**EXSM 3935: JavaScript Fundamentals Assignment**

Deadline: Sunday Oct 17, 2021 at 11:59 PM

Introduction

In this assignment, you will put what you’ve learned over the entire course to create a car registration system which takes in basic information about a vehicle and its owner, checks over all fields to ensure the correct data type is provided, and then stores the information in an array. Please be sure to check the criteria outlined below to ensure all requirements for this assignment are met.

**If your program does not run and display at least some output, you will receive a mark of zero.**

Requirements

The menu should be persistent (the choices are repeated until a valid choice is made and the program should only terminate when the user specifically selects that choice) and offer the following choices:

1. Create New Car Profile
2. View Previous Profiles
3. Quit Application

Create New Car Profile: When this choice is selected, the user should be prompted to enter the following information:

1. Owner First Name
   1. First name inputs longer than 20 characters should display a descriptive error message and prompt the user to try again
   2. First names must only contain alphabetic characters, the only acceptable symbol character is a dash (“-”). A descriptive error message should be displayed to the user and the user should be prompted to try again, if this condition is breached.
2. Owner Last Name
   1. See requirements for owner first name.
3. Car Owner Correspondence Address
   1. Not empty
4. Date Purchase
   1. Date of purchase should be entered in the format YYYY-MM-DD, should this condition fail, the user should receive a descriptive error message and be prompted to try again.
   2. All 3 portions of the date should be valid (no day above 31 and no month above 12, checking for month-specific day counts is optional). Should this condition fail, the user should receive a descriptive error message and be prompted to try again.
5. Car Brand
   1. Create a list of major car brands, if user input does not match a major car brand, the user should receive a descriptive error message and be prompted to try again. The input should be case insensitive.
6. Car Model
   1. Not empty
7. Car Year
   1. Must be a valid year between 1990 and the present year plus one.
8. Car VIN Number
   1. VIN must be exactly 17 characters long, and only consist of numeric and alphabetical characters
   2. All alphabetical characters should be stored in a capitalized form

After all fields are filled, the program should store the information in an array.

General Conditions:

1. All fields are mandatory.
2. User error should not cause loss of prior information (if a user enters an incorrect date, they should not have to go back to the beginning of the form).
3. There should be error handling for all input fields, with informative error messages.
4. Any leading or trailing whitespace should be removed from inputs.

View Previous Profiles: When this choice is selected, the user should be able to see the previous 3 profiles made over the session and then immediately return to the main menu. The previous profiles need not be persistent (after a user terminates the program and restarts, this option should display nothing).

Quit Application: The program should print “Goodbye!” and then terminate.

Criteria

| Requirement | Marks Available | Mark(s) Awarded | Comments |
| --- | --- | --- | --- |
| Menu is persistent until the user chooses to terminate the program. | 1 |  |  |
| Create New Car Profile: |  |  |  |
| Owner first name and last name fields are validated to specified conditions. | 2 |  |  |
| Car owner correspondence address field is validated to specified conditions. | 0.5 |  |  |
| Date of purchase field is validated to specified conditions. | 4 |  |  |
| Car brand field is validated to specified conditions. | 1 |  |  |
| Car model field is validated to specified conditions. | 0.5 |  |  |
| Car year field is validated to specified conditions. | 1 |  |  |
| Car VIN number field is validated to specified conditions. | 2 |  |  |
| All general conditions are met. | 4 |  |  |
| View Previous Profiles: |  |  |  |
| Last three previous profiles entered during the session are displayed. (5 marks for implementing a 2D array for information storage, 3 marks for any other method) | 5 |  |  |
| Deductions: |  |  |  |
| README.md file is not present, or does not contain a reasonable minimum amount of information. | -2 |  |  |
| Variable, constant and/or method names do not meet [the code style requirements](https://docs.google.com/document/d/12iEtYtYiKwRAjhLayI-JVNVqXkud3X_PtOE9D0sU9Cw/edit) (-1 each declaration). | -5 |  |  |
| Statements are duplicated across the beginning and/or end of all decision branches (at least one occurrence). | -1 |  |  |
| Missing semicolons (-1 each). | -3 |  |  |
| Unhandled exception(s) generated (-1 each scenario that generates one). | -3 |  |  |

**Total: /21**