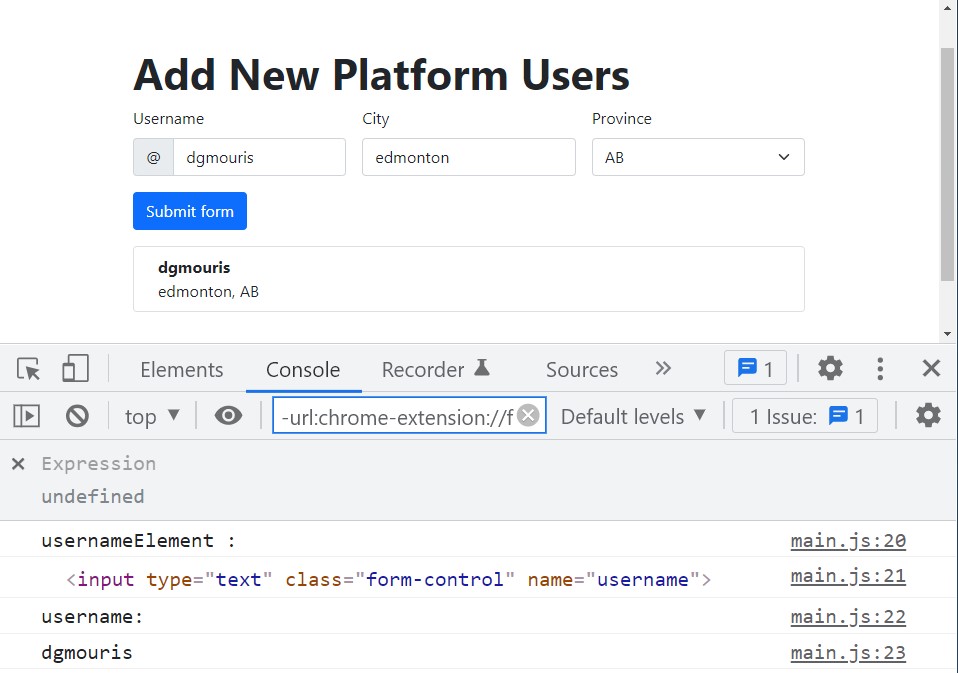
CPSC1520 – JavaScript 4 Exercise: Making Decisions and Forms

# Introduction

Validation of forms is everywhere when you create a JavaScript application and a web application in general. You want to ensure that you’re getting correct and valid information from your users so that you can (for us in the future) save this information.

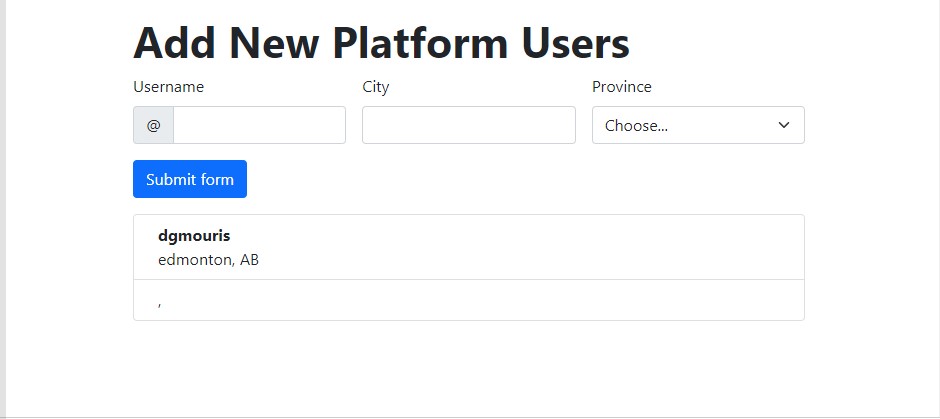
# TASK 1 – The Form and it’s Values.

1. Ensure the link to your JavaScript file is correct
2. Select the form with the class “new-user-form”
3. Add an event listener on the form that will listen for the “submit” event. Create the event object parameter for the event handler (you will need the event object to access the form).
4. Prevent the from submitting using the prevent default function.
5. Create variables for each input **element** using the “elements” attribute on the event target. Note, they should be called: usernameElement, cityElement, provinceElement
6. Create new variables for each input element **value** named: username, city, province.
7. Log the values to the console. Below you will see a screenshot of what the username element and its value will look like so you know exactly what they are. Note: you can remove these console.log lines when you know which are the variables and which are the values of the inputs.
8. Call the provided “addUser” function with the values from step 6 as arguments (so username, city, province) in the correct order (look at the function). The following screenshot shows the addition of a user row and the console log statements from step 7.



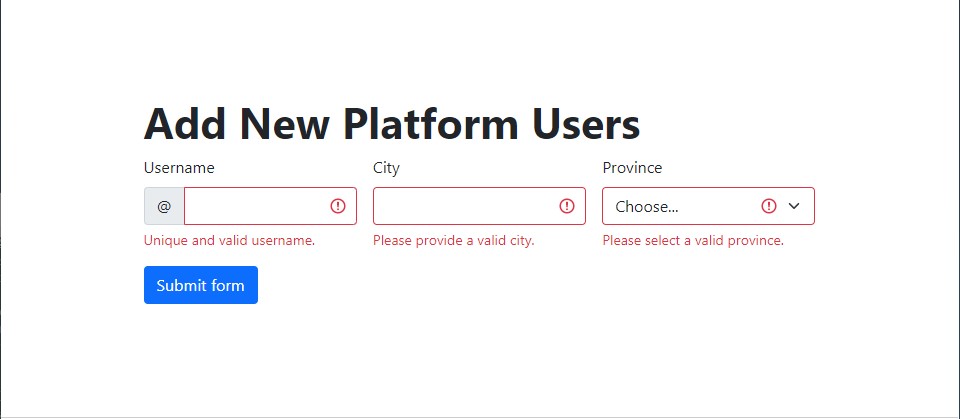
# TASK 2 – Validating the form elements

1. There is an issue with the application. When you leave the form empty, and submit, you see that an empty row is created, and it looks janky and weird. Shown below (see the empty row with only a comma)



1. Create a variable that is called “isFormValid” and set it to “true”. Note: add this line of code and all of the following lines of code above (or before) the “addUser” function is called. In the following steps you’ll use the validator functions “isValueNotEmpty” and “hasWhiteSpace” which return either true or false to validate each form value.
2. Validating username
   1. If **username value does not have whitespace and the username is not empty** this value is valid.
      1. On the usernameElement remove the “is-invalid” class using the element's classList.remove() function.
   2. If the **username does have whitespace or is empty** (the else case) the value is not valid:
      1. Set the “isFormValid” variable to false
      2. On the usernameElement add the “is-invalid” class using the elements’ classList.add() function
3. Validating the city
   1. If the **city value is not empty** this value is valid:
      1. On the cityElement remove the “is-invalid” class using the element's classList.remove() function
   2. If the **city value is empty** (the else case) this value is not valid:
      1. Set the “isFormValid” variable to false
      2. On the cityElement add the “is-invalid” class using the element's classList.add() function
4. Validating the province
   1. If the **province value is not empty** this value is valid:
      1. On the provinceElement remove the “is-invalid” class using the element's classList.remove() function
   2. If the **province value is empty** (the else case) this value is not valid:
      1. Set the “isFormValid” variable to false
      2. On the provinceElement add the “is-invalid” using the element's classList.add() function
5. Finally, if the “isFormValid” value is true, call the “addUser” function passing in our validated values. There is no need for an else block.

Note: Your form should look like the diagram below when it’s not valid (no empty row created):



1. If the form is valid, allow the form to add the row.
2. You reset the values of each input on submit (there are several ways to do this).

# **Submission**

Deploy your lab to NETLIFY and submit the URL Netlify provides to Moodle.

Create a GitHub repository for your lab and upload the URL to Moodle.

1. Drag and Drop the folder that your files are in into Netlify.
2. Click on “Site Settings” (the link is shown below)