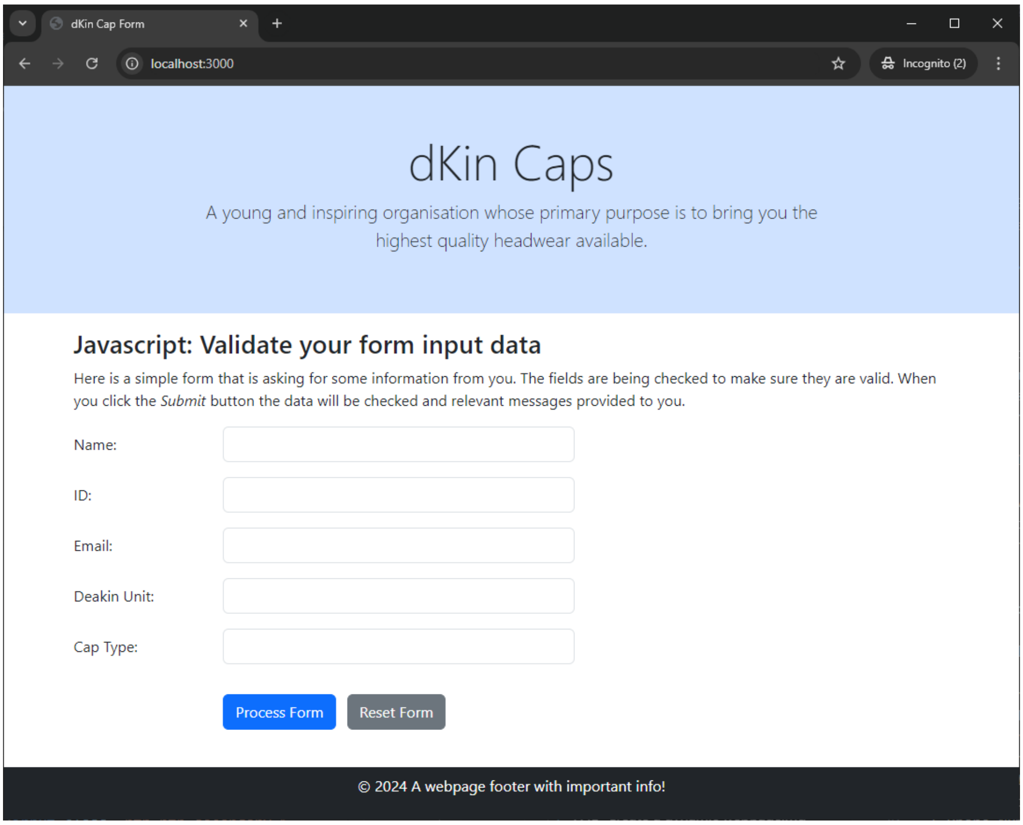
7.3C: Form Validation

Tasks

You are required to complete a simple form where the user can input a number of fields, simulating a page where a student could submit information about themselves and their preferred cap type. The form should ask for 5 separate fields, as shown in the figure below.



dKin Caps

A young and inspiring organisation whose primary purpose is to bring you the highest quality headwear available.

Javascript: Validate your form input data

Here is a simple form that is asking for some information from you. The fields are being checked to make sure they are valid. When you click the *Submit* button the data will be checked and relevant messages provided to you.

Name:

ID:

Email:

Deakin Unit:

Cap Type:

© 2024 A webpage footer with important info!

Task7.3.1 Initial Form Prompts

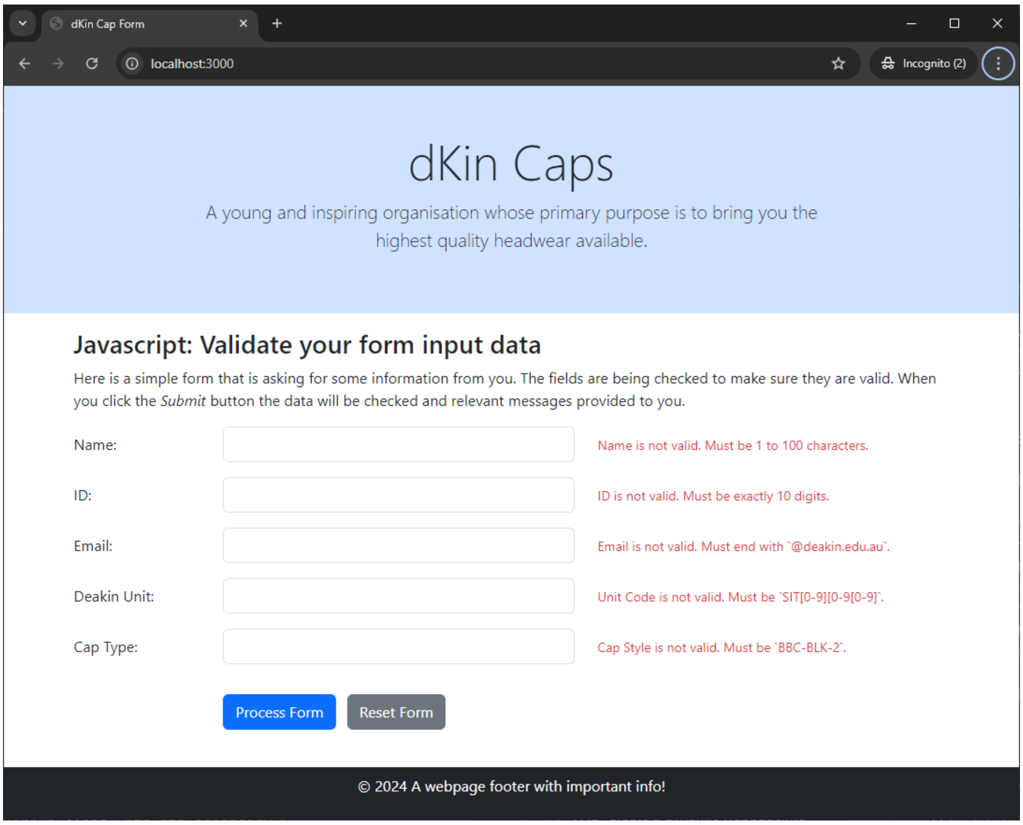
Write a javascript function that allows the data entered by the user to be validated, and where there is an error (the user has input data in an incorrect format) the form should be updated showing the correct format.

The set of:

Form Field	Expected input format rules
Name	This can’t be empty. Some input is required and can’t be greater than 100 characters.
ID	This must be number with exactly 10 digits.
Email	This must be a Deakin email address, i.e., must end in `@deakin.edu.au`

Form Field	Expected input format rules
Deakin Unit Code	Must be 6 characters in size and follow the *3 letters + 3 numbers* format (the letters can be capitalised or lowercase)
Cap Type	Must be of the format *3 letters + 3 letters + 2 numbers* all separated with a dash `-` (i.e., `xxx-xxx-nn`)

An example of the error messages given to the user are shown in the figure below (where no input is given).



dKin Caps

A young and inspiring organisation whose primary purpose is to bring you the highest quality headwear available.

Javascript: Validate your form input data

Here is a simple form that is asking for some information from you. The fields are being checked to make sure they are valid. When you click the *Submit* button the data will be checked and relevant messages provided to you.

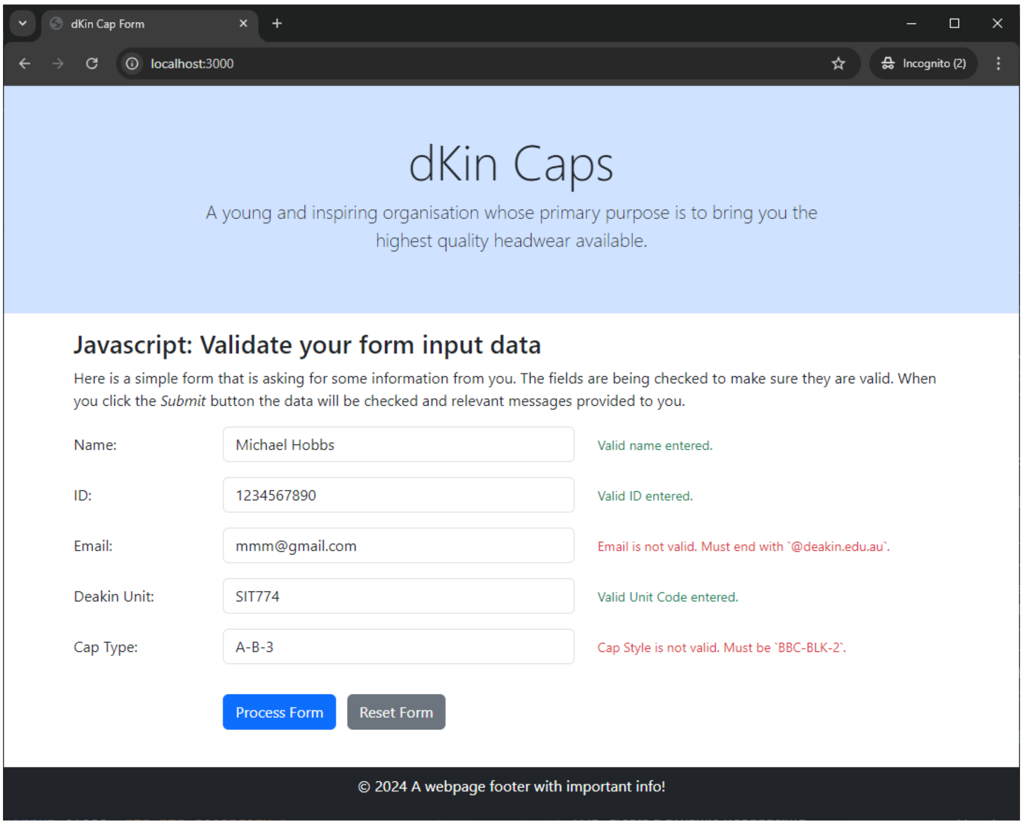
Name:	Name is not valid. Must be 1 to 100 characters.
ID:	ID is not valid. Must be exactly 10 digits.
Email:	Email is not valid. Must end with `@deakin.edu.au`.
Deakin Unit:	Unit Code is not valid. Must be `SIT[0-9][0-9[0-9]`.
Cap Type:	Cap Style is not valid. Must be `BBC-BLK-2`.

© 2024 A webpage footer with important info!

Task7.3.2 Error messages on empty fields

In completing this task, you should consider the following requirements:

1. When a field is correct/valid - give this feedback to the user.
2. Show the error message for each of the fields that has invalid input. These may be in zero or more fields, as shown in the example below:



dKin Caps

A young and inspiring organisation whose primary purpose is to bring you the highest quality headwear available.

Javascript: Validate your form input data

Here is a simple form that is asking for some information from you. The fields are being checked to make sure they are valid. When you click the *Submit* button the data will be checked and relevant messages provided to you.

Name:	Michael Hobbs	Valid name entered.
ID:	1234567890	Valid ID entered.
Email:	mmm@gmail.com	Email is not valid. Must end with `@deakin.edu.au`.
Deakin Unit:	SIT774	Valid Unit Code entered.
Cap Type:	A-B-3	Cap Style is not valid. Must be `BBC-BLK-2`.

© 2024 A webpage footer with important info!

Task7.3.3 Invalid email address message

Suggested Steps

1. Start with a simple script to check that the fields have not been left blank. If the fields are left blank, an error message should be displayed beside the fields.
   - And call this by linking to the *onclick* event to some buttons under your form. E.g.,

```
<!-- Here are the BUTTON for processing the input using the `onclick` event -->
<div class="row">
    <div class="offset-sm-2 col-sm-10">
        <input class="btn btn-primary my-3 me-2" type="button" onclick="validateForm()"
            value="Process Form">
        <input class="btn btn-secondary my-3" type="reset" onclick="resetForm()" value=
"Reset Form">
    </div>
</div>
```

     Here the when the *process* button is clicked a javascript function called `validateForm()` will be called. You can also add another function that resets/clears the form content `resetForm()`.

2. Extend the script code to check that the email must contain the pattern `@deakin.edu.au`, i.e., the input `mick@gmail.com` would be invalid!.
3. Extend the form to also request a *STUDENT ID* and *UNIT CODE*. The script code should check the following criteria:

   - The *ID* must be exactly 10 digits in length.
   - The *UNIT CODE* must have a length of 6 **and** the first 3 characters must be **letters** and the last 3 characters must be **numbers**

4. Extend the script to check the entered *Cap Type* follows the syntax of `xxx-xxx-nn`.

   - A regular expression of the form `/^[A-Za-z]{3}-[A-Za-z]{3}-[0-9]{2}$/`

5. Add a *RESET* button into the form which clears each of the ‘error messages’.

Hints

- By naming the *message fields* with `id`’s, these elements in the DOM can be found and manipulated/changed. For example, if the name field was empty, you could: `document.getElementById("namemsg").innerHTML = "<em> You did not enter your name </em>";`
- In *Bootstrap*, investigate the `form-text`, `text-danger` and `text-success` class styles for use within the form.

- You can modify class attributes using combinations of: `document.getElementById("namemsg").classList.add("text-danger");` and `document.getElementById("namemsg").classList.remove("text-success");`

   - https://www.w3schools.com/jsref/prop_element_classlist.asp

- Review the various methods/functions available to work with *JavaScript Regular Expressions* for matching patterns in strings:

   - https://www.w3schools.com/js/js_regexp.asp
   - https://www.w3schools.com/jsref/jsref_obj_regexp.asp

What will you submit?

You should submit:

- Screenshots (similar to the above examples) of the web page.
- HTML file of the web page with JavaScript.
- JavaScript file that implements the validation and reset functions.