

SFWRTECH 4WP3 – Advanced Web Programming - Assignment #2

Weight

4% of total course grade

Due date

Monday February 7th at 11:59pm

Primary learning objectives

- Interpret written instructions to develop a web application to requirements
- Use JavaScript to create an interactive website (event handlers, modifying the DOM)
- Use a front-end framework to style a website

Requirements

For Assignment #1 Part B you were asked to create a home page for a **Hamilton Homes** website, using Bootstrap Version 5 to create a navigation bar. On that navigation bar there was supposed to be a “Calculator” navigation option. In this assignment we will work on the Calculator page and create a mortgage payment calculator with form validation. Use the same navigation bar from Assignment #1, this time with “Calculator” as the active page, and save your solution file as calculator.html. You only need to submit the Calculator page, but you can submit additional files if you like (e.g. an external JavaScript file where you write your JavaScript code).

On the Calculator page, a web form should be presented to the user with the following inputs:

- Four text input text boxes:
 - Mortgage Amount
 - Interest Rate
 - Loan Length (Years)
 - Postal Code
- A calculate button
- A reset button

Do not use dropdowns, select boxes, or other types of input. Do not use the “number” type input either that restricts inputs to numbers (it must be type “text”). Note that we might actually do these things normally to help restrict user input to valid values, but we’re focused on learning form validation and how numbers work with JavaScript on this assignment.

The form input text boxes and buttons should be styled using Twitter Bootstrap. Use Version 5 of Bootstrap. In particular, the calculate button should be styled as the “primary” button, and the reset button should be styled as the “secondary” button, using the appropriate Bootstrap classes (see: <https://getbootstrap.com/docs/5.1/components/buttons/>).

When the user clicks the calculate button, a JavaScript function should execute and perform **form validation** (see: https://www.w3schools.com/js/js_validation.asp). Form validation is when we check that form values are valid (are any values blank, are any values the incorrect format/type, etc.). Your JavaScript function should check for the following errors with the form data that was entered, and it should report any of these errors to the user below the form by modifying the DOM.

Validation Errors:

- Mortgage Amount must be a positive real number.
 - Report the error to the user as “Mortgage Amount must be a positive number”.
- Interest Rate must be a positive real number.
 - Report the error to the user as “Interest Rate must be a positive number.”
- Loan Length must be a positive integer between 5-30.
 - Report the error to the user as “Loan Length must be between 5-30 years.”
- Postal Code must be a 7 character length string, and the first character of the string MUST be ‘L’. Do not worry about checking that the rest of the characters are a valid postal code, just check for the ‘L’.
 - Report the error to the user as “Must be located in Hamilton”.

To help with identifying errors, you may want to use these functions:

- `isInteger`: https://www.w3schools.com/jsref/jsref_isinteger.asp
 - Can test whether something is an integer
- `parseFloat`: https://www.w3schools.com/jsref/jsref_parsefloat.asp
 - Parses text into a float, returns NaN if it is not a number
- `isNaN`: https://www.w3schools.com/jsref/jsref_isnan.asp
 - Checks to see if argument is not a number (NaN)

If there are errors, report these errors as list items in an unordered list below the form. Place the unordered list inside a **single** Bootstrap danger alert (see: <https://getbootstrap.com/docs/5.1/components/alerts/>). Use the validation styles and associated classes of Twitter bootstrap (<https://getbootstrap.com/docs/5.1/forms/validation/#server-side>) to change the appearance of the form input text boxes (the input text box border colour and icon) to either invalid (if there is an error associated with that input), or valid (if there is no issue with that input text box). You can achieve this functionality by adding and removing the `.is-invalid` and `.is-valid` classes from the form input text boxes when the calculate button is clicked.

If there are no validation errors, use a Bootstrap success alert to present the estimated monthly mortgage payment, you can use the formula on this page (scroll down):

<https://www.bankrate.com/calculators/mortgages/mortgage-calculator.aspx>.

If the user clicks the calculate button again, form validation and potential monthly mortgage payment calculation should take place again, including updating any of the above messages and styling changes.

Clicking the reset button should clear all data from the form elements, remove any validation styles from input text boxes, and remove any error or success alerts and messages that have been displayed to the user.

Note: If you wish to use jQuery you may do so, but it is not required for this assignment. Do not use anything but JavaScript and jQuery though (i.e. no Angular, React, Vue, etc.).

Submission

Zip your solution as assignment2.zip and upload it to the dropbox on Avenue.

Marking rubric

Component	Description	Marks
Form	Correct elements	5
Form styling	Bootstrap used to style form	10
Click events	Click event handlers setup for calculate, reset	5
Reset	Reset button functionality	10
Validation	Validation rules	20
Error reporting	Reporting each error	20
Success and error styling	Error and success form inputs styling	20
Success alert	Success alert and mortgage payment calculation when all values correct	10
Total:		/100