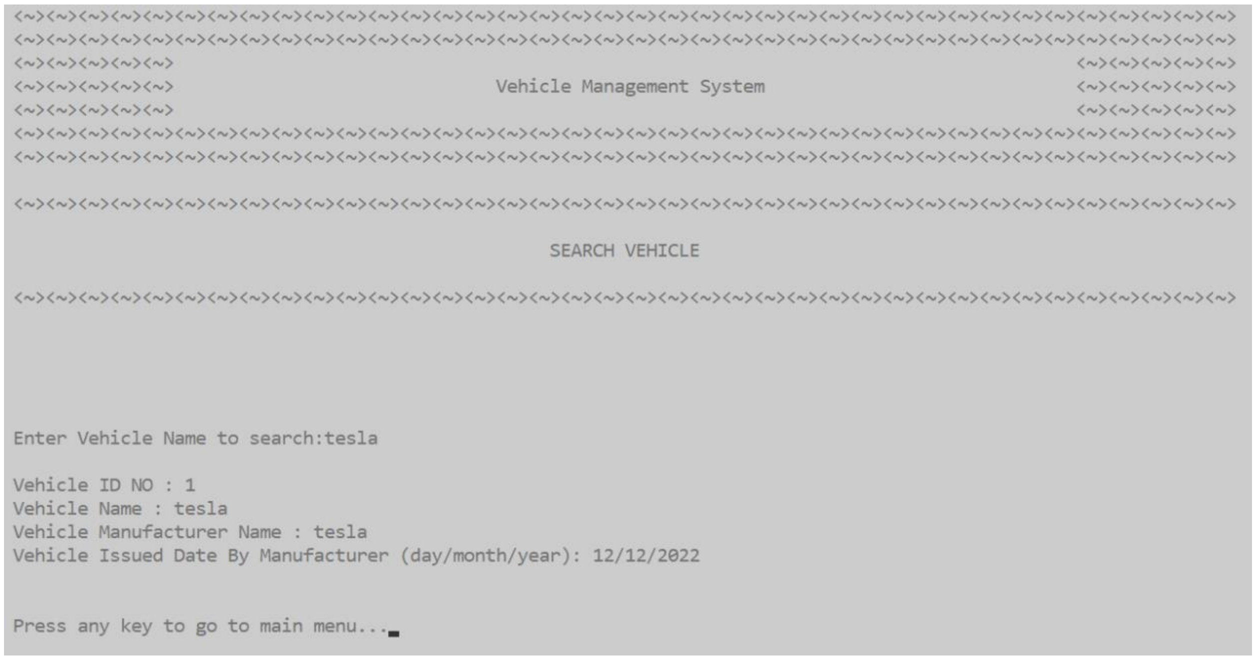
Output:

If seached vehicle is not found



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~>                              Vehicle Management System                             <~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                  SEARCH VEHICLE

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>




Enter Vehicle Name to search:audi

Not Found

Press any key to go to main menu...
```

If searched vehicle is found



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~>                              Vehicle Management System                             <~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                  SEARCH VEHICLE

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>




Enter Vehicle Name to search:tesla

Vehicle ID NO : 1
Vehicle Name : tesla
Vehicle Manufacturer Name : tesla
Vehicle Issued Date By Manufacturer (day/month/year): 12/12/2022


Press any key to go to main menu...
```

2.12 display():

Description: This function is called in the **searchVehicle()** function. It displays the vehicle details on the screen. Its return type is void and it has an integer number as a parameter.

Functionality: When the structure index is passed from the **searchVehicle()** to this function, this function prints the elements of the **body** and the **date** structure.

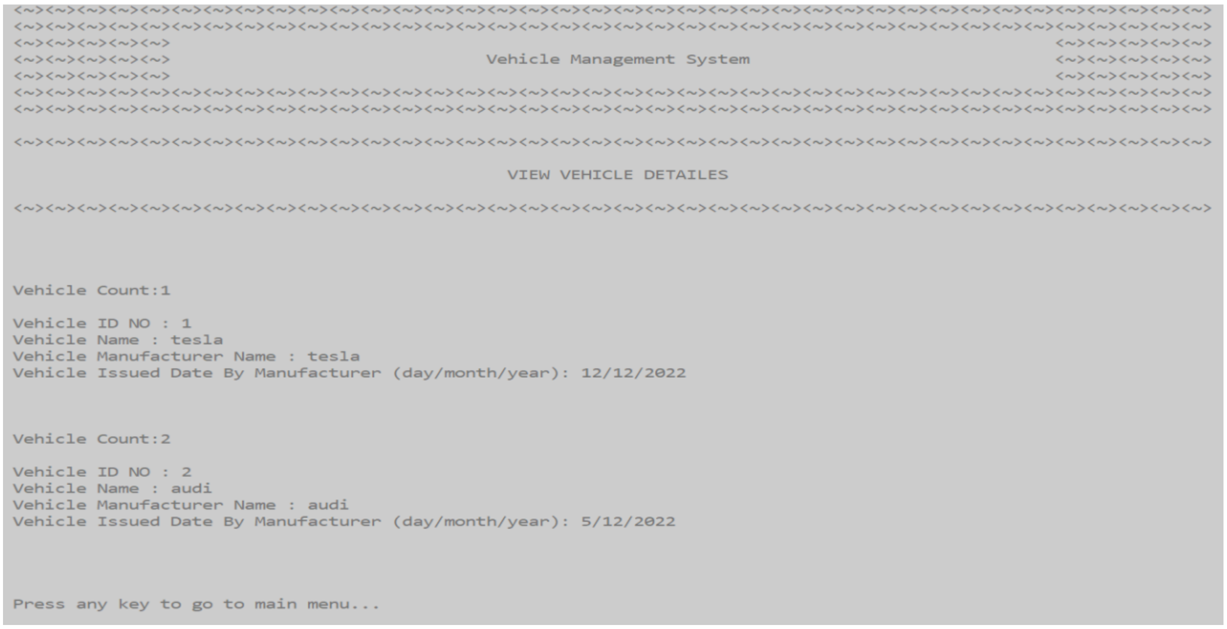
2.13 viewVehicles():

Description: This function is called when the user enters choice 3. It displays all the details of the vehicles that have been added. If there are no vehicles have been added, it will display "No Records". The vehicle count is needed to pass in the function to work properly. It is a void return type function.

Functionality: This function first clears the screen and then the **head_message()** function is called. After that, the **print_Message_in_Center()** function is called by passing "SEARCH VEHICLE" as the argument. For the beginning of the function when no vehicle has been entered, the function will display "No Records". When the counter value is more than zero, This can print details of every vehicle. A for-loop will start from 0 and end when it reached the count value. That's how every structure element will be accessed and printed by this function. In the end, the function will ask the user to press any key to go to the main menu.

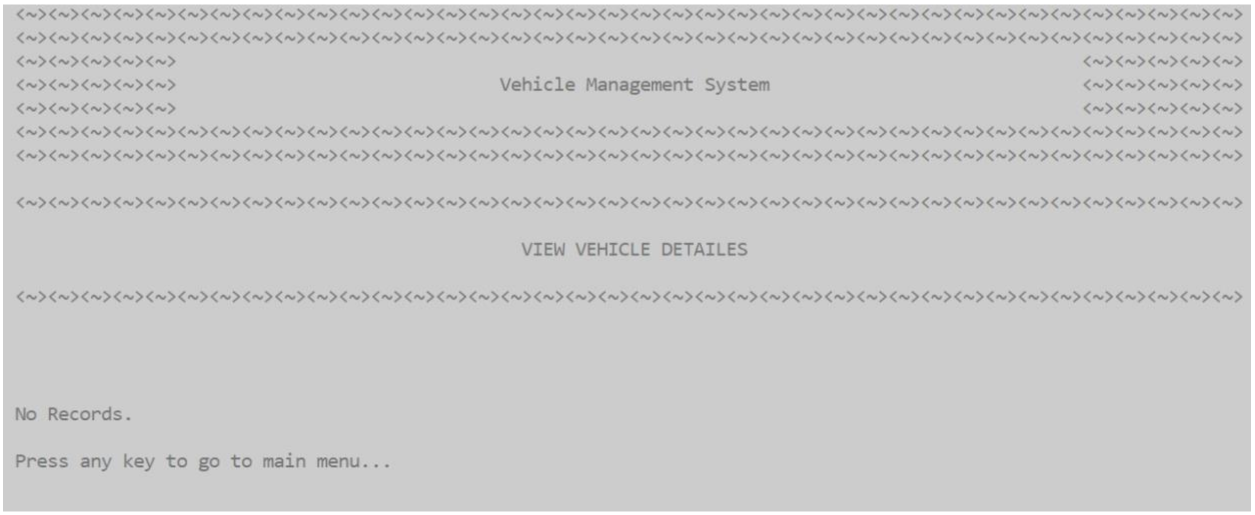
Output:

Stored vehicle details



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                          <~><~><~><~><~>
<~><~><~><~><~>                                 Vehicle Management System                                <~><~><~><~><~>
<~><~><~><~><~>                                                                                          <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                    VIEW VEHICLE DETAILES

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>




Vehicle Count:1

Vehicle ID NO : 1
Vehicle Name : tesla
Vehicle Manufacturer Name : tesla
Vehicle Issued Date By Manufacturer (day/month/year): 12/12/2022


Vehicle Count:2

Vehicle ID NO : 2
Vehicle Name : audi
Vehicle Manufacturer Name : audi
Vehicle Issued Date By Manufacturer (day/month/year): 5/12/2022



Press any key to go to main menu...
```

If there are no vehicles stored



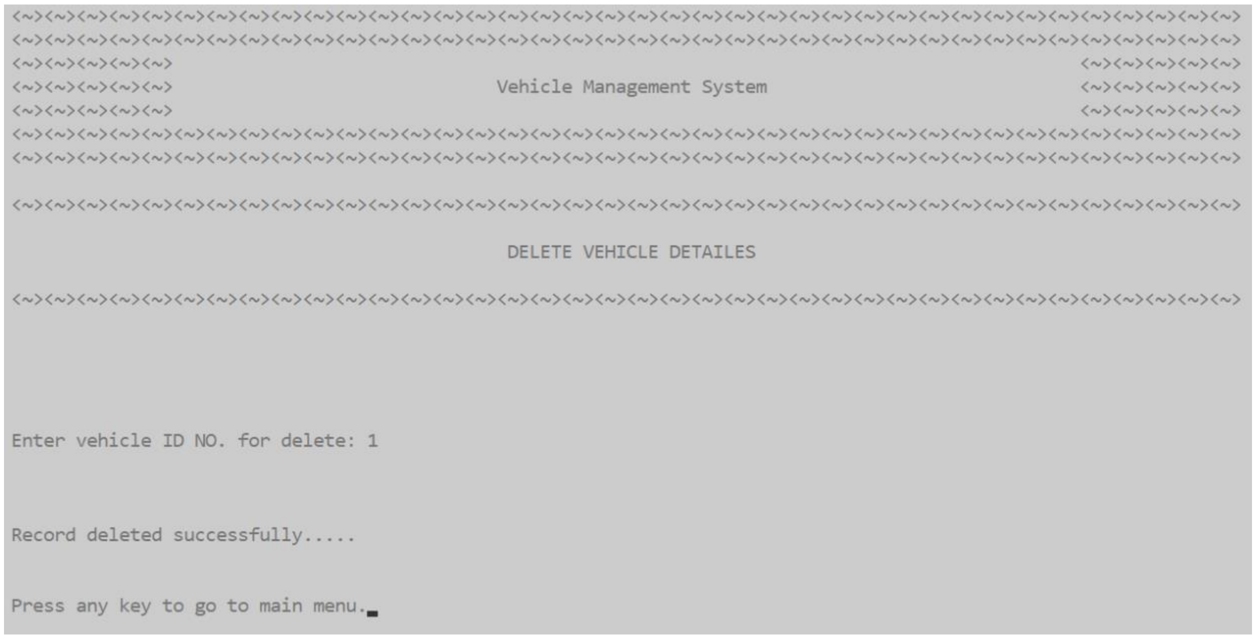
```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                          <~><~><~><~><~>
<~><~><~><~><~>                                Vehicle Management System                                 <~><~><~><~><~>
<~><~><~><~><~>                                                                                          <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                                  VIEW VEHICLE DETAILES

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>



No Records.

Press any key to go to main menu...
```

2.14 deleteVehicles():

Description: This function is called when the user enters choice 4. It asks the user which vehicle he/she wants to delete and takes the vehicle id number from the user. Then delete the details of the vehicle. This needs the counter number to work and it returns an integer.

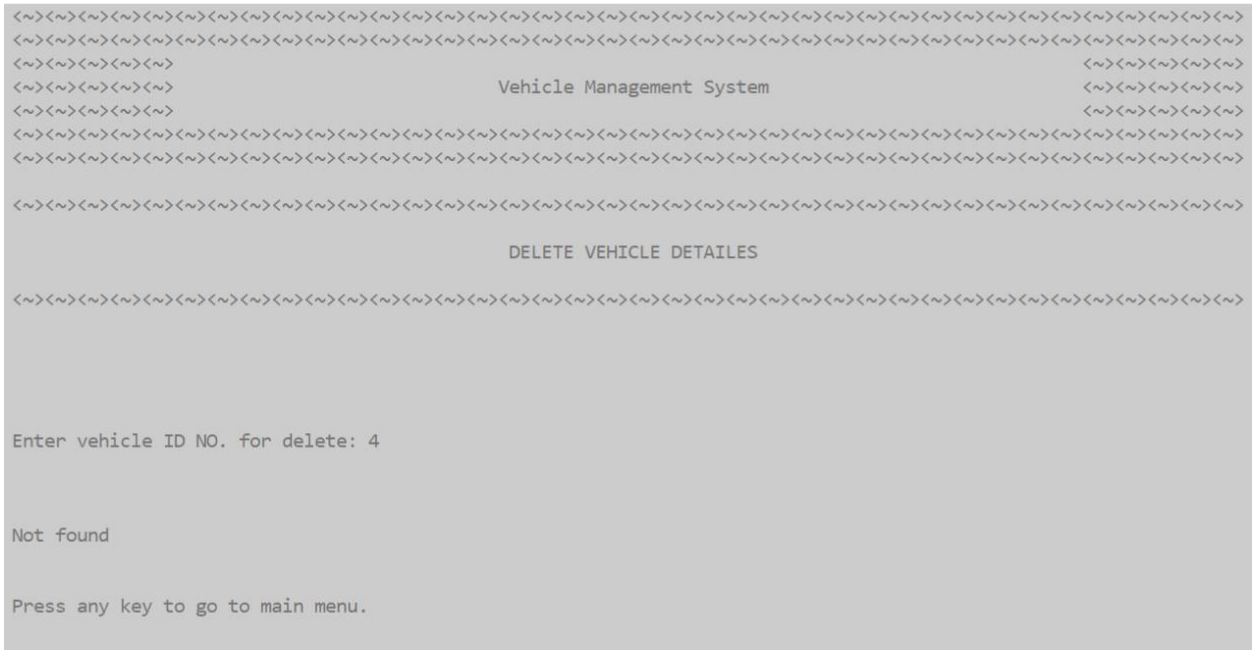
Functionality: This function first clears the screen and then the **head_message()** function is called. After that, the **print_Message_in_Center()** function is called by passing “DELETE VEHICLE DETAILS” as the argument. Then it asks the user to enter the vehicle id number the user wants to delete. The function takes the input as a string and places that inside a character array called temp. Then the function will search in which index the id number is located by using a for loop. In every loop, it compares the temp string with the id element of every body structure. When the index is found, the del variable will hold the index number of the body and the date structure. Then another loop will start from the del and that will end at the nearest count index. This loop will copy the next elements to the targeted index from the next index. This process will work for every index from the targeted index to the last index. After this process, the count number will be decreased. If the id number does not match with any index element, a “Not Found” message will be displayed. In the end, the program will tell the user to press any key to go to the main menu.

If the vehicle is found



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~>                              Vehicle Management System                             <~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                              DELETE VEHICLE DETAILES

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>





Enter vehicle ID NO. for delete: 1



Record deleted successfully.....


Press any key to go to main menu.
```

If vehicle is not found



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~>                              Vehicle Management System                             <~><~><~><~><~>
<~><~><~><~><~>                                                                                    <~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                              DELETE VEHICLE DETAILES

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>





Enter vehicle ID NO. for delete: 4



Not found


Press any key to go to main menu.
```

2.15 updateCredential():

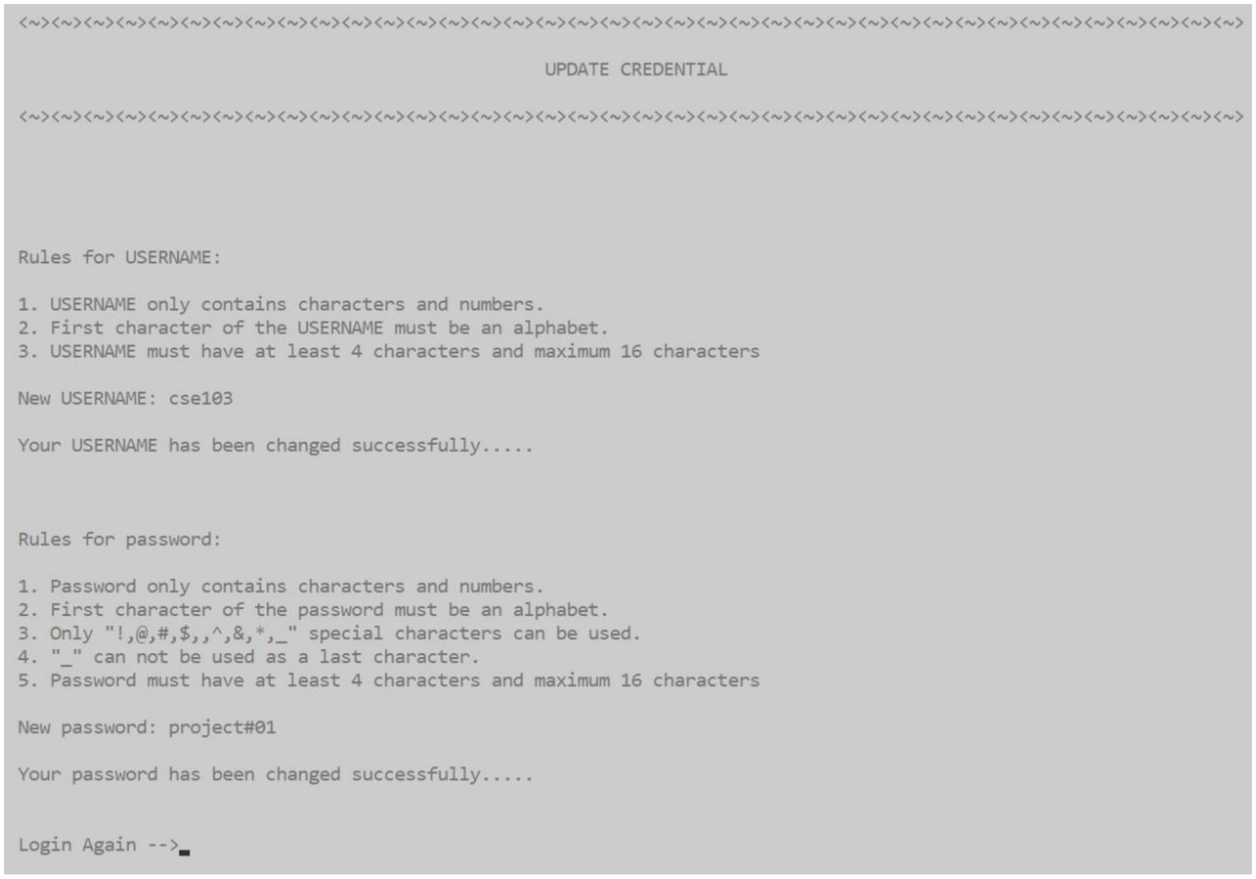
Description: When the user enters choice 5, this function is called. It asks the user and takes a input for new username and password. Then function checks whether the conditions for username and password are fulfilled or not. If the conditions are fulfilled it changes the previous password and username, then displays your password has been changed successfully. Then it asks the user to log in again.

Functionality: This function first clears the screen and then the **head_message()** function is called. After that, the **print_Message_in_Center()** function is called by passing “UPDATE CREDENTIAL” as the argument. For the username, the function takes the input as a string and places it into the crede structure called user. Then the function checks if the length of the username is between 4 to 16 characters or not, the first character consists only of alphabets or not. If the username consists outside anyone the results will be false. Then inside a for loop, every string index will be checked if there is anything except alphabets and digits. The for loop will end at the last index of the array. If it consists, the result will be false. If the result is false, a "Wrong Keyword Have Entered! Try again" message will be shown. This whole process will be inside a do-while loop so that this will work accordingly until the user enters a valid username. And at the end, a “Your USERNAME has been changed successfully.....” message will be shown.

After the username, the function works in the same manner as the username. It places the password to pass element of the crede structure. Then checks it consists of at least 4 elements and the highest 25 elements. But, for the password, the user can add “!,@,#,$,%,^,&,*,_ “ in the password. If the entered password is invalid, the same message will be shown. And this will work inside a do-while loop in order to work until the valid password is entered. And at the end, a “Your Password has been changed successfully.....” message will be shown.

After setting up the new username and password user need to log in again to further use this function.

Output:



```
<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>

                                       UPDATE CREDENTIAL

<~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~><~>




Rules for USERNAME:

1. USERNAME only contains characters and numbers.
2. First character of the USERNAME must be an alphabet.
3. USERNAME must have at least 4 characters and maximum 16 characters

New USERNAME: cse103

Your USERNAME has been changed successfully.....



Rules for password:

1. Password only contains characters and numbers.
2. First character of the password must be an alphabet.
3. Only "!,@,#,$,,^,&,*,_" special characters can be used.
4. "_" can not be used as a last character.
5. Password must have at least 4 characters and maximum 16 characters

New password: project#01

Your password has been changed successfully.....


Login Again -->
```

Here are some possible remarks about the functions of a vehicle management system:

- The ability to input and store vehicle information is a key function of the system, as it allows users to track and organize their vehicles in a centralized location.
- The ability to delete, search for, and update vehicle details can help users keep their data up-to-date and accurate.
- Generating reports can provide valuable insights into the performance and usage of vehicles within the fleet, which can be used to make informed decisions about vehicle management.
- Integration with other business tools can further streamline operations and improve efficiency by providing a single location for all vehicle-related tasks.

Overall, the functions of a vehicle management system can greatly improve the efficiency and effectiveness of vehicle operations for organizations of all sizes.

Personal comments:

"I'm really happy with how the program turned out. It was a lot of work, but it was worth it."

-M. Nura Alam Naim

Challenges: Creating this program was quite challenging. We had to find every possible part of the program to make it more user-friendly.

The first challenge we faced when was in the message in center function. That was not easy until we worked hard on it.

Secondly, we were facing problems while working with the add vehicle function. There were so many things to check and the writing was huge. So we made each function to check whether each entry is valid or not. Mostly the date validity check was very difficult. We thought and took help to complete the task.

Then the delete function was very critical. We had to think a lot to make the function usable.

Update credential function was easy for us because we have solved this problem before in websites.

Creating this program would be easier if we could use global variables. Then we didn’t have to pass the vehicle count to the function again and again. We could access the count from any function.

Opportunities: While creating the program we have learned a lot of things. We had to use the library function so many times. So, we learned how and when to use the functions properly. For example, we have to be careful while using the scanf() function with the gets() function. We have learned How to pause the functions’ work until pressing any key by the getche() function. We have learned how to pass a string or array to another function by using a pointer. Also, how to pass and get the value of the same variable. How to set a password and update the password in the function. There are so many things we have learned by creating this program.