Name: _____   Mark: **_____**/ **100**

Instructions

Your task will be to create a page with an HTML form, with JavaScript validation. This will be submitted as **a single page** with JavaScript, CSS, and HTML on the same page.

Requirements

The assignment must have **a page with a form on it that will be validated**. The form must ask for required fields to be filled, including:

- **Name** (first and last). (required)
- **Email address** (required), which should be checked for proper format. For email validation, make sure:
  - "@" symbol is present, and not in the **first, second-last, or last positions** of the string.
  - "." Character is present, and not in **the first, second-last, or last positions** of the string.
- **Telephone number** (required), which should be checked for proper format which is 12 characters, with hyphens: 204-555-1212).
- **Postal Code** (required), which should be checked for the proper number of characters (6 characters with no space).
- **A full URL** (*not required* but must be validated if present), which should be checked for proper format (starts with "http://")
- **Age** (not required), which must be validated if entered to make sure it is a number, not any other characters. Note: use a text field for better validation, not the number input type.

*Note: "required" means the **user** is required to enter these fields. **All the above must be present in your form, whether required or not.***

This form should **check to see that all the required fields have been filled**. If they have not been filled or **if any of the required fields are not in the proper format, the form should not submit** and **let the user know** that the form is not filled in correctly.

*All form validation must work in **JavaScript** not HTML5. Be sure to add the* `novalidate` *attribute to your* `<form>` *tag.*

2.   JavaScript source code is well-formatted and easy to read.

3. Logical naming conventions and usages of variables, functions and IDs: All must be appropriately named and indicative of their purpose.

4. Use JavaScript comments wherever possible. Do not just comment because you must; explain what the code is doing.

5. In all cases, let the user know why the form is not validating, and how they can fix the problems with either alerts or innerHTML.

Submission Guidelines

Upload to the specified OneDrive folder by the due date.

The file name for your file should be:
21128_2001_Assignment1_lastname_firstname.html

*(Note: All code must be **original** (created by you). Any code that is **copied** from online sources, or from the PDFs, or copied from in-class examples is **plagiarism**. Please refer to the "Academic Misconduct" section of your course outline.)*

Due Date and Time:

Tuesday, June 30th, 2020 at 09:00

Marking Criteria

Marks for this project will be given as follows:

University of Winnipeg

Criteria	Needs Work	Proficient	Excellent	Mark
Form Validation: Required fields	None of the fields are required	Required form fields are not marked as required to the user. (-5 marks) Fields that are supposed to be required can be left blank (-2 marks for each) on the form.	Fields that are supposed to be required must be filled in, and the user can clearly see which fields are required.	
	0	1-19	20	**/ 20**
Marking Comments				
Email Address Validation	No email address validation.	Email address validation could be better.	Email address validation as specified.	
	0	1-9	10	**/ 10**
Marking Comments				
Form Validation: Telephone Number	No Telephone number validation.	More than 12 characters (along with dashes) can be entered. (-5 marks) Less than 12 characters can be entered (-5 marks)	Telephone Validation as specified.	
	0	5	10	**/ 10**
Marking Comments				
Form Validation: URL	No URL validation found.	Form checks for URL even if one is not entered, (-5 marks)	Form checks for proper URL, only if one is filled in.	
	0	5	10	**/ 10**
Marking Comments				
Form Validation: Age	No age validation found.	Form checks for a proper age even if one is not filled in. (-5 marks)	Age validation as specified.	
	0	5	10	**/ 10**

Marking Comments				
Code Formatting	Code is badly formatted.	Code could have been formatted better.	Code is well formatted.	
	0	1-9	10	**/ 10**
Marking Comments				
Logical Naming Conventions	Variable, function and ID names are badly named.	Variables, functions and IDs could have been better named.	Variables, functions, and IDs are appropriately named.	
	0	1-9	10	**/ 10**
Marking Comments				
Errors	More than 3 errors	-3 marks for each error that requires fixing for the code to work (up to 3)	No errors.	
	0	1-7	10	**/ 10**
Marking Comments				
JavaScript Comments	No JavaScript comments anywhere.	JavaScript commenting could have been better.	Good JavaScript commenting.	
	0	1-9	10	**/ 10**
Marking Comments				
			Total Marks	**/ 100**

Additional Comments	