Swinburne University of Technology

COS10011/60004
Creating Web Applications

Assignment Part 2

Develop an Interactive Website

Important Dates:

Due Date	**10am on Monday in Week 8** (Late submission penalty: 10% of total available marks per day)
Demonstration	**Your tutorial, Week 8** (Demonstration is optional)

Contribution to Final Assessment: 30%

Note: ***You must meet the Essential Requirements of this assignment to be eligible to submit Part 3 of the assignment.***

Note: Do *not* use JavaScript libraries (e.g. jQuery) in the main part of this assignment. You may create an additional alternative implementation using a library as an enhancement (see enhancements section below).

Purpose of the assignment

In this assignment you will further enhance the website you developed in Assignment Part 1. In particular you will

- Use client-side storage to transfer data between pages.
- Use JavaScript to validate data entered into HTML forms and provide user feed back

Like Part 1, there will be an opportunity to optionally enhance your website beyond the basic requirements.

HTML

This part of the assignment requires minimal alteration to the HTML you wrote in Part 1.
All pages should be valid HTML5

CSS

All pages should be styled appropriately using CSS as in Part 1, and should be valid CSS3.
Minor additional CSS styling might be required.

If you wish to make other HTML and CSS alterations to your Assignment Part 1 that is OK (but you must keep your assigned job role).
Remember: You need to implement your website in standard HTML5.

Web Site Description

Data Validation

In Part 1 of the assignment you validated most of the inputs on the **`apply.html`** form using HTML5. In this part of the assignment we will use JavaScript to do some additional data validation,

in particular where the data entered into one field is validated against the value in another (e.g. postcode and state must be consistent) then this will need to be done in JavaScript.

Specific data validation rules ***in addition*** to those define in Part 1 are:

1. For the date of birth text field, a valid date must be entered in valid dd/mm/yyyy format. Applicants must be at between 15 and 80 years old at the time they fill in the form.
2. The selected state must match the first digit of the postcode
   VIC = 3 OR 8, NSW = 1 OR 2 ,QLD = 4 OR 9 ,NT = 0 ,WA = 6 ,SA=5 ,TAS=7 ,ACT= 0
   (e.g. the postcode 3122 should match the state VIC)
3. If the "Other skills..." is selected in the Skills Checkbox list, the Other Skills text area cannot be blank.
4. If the above data does not validate appropriately, meaningful feedback should be given to the user. Error messages should be displayed in an appropriate place ***on the Web page itself*** (rather than using an alert box).

Data transfer using Local and Session Storage

1. **`jobs.html`** page. Add an Apply hyperlink in each job description section. When the user clicks on this link they will be transferred to the application form page **`apply.html`** Using JavaScript, the Company's *position description reference number* (5 characters) will be stored using **local** client-side storage.
2. **`apply.html`** page. When this page is loaded, the job reference number) will be retrieved from local storage, and will be displayed as ***read-only*** in the form. This data value will also then need to be sent to the server, along with the other personal data the user enters into the form. *(Hint: Lab 7 shows how to use hidden input elements to transfer form data.)* While nothing will be stored on the server in this assignment (we will do this in Assignment Part 3), this process will allow the form data passing to be tested.
3. After a user has applied for one job, if they apply for another job *during the same browser session*, the browser should remember their details and automatically pre-fill the application form with the information about the applicant. Use session storage for this purpose.

Implementation of JavaScript

There should be **no** JavaScript embedded in your HTML files. This precludes both event registration (e.g. `<form onsubmit="return validate()" ...`) and function definitions in the HTML.

JavaScript should be in a file called **`apply.js`** located in a **`scripts`** folder.

Enhancements using JavaScript

You should complete the above requirements before attempting any enhancements.

As with Part 1 you have an opportunity to implement enhancements to your Web site using techniques not covered in the tutorials. Each enhancement must be described on a page called **`enhancements2.html`**. The entries on this page should:

- briefly describe the interaction required to trigger the event ***and*** what a programmer has to do to implement the feature.
- provide a hyperlink to the page where the enhancement is implemented in your Web site.
- reference any 3rd party contribution to the enhancement

It is a good idea to discuss your proposed enhancements with your tutor before you implement them.

The JavaScript enhancements themselves should be in a separate **`enhancements.js`** file. Make sure there are adequate comments to explain the enhancement (including its source if applicable).
Examples of JavaScript and other enhancements you might make include (but are not limited to):

- Have your jobs written in JavaScript and dynamically display the data in the jobs page.
- Use the JavaScript methods `querySelector()` that take a CSS selector as an argument to manipulate the web page in response to user action.
- Create an extra client side JavaScript dynamic effect: e.g. Slideshow, random image displayed onload, etc. The code and structure of this is open, but must be documented and explained as clearly as possible.
- On apply.html, implement a timer so that the user only has a limited time to complete the application after which a warning is displayed and the browser redirects to the home page.
- Use JavaScript to change the Menu display, to reflect the current page being viewed.
- *Re-implement* your JavaScript using a library such as jQuery. Add some enhancements the library provides. **No** library code should be included in your **`apply.js`** file. This alternative implementation should be in the file **`enhancements.js`** file. Explain the difference in approach using the library and using plain JavaScript.
- …

Any enhancements that are not listed and linked on the page **`enhancements2.html`** ***and*** implemented in **`enhancements.js`** will not be assessed.

***Up to 10 marks will be allocated to each enhancement. A maximum of 2* enhancements will be assessed.**

Web Site Folder Structure and Deployment Requirements

Your website folder structure should follow a similar structure as Assignment 1.
All files should be under a folder `/assign2`. JavaScript should sit in an `assign2/script` folder.

```
assign2/              You must have this folder – case sensitive!
  index.html
  jobs.html
  apply.html
  about.html
  enhancements.html
  enhancements2.html
  ...other html pages

  scripts/            Folder for your JavaScript
  images/             Folder for images for your page content
  styles/             Folder for style.css other css files
  styles/images/      Folder for images referred to by your css files e.g. background
```

Notes:

- HTML files should only be in the base "assign2/" folder – not anywhere else.
- All links to your files (JavaScript, CSS or images) should be ***relative***. ***Do not use absolute links***, as these links will be broken when files are transferred for marking. No marks will be allocated if links are broken.

Assignment Submission (Canvas + Mercury)

Your assignment should be uploaded to Mercury on or before your deadline.

An electronic copy of your assignment should be submitted through Canvas on or before your deadline.

- Make sure all your files are in the correct folders and compress your root folder with all your sub-folders with HTML, CSS, Javascript and images into a zip file named “assign2.zip”. Submit this to Canvas. When the zip file is decompressed, the entire website should be able to be run from index.html without needing to move any files.
- You can submit more than once through Canvas. Your last submission will be marked.
- Note that all deliverables must be submitted electronically. There is no need to submit an assignment cover sheet.

Make sure you complete your Canvas submission process.

Mark Sheet

Marked by: ……

Declaration: Fill this in before you start

I hereby confirm that the assignment to be demonstrated is identical to that I submitted to

Student number …….. Student name ……..

Signature ……. Date ……

Allocated job title ……..

Tutorial Day …… Tutorial Time …….. Tutor Name …….

Marker to Complete
Mercury date file check ☐ Days late penalty if applicable (10%/day) ……..

Essential Requirements

Tick box ☑ if requirement met

To meet the essential requirement s

- **`jobs.html`** and **`apply.html`** must be valid HTML5 running on mercury ☐
- when the **Apply** hyperlink is click on a job description, the *position description reference number* for that job must be transferred from **`jobs.html`** and **`apply.html`** via local storage. This number must be then transferred to the server when the form is submitted ☐
- at least one data input must be check as specified above using JavaScript and if invalid format and appropriate message displayed on the web page ☐

All essential requirements met	**Y/N**

If your assignment fails to meet these essential requirements you will need to make it compliant before submitting Assignment Part 3.

Total marks = 100

Requirements	**Mark**
Data transfer via local storage - job data stored from **`jobs.html`** ☐ (4) - job displayed in **`apply.html`** **as read-only** ☐ (6) - all data correctly transferred to server on form submit ☐ (10) - data pre-filled if second job applied for during session ☐ (20)	/40
Data format and range checking using JavaScript (10 marks each) - DOB in valid dd/mm/yyyy format with age between 15 and 80 ☐ - State and postcode match ☐ - Other skills text area not blank if check box selected ☐ - Appropriate error messages written to web page ☐	/40
Subtotal	**/80**

A maximum of 2 enhancements will be assessed if listed and linked from enhancements2.html. Up to 5 marks are available per feature. Poorly implemented or trivial enhancements may receive less or zero marks.

Enhancements **Name**	**Described**	**Linked to where implemented on your Web site**	**Source (if applicable)**	**Mark**
	Y/N	Y/N	Y/N/na	/10
	Y/N	Y/N	Y/N/na	/10
Sub-total				**/20**

*Deductions may be made during the demonstration or during code inspection **after** the demonstration.*

Requirement	**Deduction if not met**	**Deduct**
HTML5		
- Meta-data follows in-house standard	-2	
- Html has no Style information embedded	-2	
- HTML form elements follow in-house standard	-2	
- No deprecated elements/attributes used	-2	
JavaScript		
- All JavaScript is in an external file	-4	
- No 3rd party libraries used	**up to -10**	
- Header comments as per in-house standard	-2	
- Line comments as appropriate	-2	
Web site		
- All third party content acknowledged properly*	up to -100	
- Directory Structure as defined above	-4	
Total Deductions		

* Note: Failure to acknowledge third party code or content *at all* is plagiarism and may result in zero marks for this assessment or other penalties in accord with Swinburne policy.

A final assignment mark will *not* be provided during the demonstration. All code is inspected after the demonstration by your tutor before a final mark is allocated.

Comments: ……
……