

# Code Explanation

## User Interface of Register page

The screenshot shows the 'Register/Login Account' page of the 'Service UNSW WORKSHOP'. The page has a dark blue header with the site name and a navigation bar with links to 'Home', 'Workshop programme', and 'Register/Login Account'. The main content area is a light blue box with two sections: 'Login details' and 'User details'. The 'Login details' section contains three input fields: 'Email' (placeholder: 'Enter your Email'), 'Password' (placeholder: 'Enter your password'), and 'Confirm Password' (placeholder: 'Re-Enter same password'). The 'User details' section contains four input fields: 'First Name' (placeholder: 'Enter your First Name'), 'Last Name' (placeholder: 'Enter your Last Name'), 'DOB(dd/mm/yyyy)' (placeholder: 'Enter your Date of birth'), and a 'Gender' dropdown menu (currently set to 'Male'). Below these fields is a checkbox labeled 'I want to receive marketing material' which is checked by default. At the bottom of the form are two buttons: 'Cancel' and 'Register Now!'.

Figure 1 User Interface of Register page

User can access this page by clicking the “Register/Login Account” tab on the top navigation bar. Users need to enter both login details and user details for successful creating an account. Checkbox for marketing material is checked by default.

## Validation of input term

Every entry has a validation which provide feedback to user in real time except “Confirm password” term which only provide feedback when off-focus. This is achieved by `addEventListener()` function to the element which monitor the input when user release a key (locate at Register.js line1-16).

There are 5 checkers function (check Email, Password, confirm password, first name, last name) in total which check if the input term in correct regex format. The Dob checker is checking the dob by string manipulation. The Confirm password checker is called only when user off-focus the confirm password input and other will be called when listener gets a signal.

Every checker function has ability to change an element’s text and class called `_____HL` which indicates the highlight area (feedback) locate next to the input box which provide

feedbacks as Figure 2 shown. The feedback will give user suggestion on which part may contain errors in red color with yellow border and keep green if no error.

Figure 2 shows a form with two input fields. The 'Email' field contains the text 'a' and is highlighted with a red border and the message 'Email Address is invalid' in red text. The 'Password' field contains a series of dots '.....' and is highlighted with a green border and the message 'No errors' in green text.

Figure 2 highlighted text

There are also 4 validation function to help checker function determine whether the inputs in correct format which locate between line 183-225. The separation of checker and validation function makes it easy to change and less fragile in later adjustments.

Email validation is by regex format of :

```
/^([0-9a-zA-Z.-]+)@([0-9a-zA-Z.-]+)\.([a-zA-Z]{2,4})$/g
```

This will allow any 0-9 , a-z, A-Z, ., - at the beginning of the input follow by @ and any 0-9 , a-z, A-Z, ., - before dot and a 2-4 character at ending.

Password validation is used for password input and there is not validation of confirm password input since the confirm password checker will simply compare them.

Password validation is by regex format of:

```
/((?=.*\d)(?=.*[a-z])(?=.*[A-Z]))/g
```

This will make the Password having both digits or a-z and A-Z by positive look heads which acts like an and or &&.

Name validation is used for both last name and first name inputs. Because they should follow same rules. Name validation is by regex formats of:

```
/(^[A-Za-z \-' ]+$)/g
```

This specify that inputs can only contain A-Z, a-z, space, - in the name inputs.

Dob validation is a bit difference which not used of regex but string manipulation. The input will be first cut in pieces by separator “/”. There should be only 3 elements in array which indicates the day, month, and years. Then the day and month elements are further checked by length since day and month can not go over 2 digits but could be 1 digits and years can only be 4 digits. Then, the string will be parse into integer for integer comparison since days need to be between 1-31 and month need to be 1-12 and year can not go beyond 2021. Additionally, February is checked for any days that go beyond 29<sup>th</sup>.

There is also a validation of all info function that is called when submit, it run each checker function before submitting and return false(stop submission) when something is still wrong.