

## Medical Informatics (BIOM9450)

### Tutorial 4: Client-Side Form Validation Using JavaScript

There were many functions and scripts used to complete this tutorial and, in this report, I would be going through the design choices for the JavaScript validation process.

The first part was to validate the email address, and this was done using the code below:

```
<!-- Adding Email Address and Verifying It-->
<div class="form-group">
  <label class="col-md-4 control-label" for="Email">Email</label>
  <div class="col-md-4">
    <input id="Email" name="Email" type="text" placeholder="Enter Email" class="form-control input-md" required="" value="" onChange="validEmail(this.value)">
    <p id="EmailCheck"></p>
    <script>
      function validEmail(val) {
        const check = /^[0-9a-zA-Z.-]+@[0-9a-zA-Z.-]+\.[a-zA-Z]{2,4}$/g;
        if (check.test(String(val).toLowerCase())==true) {
          document.getElementById("EmailCheck").value = true;
          document.getElementById("EmailCheck").innerHTML = "No errors";
        } else {
          document.getElementById("EmailCheck").value = false;
          document.getElementById("EmailCheck").innerHTML = "Email Address is invalid";
        }
      }
    </script>
  </div>
</div>
```

As shown from the code snippet, a regex expression is used to check for numbers and letters before and after the “@”, and 2 to 4 characters after the “.”

Next, there is the typing of the password and confirmation of password. The code snippet is as follows:

```
<!-- Adding Re Enter Password and Verifying It-->
<div class="form-group">
  <label class="col-md-4 control-label" for="Password">Password</label>
  <div class="col-md-4">
    <input id="Password" name="Password" type="password" placeholder="Enter Password" class="form-control input-md" required="" value=""
      onChange="validPassword(this.value)">
    <p id="PasswordCheck"></p>
    <script>
      function validPassword(val) {
        var check = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])/g;
        if (check.test(String(val))==true && val.length > 8) {
          document.getElementById("PasswordCheck").value = true;
          document.getElementById("PasswordCheck").innerHTML = "No errors";
        } else {
          document.getElementById("PasswordCheck").value = false;
          document.getElementById("PasswordCheck").innerHTML = "Passwords must contain at least 8 characters and include uppercase, lowercase and numbers";
        }
      }
    </script>
  </div>
</div>

<!-- Adding Re Enter Password and Verifying It-->
<div class="form-group">
  <label class="col-md-4 control-label" for="ReEnter Password">Re-enter Password</label>
  <div class="col-md-4">
    <input id="ReEnterPassword" name="ReEnter Password" type="password" placeholder="Repeat Password" class="form-control input-md" required="" value=""
      onChange="validReEnterPassword(this.value)">
    <p id="ReEnterPasswordCheck"></p>
    <script>
      function validReEnterPassword(val) {
        var Password = document.getElementById("Password");
        if (Password.value == val) {
          document.getElementById("ReEnterPasswordCheck").value = true;
          document.getElementById("ReEnterPasswordCheck").innerHTML = "No errors";
        } else {
          document.getElementById("ReEnterPasswordCheck").value = false;
          document.getElementById("ReEnterPasswordCheck").innerHTML = "Passwords must match";
        }
      }
    </script>
  </div>
</div>
```

As seen from the code snippets, the passwords were again checked using regex to include uppercase letters, lowercase letters and numbers. After that, the whole string was compared with the one typed in the re enter section and accept it if it is correct.

Following that, I have included the First Name and Last Name section which are identical. This is shown in the code snippets below:

```
<!-- Adding First Name and Verifying It-->
<div class="form-group">
  <label class="col-md-4 control-label" for="FirstName">First Name</label>
  <div class="col-md-4">
    <input id="FirstName" name="FirstName" type="text" placeholder="Enter First Name" class="form-control input-md" required="" value=""
      onChange="validFirstName(this.value)">
    <p id="FirstNameCheck"></p>
    <script>
      function validFirstName(val) {
        var check = /^(^([A-Za-z \-' ]+$))/g;
        if (check.test(String(val)) == true) {
          document.getElementById("FirstNameCheck").value = true;
          document.getElementById("FirstNameCheck").innerHTML = "No errors";
        } else {
          document.getElementById("FirstNameCheck").value = false;
          document.getElementById("FirstNameCheck").innerHTML = "First name should contain only letters, apostrophes, spaces and hyphens";
        }
      }
    </script>
  </div>
</div>

<!-- Adding Last Name and Verifying It-->
<div class="form-group">
  <label class="col-md-4 control-label" for="LastName">Last Name</label>
  <div class="col-md-4">
    <input id="LastName" name="LastName" type="text" placeholder="Enter Last Name" class="form-control input-md" required="" value="" onChange="validLastName
      (this.value)">
    <p id="LastNameCheck"></p>
    <script>
      function validLastName(val) {
        var check = /^(^([A-Za-z \-' ]+$))/g;
        if (check.test(String(val)) == true) {
          document.getElementById("LastNameCheck").value = true;
          document.getElementById("LastNameCheck").innerHTML = "No errors";
        } else {
          document.getElementById("LastNameCheck").value = false;
          document.getElementById("LastNameCheck").innerHTML = "Last name should contain only letters, apostrophes, spaces and hyphens";
        }
      }
    </script>
  </div>
</div>
```

Again, names are checked using regex to only contain letters, apostrophes, spaces, and hyphens.

Lastly, there is the checking of the Date of Birth using the code snippet below:

```
<!-- Adding Date of Birth and Verifying It-->
<div class="form-group">
  <label class="col-md-4 control-label" for="DOB">DOB(dd/mm/yyyy)</label>
  <div class="col-md-4">
    <input id="DOB" name="DOB" type="text" placeholder="Enter DOB" class="form-control input-md" required="" value="" onChange="validDOB(this.value)">
    <p id="DOBCheck"></p>
    <script>
      function validDOB(val) {
        var birthday = val; // Don't get Date yet...
        var regexVar = /^(^([0-9]{2})\/([0-9]{2})\/([0-9]{4}))$/; // add anchors; use literal
        var regexVarTest = regexVar.test(birthday); // pass the string, not the Date
        var userBirthDate = new Date(birthday.replace(regexVar, "$3-$2-$1")); // Use YYYY-MM-DD format
        var todayYear = (new Date()).getFullYear(); // Always use FullYear!!
        var cutOff150 = new Date();
        var future = new Date();
        cutOff150.setFullYear(todayYear - 150);
        future.setFullYear(todayYear);
        if (!regexVarTest) { // Test this before the other tests
          document.getElementById("DOBCheck").value = false;
          document.getElementById("DOBCheck").innerHTML = "Please enter date of birth as dd/mm/yyyy";
        } else if (isNaN(userBirthDate)) {
          document.getElementById("DOBCheck").value = false;
          document.getElementById("DOBCheck").innerHTML = "Invalid DOB";
        } else if (userBirthDate < cutOff150) {
          document.getElementById("DOBCheck").value = false;
          document.getElementById("DOBCheck").innerHTML = "Invalid DOB: More than 150 years old";
        } else if (userBirthDate > future) {
          document.getElementById("DOBCheck").value = false;
          document.getElementById("DOBCheck").innerHTML = "Invalid DOB: Future date";
        } else {
          document.getElementById("DOBCheck").value = true;
          document.getElementById("DOBCheck").innerHTML = "No errors";
        }
      }
    </script>
  </div>
</div>
```

This was checked using regex to ensure that the Date of Birth was a valid date that was not too far in the past or in the future. To add on, these two screenshots below show the results when there is an input. The one on the left shows inputs with no errors and the one on the right shows incorrect inputs.

### User Details

First Name

No errors

Last Name

No errors

DOB(dd/mm/yyyy)

No errors

Gender

Male ▾

☐ I want to receive marketing material

### User Details

First Name

First name should contain only letters, apostrophes, spaces and hyphens

Last Name

Last name should contain only letters, apostrophes, spaces and hyphens

DOB(dd/mm/yyyy)

Invalid DOB

Gender

Male ▾

☒ I want to receive marketing material

When the forms are accurate, the user can then hit the “Register Now!” button and the system checks if all their inputs are accurate. This is done using the code snippet below:

```
<!-- Adding Reset and Register Now Buttons -->
<div class="form-group">
  <label class="col-md-4 control-label" for="Register"></label>
  <div class="col-md-8">
    <button id="Register" name="Register" class="btn btn-default">Cancel</button>
    <button id="RegisterNow" name="RegisterNow" class="btn btn-success" onclick="myFunction()" formaction = "http://engpwws005/z5161724$/TUT4_w5_Ian/TUT4_w5_Ian_3.html">Register Now!</button>
  <p id="FormCheck"></p>
  <script>
    function myFunction() {
      var FirstName = document.getElementById("FirstNameCheck").value;
      var LastName = document.getElementById("LastNameCheck").value;
      var DOB = document.getElementById("DOBCheck").value;
      if (FirstName == true && LastName == true && DOB == true) {
        document.getElementById("FormCheck").innerHTML = "Registration Success";
        location.href = "http://engpwws005/z5161724$/TUT4_w5_Ian/TUT4_w5_Ian_3.html";
      } else {
        alert("Oops! It appears that you have not filled in the form correctly!");
      }
    }
  </script>
</div>
</div>
```

If there are some errors in the form filling in, the following alert will be displayed on screen along with a toolbox showing to fill in a missing detail:

Slide 1 x TUT4\_JS\_W5.pdf x Service UNSW WWDC Main Page x

Not secure | engpwws005/z5161724\$/TUT4\_W5\_lan/TUT4\_W5\_lan.html?FirstName=Ian-Nathan&LastName=Hadinoto1&DOB=15%2F07%2F2022&Gender=1...

Apps myUNSW Portal Moodle Outlook Gmail OneDrive Library Genesis Reading list

engpwws005 says  
Oops! It appears that you have not filled in the form correctly!  
OK

**Password**  
Enter Password

**Re-enter Password**  
Repeat Password

Login

**User Details**

**First Name**  
Enter First Name

Please fill out this field.

**DOB(dd/mm/yyyy)**  
Enter DOB

**Gender**  
Male

☐ I want to receive marketing material

Cancel Register Now!

Created with: [https://www.w3schools.com/css/css\\_templates.asp](https://www.w3schools.com/css/css_templates.asp)

Logos are obtained from <https://99designs.com.au/inspiration/logos/messaging>

If the details, the First Name, Last Name and Date of Birth, are all “true”, the webpage changes to show the successful registration, as seen below:

Slide 1 x TUT4\_JS\_W5.pdf x Service UNSW WWDC Program x

Not secure | engpwws005/z5161724\$/TUT4\_W5\_lan/TUT4\_W5\_lan\_3.html?FirstName=Ian&LastName=Hadinoto&DOB=15%2F07%2F1998&Gender=1&Regist...

Apps myUNSW Portal Moodle Outlook Gmail Citrix Gateway UNSWBC 2021 Mai... Basic Git command... OneDrive Library Genesis Reading list

About Us Workshop Program Contact Information Service UNSW WWDC 2022

**Registration Successful!**

Created with: [https://www.w3schools.com/css/css\\_table.asp](https://www.w3schools.com/css/css_table.asp)

Logos are obtained from <https://99designs.com.au/inspiration/logos/messaging>