

IMPORTANT SUBMISSION INSTRUCTIONS

- **Submit your assignment early and submit any improvements you have made often.** No excuses for the server being down or your computer crashing or the Martians attacked... will be accepted.
- **Your assignment must be submitted to the eLearning system using the *Submit Assignment 2* link in one zip file called **assignment2_xx.zip** where xx stands for the version number of your assignment. Use the version number to help **you** keep track of **your** submissions.**
- If you are submitting multiple versions of your assignment, only the one with the **most recent date** will be marked regardless of version number.
- **DO NOT EMAIL YOUR ASSIGNMENT** to either the instructor or the teaching assistant. **It will not be considered as submitted** if you do, you will receive zero for the assignment if it is not placed on the E-Learning system by the deadline.

Plagiarism

- Students are reminded that plagiarism is a serious offence. **Students must submit a signed soft copy of the plagiarism declaration or receive zero for the assignment.**
- Simply cutting and pasting code found on the web, attributing it to the author and then submitting it as a completed assignment will not benefit you. It must be your idea. **Do yourself a favour... do your own work!**

Late Submissions

- You are allowed late submissions for up to four days after the due date. Late submissions will incur penalties as shown below. Penalties are cumulative, so for example, an assignment that is two days late will receive a 15% penalty.

Days Late	% Penalty
1	5
2	10
3	15
4	20

Assignment Details

This is the second part of the assignment where you are to use HTML5, CSS, JavaScript and PHP to create a **responsive** Web application for a government Vehicle Licensing and Registration Management System (VLRMS). This enables legal drivers to register and log into the system so that they can update their registration and insurance information if necessary and then access the main control panel. The functional requirements that must be met for this assignment are described below.

1. Figure 1 shows the Account Registration page that you created in the first assignment. Notice that there are two new fields in this page: Email and Telephone. The HTML file for

Figure 1: The driver registration page of the VLRMS

this page is named **registration.html** and is used by a driver to register for an online account with the VLRMS. This HTML page is used by the PHP script **registration.php**. It is the registration.php script that the user calls, not the registration.html page.

- a. This page can be accessed directly or through the Sign Up link on the Driver Login page.
- b. The fields with bold label text are required, the others are optional. If any information is missing from a required field, the textbox is highlighted in red with the error message “Error: information missing” is displayed.
- c. Error messages are also displayed on the page in red text, as shown in Figure 1, if the associated JavaScript and/or PHP validation functions return false.
- d. The national id must be ten integers long with a hyphen separating the first six digits from the last four. Use the JavaScript function **isIdValid(id)** as described in Assignment 1 that returns a Boolean value to indicate validity. Create a similarly named function in PHP declared as **isIdValid(string \$id):boolean** where the *id* parameter is the data entered in the textbox.
- e. The license number is a fifteen-digit string with the first seven digits being randomly generated and the remaining eight digits being a valid calendar date in the format YYYYMMDD that is **at least** 16 years before the current date. Create a JavaScript function called **isLicenseNumValid(LicenseNo)** that returns a Boolean value to indicate validity. Create a similarly named function in PHP declared as **isLicenseNumValid(string \$LicenseNo):boolean**.

- f. The first and last names must be characters only. Use the JavaScript function **isNameValid(name)** from Assignment 1 and create a similarly named function in PHP declared as **isNameValid(string \$name):boolean** where the *name* parameter is the data entered in the textbox.
- g. The email on the client side must as far as possible accept all valid email addresses as defined by the RFC822 standard. Create a JavaScript function called **isEmailValid(email)** that returns a Boolean value based on the example code

```
/**
 * Code taken from: https://stackoverflow.com/questions/940577/javascript-regular-expression-email-validation
 */
function checkEmail() {
    var email = document.getElementById('txtEmail');
    var filter = /^((([^\<>()\\[\]\/\.,;:\s@"]+(\.[^\<>()\\[\]\/\.,;:\s@"]+)*)(("[.]+)@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\]|\([a-zA-Z\-\0-9]+\.\.?[a-zA-Z]{2,}))$)/;
    if (!filter.test(email.value)) {
        alert('Please provide a valid email address');
        email.focus();
        return false;
    }
}
```

Figure 2: Example JavaScript email validation function

- shown in Figure 2. Also create a similarly named PHP function declared as **isEmailValid(string \$email):boolean** that uses the [filter_var](#) function and the [email validation filter](#) to perform the email validation.
- h. The telephone number is optional, which means the field can be empty; however, both fields must empty to be considered empty. If one telephone field is filled it is considered an invalid telephone format. Create a JavaScript function called **isTelephoneValid(telephone)** that checks both fields – called *the prefix* and *the line number* fields respectively and returns a Boolean value to indicate validity. The prefix field is a 3-digit number that cannot start with a 0 or a 1. The line number field is a 4-digit number. Create a similarly-named PHP function declared as **isTelephoneValid(string \$prefix, string \$line_number):boolean**.
 - i. The address fields are optional, which means they can be empty; however, having the Address 2 field filled and the Address 1 field empty is invalid. Fields must be only alphanumeric characters and spaces. Use the JavaScript function **isAddressValid(addr)** from Assignment 1, that returns a Boolean value to indicate validity. Create a similarly-named PHP function declared as **isAddressValid(string \$addr):boolean** to perform validation on the server.
 - j. Parishes are to be hard coded into the HTML page.
 - k. Clicking the Register button invokes the **newDriver(driver)** JavaScript function. If the JavaScript function does not run, the PHP script **register_driver.php** must be aware of this and perform validation on the server side. If the JavaScript function detects errors, the **register_driver.php** script is not invoked. If the JavaScript runs and does not detect errors, it must tell the PHP script that no server-side validation is needed.

[25 marks]

2. Figure 3 shows the VRLMS Driver Sign In page. The HTML file for this page is named **index.html**, which is used by the **index.php** script. The user interacts with the index.php

Barbados Revenue Authority
Vehicle Licensing and Registration System

Driver Sign In

National ID

Password

[Forgot Password](#) [Sign Up](#)

Figure 3: The VLRMS driver log in page

script, not the index.html page.

- This is the first page members of the public sees of the application. This page is shown to all users that are not logged in.
- When a user attempts to access any other page in the system and they are not logged in, they are to be redirected to this page and made to log in.
- This page requires two pieces of information: a national ID and a password. The data format of both the national ID and password data must be validated by two JavaScript and PHP functions:
 - isNationalIdValid(*natL_id*)** and **isNationalIdValid(string *\$natL_id*):boolean**. Takes the string entered in the username textbox as parameter *natL_id* and returns a Boolean value that indicates whether the string is in the correct format. A valid national id consists of eleven characters: six digits representing a valid date in the format YYMMDD; one hyphen (-); and four digits.
 - isPasswordFormatValid(*passwd*)** and **isPasswordFormatValid(string *\$passwd*):boolean**. This is the same function as Assignment 1. The specifications are given here for convenience. Takes the string entered in the password textbox as parameter *passwd* and returns a Boolean value that indicates whether the string is in the correct format. A valid password must: (i) be between 8 – 16 characters long; (ii) contain only alphanumeric characters but must start with a character and have at least one number.

These functions **can use regular expressions, but string manipulation functions are advised**. As described in Question 1, the PHP validation functions are only used if the JavaScript functions are not invoked.

- The Forgot Password link remains on the page when clicked.
- Create a PHP script called **sign_in_user.php** that takes a valid national ID and password and checks it with entries in the **Citizen** table of the database. If the id/password pair matches a database record, then retrieve information from the record, and update the PHP session management superglobal `$_SESSION` the first name, last name and license number. **Also see Question 7.**

- f. Errors generated either from an improperly formatted national ID or password, or if the user is not a valid user, must be displayed in a red font on the same page. No dialog boxes are to be used to display the error messages. **[20 marks]**
3. Figure 4 shows the VLRMS that drivers would use to upload a picture of their driver's license. This page is displayed if the driver registers using the **registration.php** script, but

Figure 4: VLRMS upload driver's license image page

no matching driver's license number was found for the national ID and name provided by the user.

- The HTML file for this page is called **license_upload.html** and it is used by the PHP script of the same name - **license_upload.php**.
 - This page is under [session management](#). This means that only a logged in user can access this page. If an unauthorized user attempts to access this page, they are redirected to the Driver Sign In page and made to log in.
 - This page requires two pieces of information: the license number and an image in the PNG or JPG format that represents a driver's license image. The license number is validated using the same function as described in Question 1. The uploaded file type is checked during the [handling file upload process](#).
 - You must demonstrate that the **license_upload.php** script updates the Citizens table with the license number provided by this form and sets the **notify_govt_license_verification** flag to ON. **[15 marks]**
4. Figure 5 shows the VLRMS that drivers would use to upload a copy of their insurance cover note. This page is displayed if the driver registers using the **registration.php** script, and

Figure 5: VLRMS upload copy of insurance cover note page

then no unexpired insurance policy was found for a car with the given national ID provided by the user.

- a. The HTML file for this page is called **cover_note_upload.html** and it is used by the PHP script of the same name – **cover_note_upload.php**.
 - b. This page is under [session management](#). This means that only a logged in user can access this page. If an unauthorized user attempts to access this page, they are redirected to the Driver Sign In page and made to log in.
 - c. This page requires four pieces of information: the policy number, the expiry date, the issued date, and a PDF document that represents the insurance cover note. The dates field must use the HTML5 **date** input type so that no additional validation is needed. The policy number is validated with the JavaScript function **isPolicyIdValid(policy_id)** and its corresponding PHP function **isPolicyIdValid(string \$policy_id):boolean**. A valid policy ID starts with three capitalized letters from the following groups: ICB, SGI, CGI COI, CIB and TIC. These are followed by eleven digits and then one of the following letters: A, L, H or V. The uploaded file type is checked during the [handling file upload process](#).
 - d. You must demonstrate that the **cover_note_upload.php** script updates the Insurance table for the given policy number with the dates provided by this form and sets the **notify_govt_insurance_verification** flag to ON. **[15 marks]**
5. The main console page for drivers, named **public_console.html**, is shown in Figure 6.
- a. This is the only page a driver sees after successfully signing in, that is, a valid cover

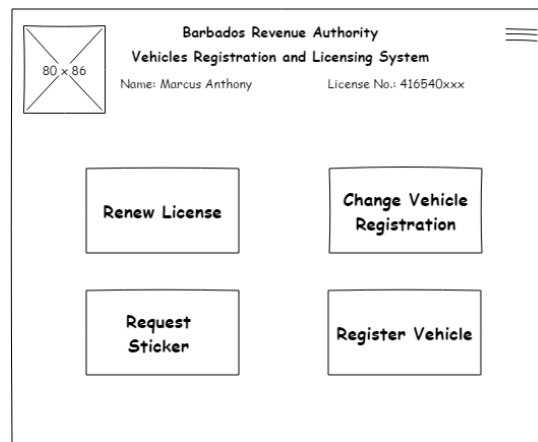


Figure 6: The main console page for drivers

note and license is in the database for the driver. If the user is not signed in, they are redirected as described in Question 1.

- b. The page displays the user's first name, last name and license number that is retrieved from the database. The panels are displayed in the arrangement shown. The text in the panels are links that remain on the console page when clicked.

[10 marks]

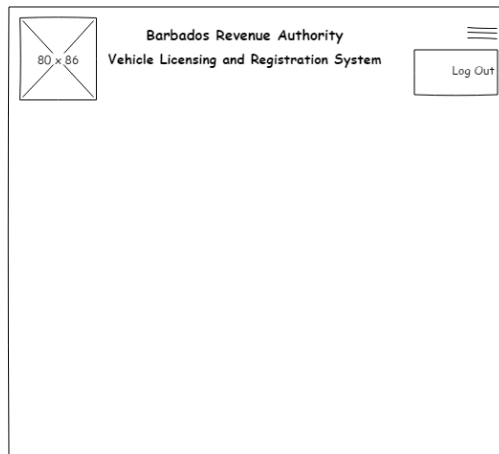


Figure 7: The dropdown menu from the hamburger menu item

6. The hamburger menu appears in the position shown in Figure 7 for each page whenever the hamburger icon is clicked. Its appearance toggles by clicking the icon. It sits above the page content. It has one menu item, Log Out, that invokes the **logout.php** script to log out the user and redirect them to the sign in page.

[10 marks]
7. The data for the application is stored in a database called vlrms. The SQL file that contains the table is provided in a file called vlrms.sql. Use this file to create the database for the assignment.

[5 marks]