DOCUMENT OBJECT MODEL: EXERCISE COVER LECTURE 7

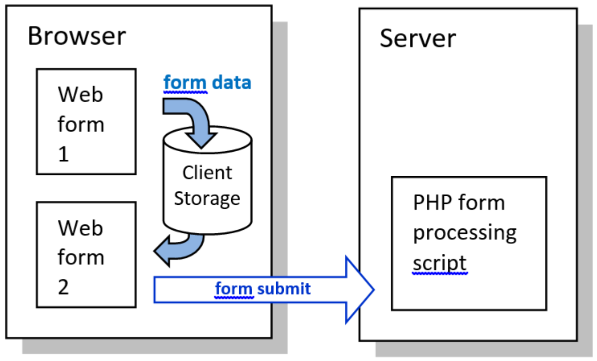
Note: This exercise **cover lecture 7**. The code is in the lecture 7 slides.

We are going to use lab 5 exercise files for this lab.

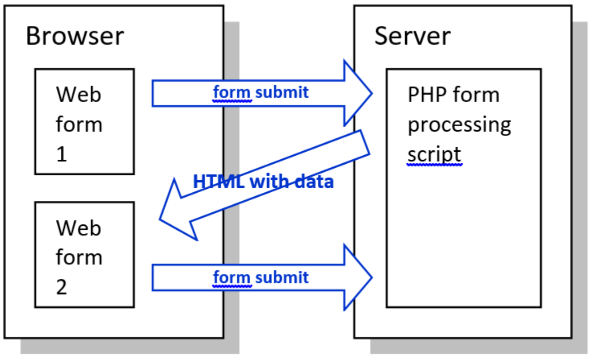
TASK 1: TRANSFERRING DATA BETWEEN HTML PAGES USING CLIENT WEB STORAGE

Sometimes we want to transfer information entered in a form on one Web page to another Web page. For example, for the **register.html** form we developed in lab5, we want to display a confirmation page for the user to accept before the data is submitted to the Web server.

There are a couple of methods for doing this.



Method 1 – Using Client Storage



Method 2 – Server-side processing

Method 1 transfers data the user has entered into their Form 1 via local client storage which is managed by the browser. This data set is then used to initialize the second form when it loads. After the user has checked it (possible provide extra information) this second form is then submitted to the server.

In Method 2, Form 1 submits the user data set to a PHP script on the server. The script then used to create a new HTML form page that incorporates the data the user sent to the server. This second form then completed by the user and submitted to the server.

In Web Storage data are stored in ***key/value*** pairs. There are two new objects for storing data on the client:

- **localStorage** - stores data with no expiration date
- **sessionStorage** - stores data for one session (while the browser window /tab is open)

In this Task we will use **sessionStorage**. The way to store a key/value pair is:

```
        sessionStorage.key = value;
```

For example, to store the value of an HTML element with `id = username` in JavaScript we could say:

```
        sessionStorage.uname = document.getElementById("username").value;
```

or alternatively

```
        sessionStorage.setItem("uname","username");
```

To retrieve a value from sessionStorage (or localStorage) and store it in a JavaScript variable we write:

```
        value = sessionStorage.key;
```

For example:

```
        var username = sessionStorage.uname
```

As explained above, we will send data from **register.html** to the Web page **confirm.html** via **sessionStorage**. The basic procedure is:

1. Set the form on **register.html** to hyperlink to **confirm.html** .
2. After the data on the **register.html** form have been successfully validated, write the value of the input fields or selected options to **sessionStorage**.
3. When the **confirm.html** window loads, read the values from **sessionStorage** and write these to the HTML using JavaScript.
4. The data can be written as text onto the HTML page, or can be set as values for input elements in a form. Data need to be written to form inputs if you want to forward these values onto the server when the user submits the form in **confirm.html**. These input elements can be hidden from the user if required.

STEP 1: MODIFY THE FORM ON **REGISTER.HTML** SO DATA WILL SEND TO **CONFIRM.HTML**.

1. Copy your **lab5** folder and rename it to **lab6**.
2. Then change the **lab5.html** to **lab6.html.**
3. Then create a new html file name **confirm.html.**
4. In **register.html**, change the action attribute of the form so that it hyperlinks to **confirm.html**.

```
<form id="regForm" method="post" action="confirm.html" novalidate="novalidate">
```

STEP 2: STORING USER DATA.

In this step we will write a function that stores the values the user has entered into the form in **sessionStorage**. For now we will just get two value which is the **firstname** and **email.** We will call this function at the end of the **validateForm().**

1. In **validation.js**, create a function called **storeBooking()** as shown below. As we already have variables for **firstname** and **email** in the **validateForm**() **function**, we will pass these to the function as parameters.

```
function storeBooking(firstname, email){
//get values and assign them to a sessionStorage attribute.
//we use the same name for the attribute and the element id to avoid confusion

sessionStorage.firstname = document.getElementById("vName").value;
sessionStorage.email = document.getElementById("vEmail").value;
//add other elements here
alert ("Firstname stored: " +sessionStorage.firstname); //added for testing
alert ("Email stored: " +sessionStorage.email); //added for testing
}
```

2. **Extend** the function to store the rest of the user information input in **sessionStorage.**

3. Invoke the **storeBooking()** function from the **validateForm()** function as shown below. We only want to do this if the validation returns **isAllOK** returns true.

```
function validateForm(){
"use strict";   //directive to ensure variables are declared
// code from lab 5

if (isAllOK){
      storeBooking(nameOK,emailOK)
}

      return isAllOK;
}
```

4. Load **register.html** and run. You can use a temporary alert box as shown above to test if values are being correctly stored in **sessionStorage**.

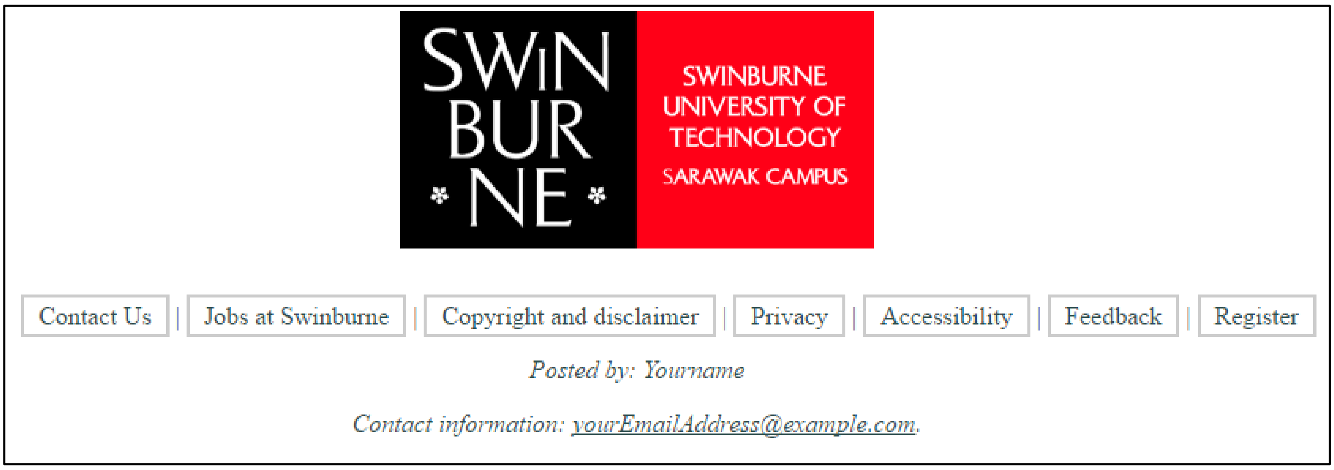
STEP 3: CREATE / MODIFY A WEB PAGE TO ACCEPT DATA

1. Now open **confirm.html** you create in step 1.
2. Add the following code to **confirm.html**

```
<!DOCTYPE html>
<html>
 <head>
    <meta charset="utf-8" />
    <title>COS10011 Creating Web Applications</title>

    <!-- link to my CSS -->
    <link href= "styles/jefferystyle.css" rel="stylesheet" />

 </head>
 <body>

<!-- navigation javascript -->
<script src="js/navigation.js"></script>

<!-- footer javascript -->
<script src="js/footer.js"></script>

 </body>
</html>
```

3. What we are doing here is creating a simple html files base on the design we have done in previous exercise. The reason for doing this is to make the page consistent through all the pages.
4. Please note that your code might different from the code above. Please modify it so it will be the same as yours.

5. Run your **confirm.html** and the page should look something like the picture below:


SWIN BUR NE
SWINBURNE UNIVERSITY OF TECHNOLOGY
SARAWAK CAMPUS
Contact Us | Jobs at Swinburne | Copyright and disclaimer | Privacy | Accessibility | Feedback | Register

Posted by: Yourname

Contact information: yourEmailAddress@example.com.

6. Next, create a new file name **confirm.js** and put it inside your JavaScript folder.
7. Next open **confirm.html** file and in the **<head>,** create a hyperlink which link to external **confirm.js.**

```
    <!-- confirm javascript -->
    <script src="js/confirm.js"></script>
```

8. Next, create a basic form to output all the user input from the **register.html**. In this example, I only store two value that is the name and email address.

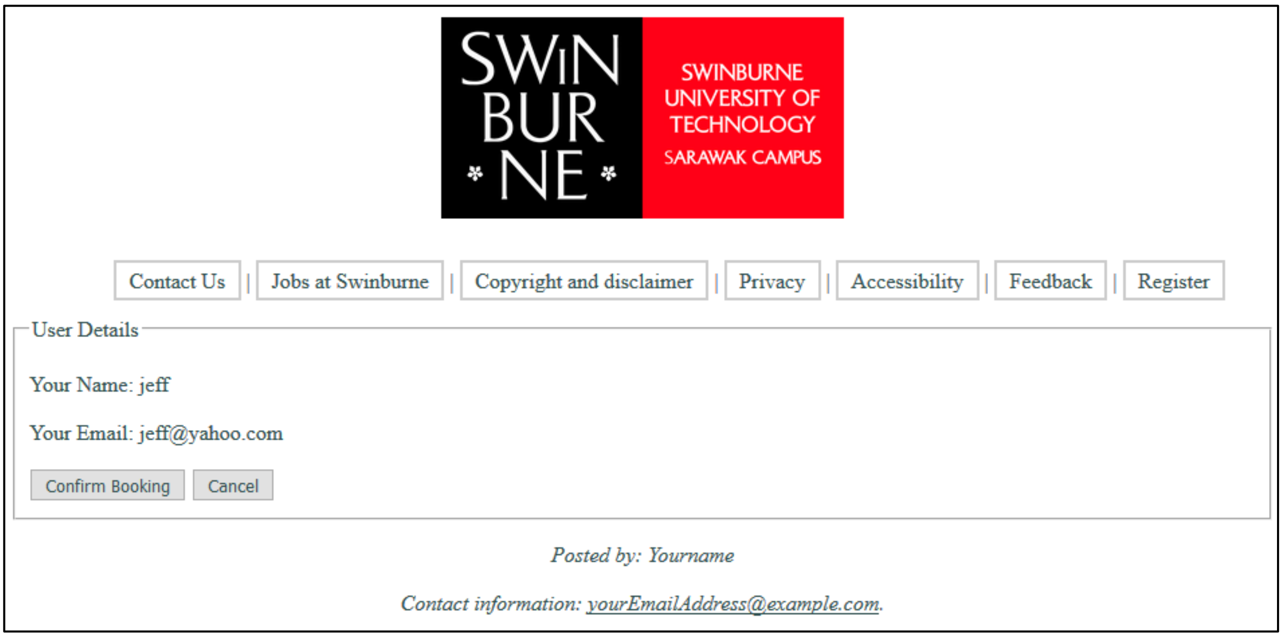
```
<form id="bookform">
    <fieldset>
        <legend>User Details</legend>
        <p>Your Name: <span id="confirm_name"></span></p>
        <p>Your Email: <span id="confirm_email"></span></p>
        <!--Create the rest of the data output here -->

        <input type="submit" value="Confirm Booking" />
        <input type="button" value="Cancel" id="cancelButton" onclick="cancelBooking()">
    </fieldset>
</form>
```

9. Next, continue creating all data that need to be store such as **Date of Birth, Your Unit, Your tutor, Issue, Description of Issue, Date** and **Time.**
10. All right. So we done with the **confirm.html**

11. Now open **confirm.js** file.
12. Add the following code:

```
"use strict";
/*get variables from form and check rules*/
function validate(){

    var errMsg = "";                                /* stores the error message */
    var result = true;                              /* assumes no errors */

    if(result){
        alert("Booking Successfully");
    }else{
        alert("Booking Failed");
    }
    return result;    //if false the information will not be sent to the server
}


function getBooking(){
    //if sessionStorage for username is not empty
    if(sessionStorage.firstname != undefined){
        //confirmation text
        document.getElementById("confirm_name").textContent = sessionStorage.firstname;
        document.getElementById("confirm_email").textContent =sessionStorage.email;
    }
}

function cancelBooking(){
    window.location = "register.html";
}

function init () {

    //call the getBooking() function.
    getBooking();

    // link the variable to the HTML element
    var bookForm = document.getElementById("bookform");
    var cancelBooking = document.getElementById("cancelButton");

    //check which button is being click.
    if(bookForm){
        bookForm.onsubmit = validate;
    }else{
        cancelBooking.onclick = cancelBooking();
    }
 }

window.onload = init;
```

13. Now test your registration form. Once you click **Register**, it will go to **confirm.html** page and you should see the result like the picture below:


SWIN BUR NE
SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS
Contact Us | Jobs at Swinburne | Copyright and disclaimer | Privacy | Accessibility | Feedback | Register
Confirm Booking
Cancel

User Details

Your Name: jeff

Your Email: jeff@yahoo.com

Posted by: Yourname

Contact information: yourEmailAddress@example.com.

14. All right, you done with the last tasks.
15. Once you done, zip all the files and submit through blackboard.