

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.

Exercise Step 1 – The Form and it's Values

1. Link your JavaScript file
2. Select the from with the class "new-user-form"
3. Add event listener on the form that will listen to the "submit" event. Pass the "event" object into the event handler.
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 values named: username, city, province
7. Print them out to the console here's an example of what the username element and it's value to the console 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 "addUser" function with the values from step 6 (so username, city, province) in the correct order (look at the function).

It should look like this:

The screenshot shows a web form titled "Add New Platform Users". The form has three input fields: "Username" (containing "dgmouris"), "City" (containing "edmonton"), and "Province" (a dropdown menu showing "AB"). Below these fields is a blue "Submit form" button. Under the button, the form displays the submitted data: "dgmouris" and "edmonton, AB".

Below the form, the Chrome DevTools console is open, showing the following output:

```
Expression
undefined

usernameElement :  main.js:20
username: dgmouris main.js:23
```

Exercise Step 2 – Validating the form elements

1. You see an issue with your application. When you leave the form empty you see that an empty row is created and it looks janky and weird. Shown below:

Username	City	Province
@		Choose...
<button>Submit form</button>		
dgmouris	edmonton, AB	

2. 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.
3. Validating username
 - a. If **username value does not have whitespace and the username is not empty** this value is valid.
 - i. On the usernameElement remove the “is-invalid” using the elements’ classList.remove function.
 - b. If the **username does have whitespace or is empty** (the else case) the value is not valid:
 - i. Set the “isFormValid” to false
 - ii. On the usernameElement add the “is-invalid” using the elements’ classList.add function
4. Validating the city
 - a. If the **city value is not empty** this value is valid:
 - i. On the cityElement remove the “is-invalid” using the elements’ classList.remove function
 - b. If the **city value is empty** (the else case) this value is not valid:
 - i. Set the “isFormValid” to false
 - ii. On the cityElement add the “is-invalid” using the elements’ classList.add function
5. Validating the province
 - a. If the **province value is not empty** this value is valid:
 - i. On the provinceElement remove the “is-invalid” using the elements’ classList.remove function
 - b. If the **province value is empty** (the else case) this value is not valid:
 - i. Set the “isFormValid” to false
 - ii. On the provinceElement add the “is-invalid” using the elements’ classList.add function
6. Return the function early if the “isFormValid” value is false so that the “addUser” function (that is called with our validated values) does not execute and we don’t add an empty row.
Note: Your form should look like the diagram below when it’s not valid (see that there is not an

empty row created):

Add New Platform Users

Username City Province

@ Choose...

Unique and valid username. Please provide a valid city. Please select a valid province.

Submit form

7. If the form is valid allow the form to add the row.
8. You reset the values of each input on submit.

Test Cases

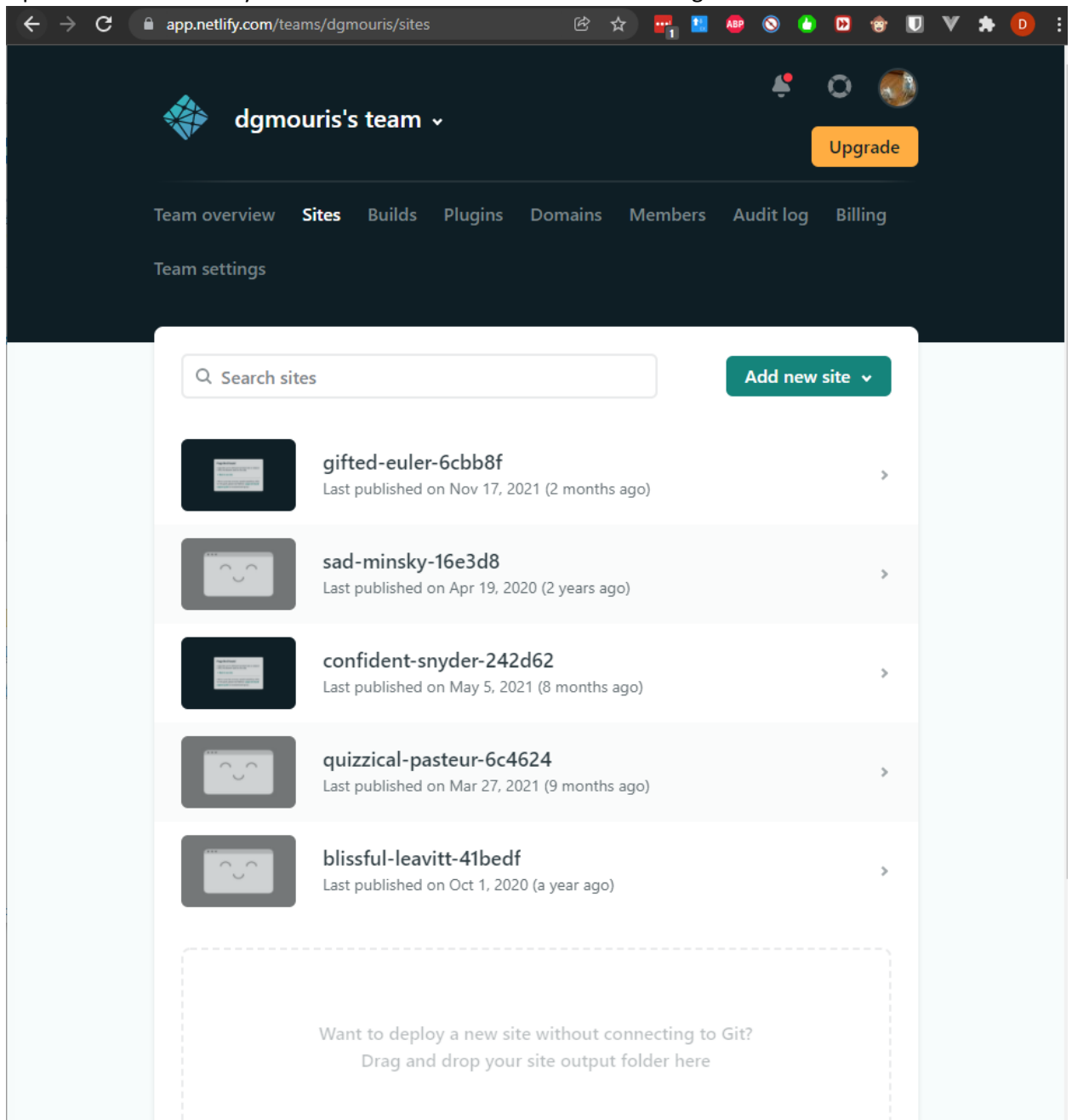
You might be confused and not know what to test here some values to help you out.

- Test case: form invalid
 - Username:
 - City: Edmonton
 - Province: AB
- test case: form invalid
 - Username: ghost gambit
 - City: Edmonton
 - Province: AB
- test case: form invalid
 - Username: marshmallow
 - City:
 - Province: AB
- test case: form invalid
 - Username: marshmallow
 - City: Edmonton
 - Province:
- Test case: valid form
 - Username: marshmallow
 - City: St john's
 - Province: NL

Exercise Step 3 – upload your site to Netlify

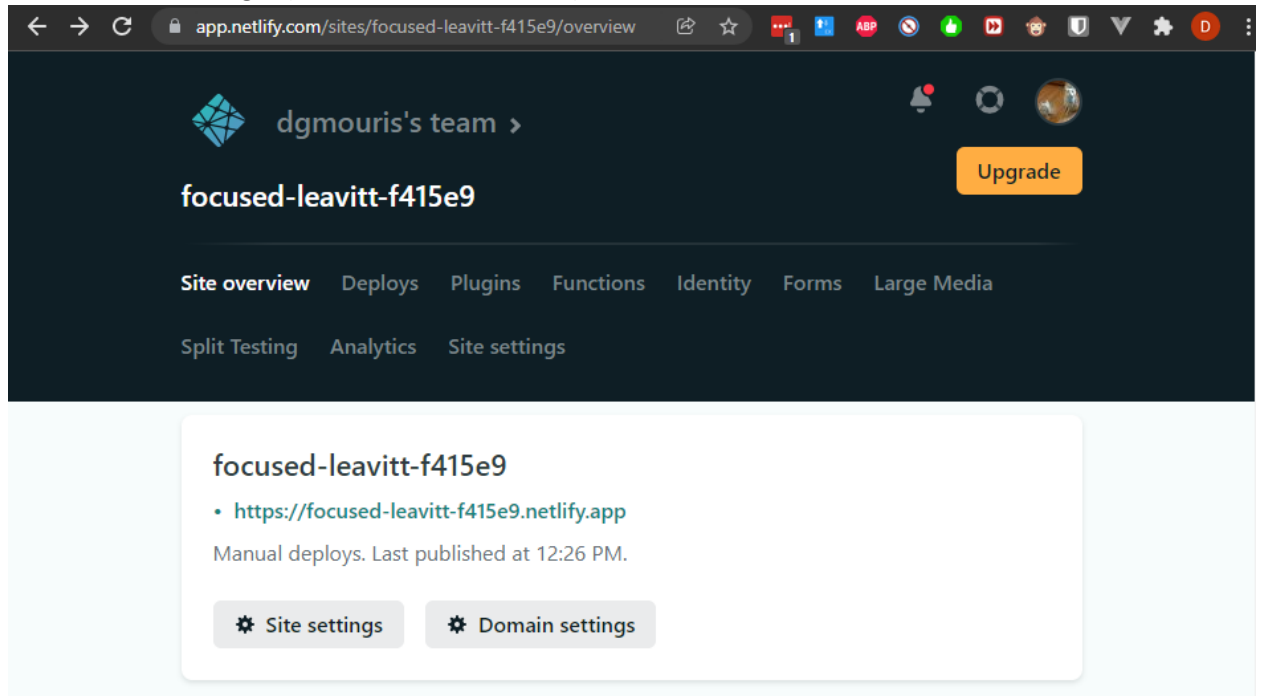
1. Sign in (or signup if you don't have an account) to <https://netlify.com/>

2. Open the “sites” tab in your browser it should look a little something like this.

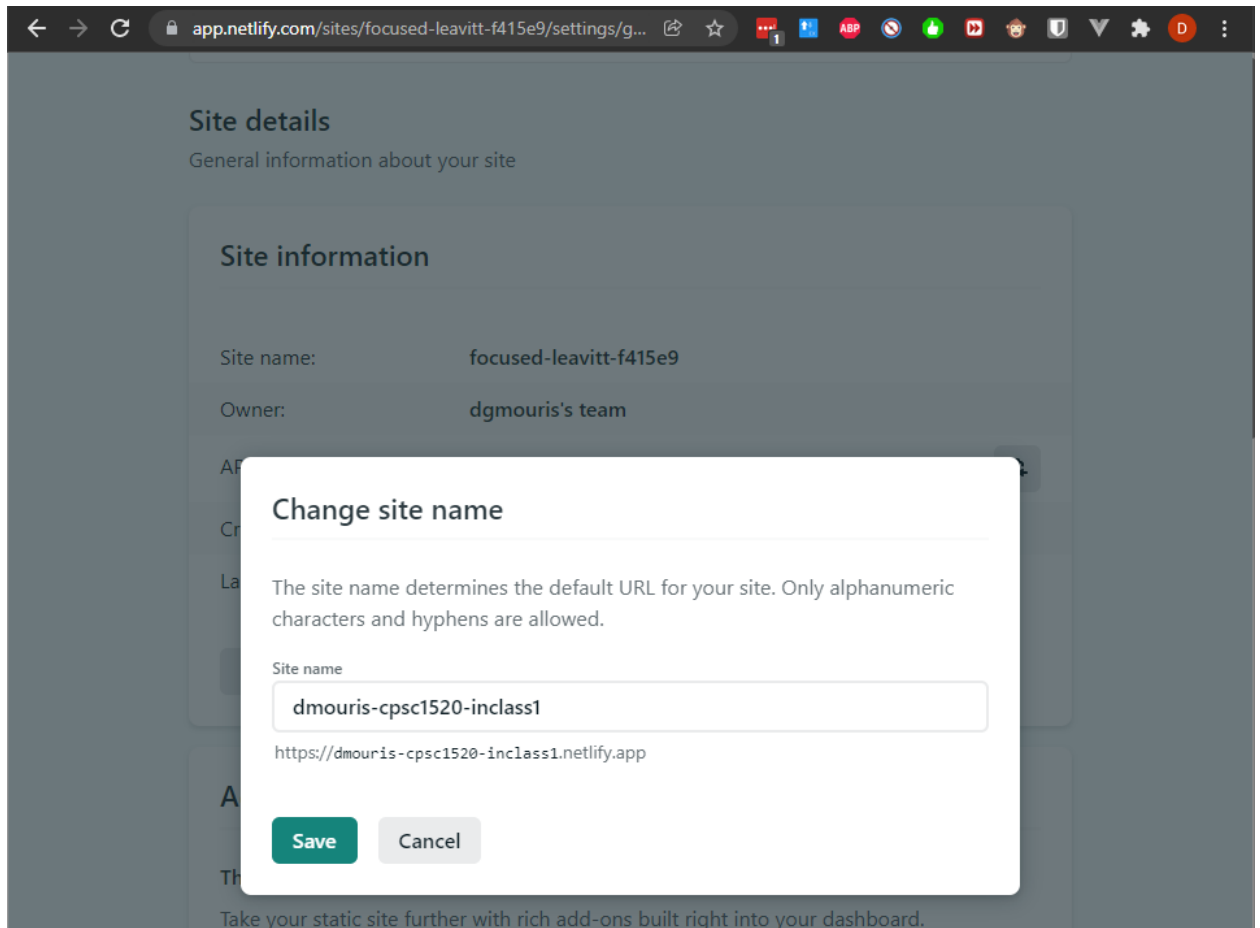


3. Drag and Drop the folder that your files are in into Netlify.

4. Click on “Site Settings” (the link is shown below)



5. Under site details change the site name to be “<your-nait-username>-cpsc1520-inclass4” like so.



Grading

I'll give full marks if:

- The name of **your Netlify app** is renamed to **your name**.
- The form validates each input element separately.
- An invalid form submission should look like this:

A screenshot of a web form titled 'Add New Platform Users'. The form has three input fields: 'Username' with a placeholder '@', 'City', and 'Province' with a dropdown menu showing 'Choose...'. Each field has a red border and a red error icon. Below the fields are three error messages: 'Unique and valid username.', 'Please provide a valid city.', and 'Please select a valid province.'. A blue 'Submit form' button is at the bottom.

- A valid form submission should look this:

Add New Platform Users

Username	City	Province
@ <input type="text"/>	<input type="text"/>	Choose... ▼
<input type="submit" value="Submit form"/>		
<div>ghostgambit edmonton, AB</div>		

When I open your netlify project it should look like this.

If you don't follow the instructions you'll get a zero. There are no marks in between.