Practical Set 1 Marking/Testing

Weight: 17% (+ 3% from checkpoint)

Note:

- This document is to be read in conjunction with the original weekly exercise documents which define the requirements for each question
- This document provides instructions/guidelines on the Prac Set 1 specific questions as identified in the table below.

			Weight of Marking Criteria (see Rubric on page 2)			
Exercise	**From Week:**	**Question**	**Code Functionality/ Correctness**	**Code Readability**	**Form Design**	**Web Page Design**
A. 50% of Prac Set mark	**Week 2 practical**	Exercise 4*	50%	10%	0%	40%
		Exercise 5*				
		Exercise 6*				
		Exercise 7*				
	Week 3 practical	Exercise 1*				
B. 50% of Prac Set mark	**Week 5 / 6 practical**	Exercise 1	60%	10%	30%	0%
		Exercise 2				

Exercise A

HTML 5 semantic markup and CSS for styling. This is the final marking for the week 2 and week 3 exercises that were reviewed in the checkpoint. Mark these files in the same way you did for the checkpoint. Closer attention was given to the **correctness** of the solution.

- View the files in the Chrome browser and check for correct rendering of html and css according to the page descriptions and css rules defined.
- View the html and css source code for readability
- Validate the html pages for html5 conformity @ validator.w3.org

Exercise B

Web form (week 5 exercise 1) + JavaScript code to enforce form validation rules (week 5 exercise 2).

- Both exercises can be marked as one unit from the week5 folder if present.
- View the form in the Chrome browser and check for a responsive layout by resizing your browser window.
- Check the form validation by running the tests for each validation rule (see **Test Items** below) from the prac document, noting that the form must not submit if any of the validation fails.
- Check that Required field validation is not achieved via the 'required' attribute in the html input device tag. If this is the case, edit the source code to add the 'novalidate' attribute into the form tag and re-run the tests. As stated in the prac document the validation **must** be achieved via JavaScript.
- As stated in the prac document the error messaging must be inline only and meaningful.
- There will be varying levels of achievement for the JavaScript. It is appropriate to award marks toward attempted code even if it does not fully achieve the desired result. This also applies to code that does not work due to syntax errors. Review the descriptors in the rubric to choose the most appropriate percentage to apply.
- Students were told they did not need to write their own regEx and were given sites that they could utilise for sourcing their regEx.

Test Items

Notes

a) Sequencing of these tests is up to the marker but needs to consider the validation rules for each field and that the validation may fire at different times (eg, onblur, onsubmit, etc) depending on the field and solution method.
b) However, some validations really need to be onsubmit (eg, required fields).
c) Error messages **MUST** be achieved with DOM notifications not alert boxes.
d) When error conditions are fixed, the DOM notification for the field should be cleared but this may happen at different times depending upon the field.

1. Testing required field validation: Submit empty form.
   result 1: `Quote Date`, `Staff Number`, `Customer First Name`, `Customer Surname`, `Street Address`, `Suburb` should be flagged as error condition since they are **required** fields.
   result 2: Form **should not** submit

2. Testing required field validation**: Reload form** and enter:

Fields	a) `Quote Date`	b) i `Staff Number`	c) iv `Customer First Name`	c) v `Customer Surname`	c) vi `Street Address`	c) vii `Suburb`	d) xi `Product ID`
Correct Value	*10/11/2021*	*ABC1234*	*Fred*	*Smith*	*18 The Street*	*Town*	*P1*
Result	Form should submit after entering required field values						

3. Testing input data formats: **Reload form** and enter:

Fields	1a) Quote Date	1b) i Staff Number	1c) iv Customer First Name	1c) v Customer Surname	1c) vi Street Address	1c) vii Suburb	1c) viii Postcode	1c) ix Phone	1c) x email	1f) Install Date
Incorrect Value(s)	10112021	hello	Bob=Joe	$mith	!! The Street	H3re	asdf 123 12345	1234567890 0123456789 02123456	@me.com	hello
Corrected Value	*10/11/2021*	*bcd4321*	*Bob-Joe*	*Smith Jones*	*1/18 The Street*	*Here-ville*	*1234*	*0212345678* *0412345678*	*bob@me.com*	*01/12/2021*
Results	1. Incorrect input formats should be flagged as errors in page && form should not be allowed to submit 2. Correct input formats should result in error message being cleared && form should be allowed to submit if no other errors in form input									

4. Testing Product Details & Calculations:

Fields	1d) xi Product ID	1d) xii Product Name	1d) xiii Quantity	1d) xiv Unit Price	2a) Line Total	2b) Subtotal	2c) GST	2d) Total	1d) xv Deposit	2e) Total Due
a. Product 1 Incorrect Values	*P1*	*Some text*	A -3 0 22	B -3 0						
	4a tests validation of Quantity and Unit Price inputs									
b. Product 1 Values	*P1*	*Some text*	*20*	*20.50*	*410.00*	*410.00*	*41.00*	*451.00*	A -3 500.00 *200.00*	*251.00*
	4b tests automatic calculations based on 1 product entered && validation of Deposit. Cells in green are auto updated fields not inputs									
c. Product 1 Values	*P1*	*Some text*	*4*	*20.50*	*82.00*					
Product 2 Values	*P2*	*Some text*	*1*	*320.00*	*320.00*					
Product 3 Values	*P3*	*Some text*	*10*	*200.00*	*2000.00*	*2402.00*	*240.20*	*2642.20*	*1000.00*	*1642.20*
	4c tests automatic calculations based on 3 products entered. Cells in green are auto updates not inputs									