

**HTML 5 Formative Assessment 2 (Internet Programming)**

***Unit Standards covered:***

* 115369 Design and build a web-site using simple HTML
* 115368 Apply advanced HTML and associated
* techniques to build a web site for business applications
* 115391 Demonstrate an understanding of the principles of the internet and the world-wide-web

**Instructions to Learner:**

1. Learner needs to include all criteria evidence of all tasks.
2. The work handed in must be your own and valid.
3. Please take note that failing to adhere to the criteria of each task within this assessment will lead to an immediate NYC.
4. The learner must achieve a mark of 80% to be deemed as competent

Learner Name and Surname: a 

Assessor Name and Surname: 

Learner ID Number: 

Date: 

# 1. Introduction

JavaScript is the most popular scripting language used to create web sites and enables web pages (HTML+CSS) to present a dynamic and interactive user interface.

Given your supreme abilities with HTML 5 and CSS, which you have obtained in the previous assessments and projects, you will now be required to use JavaScript to create a more interactive user experience, using your Previous project as a Template , you will now make it a generate Template Website(see <https://resume.io/create-resume> Example).

On completion, your assignment must consist of the following components:

* Person XML file;
* Login page;
* Gallery Page;
* About Page;
* Contact Page;
* Social Page;
* Home Page;
* Edit details Page;
* External CSS files;
* External JavaScript files

# 2. Constraints

1. You must complete this assignment individually.
2. You may ask your facilitator for help but they will not be able to give you the solutions.
3. You must produce all of the source files yourself; you may not use any tools to generate source files or fragments thereof automatically.
4. Your assignment will be tested using the latest version of Internet Explorer, Chrome or Mozilla Firefox.
5. All of your JavaScript and CSS code should be in external JavaScript and CSS files, respectively.
6. You may not use Light Box or any other JavaScript libraries. Only HTML 5, CSS, JQuery and pure JavaScript may be used. The use of any prohibited libraries will result in a zero mark for that particular task.
7. Please read the entire specification before starting work on this assessment. The reason for this is that later tasks may be easier to implement if you keep them in mind when working on earlier tasks.

# 3. Submission Instructions

You are required to upload all of your source files (e.g. HTML, CSS, JavaScript, Images etc.) to the VQ Manager. Make sure that you test your submission in the Visual Studio thoroughly. Once you are sure that everything works, archive all of the relevant files and upload them to the VQ Manager before the deadline.

No late submissions will be accepted so make sure that you upload in good time.

**4. Online resources**

**HTML5 - https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5**

**CSS - https://developer.mozilla.org/en-US/docs/Web/CSS**

**JavaScript – https://developer.mozilla.org/en/docs/Web/JavaScript**

**XML - https://developer.mozilla.org/en-US/docs/XML**

**JSON – https://developer.mozilla.org/en-US/docs/User:Waldo/JSON**

# Cookies - https://developer.mozilla.org/en-US/docs/Web/API/document.cookie

\*Please note that these resources are adequate places to start understanding some of the concepts covered in this assignment. It may, and probably will, be necessary to do further research during the completion of this assignment.

## 5. Rubric for marking

|  |  |
| --- | --- |
| **Task 1**  Well-Formed XML  Correct structure as specified | **5**  2  3 |
| **Task 2**  Login form (Inputs correct)  Parsing XML  Object construction  Saving cookie | **15**  2  5  5  3 |
| **Task 3**  Retrieving cookie  Elements dynamically created  Data displayed | **15**  5  5  5 |
| **Task 4**  Retrieve cookie & array  Display course stack  Sorting (ascending & descending) Animation of sorting | **30**  5  5  5  15 |
| **Task 5**  Carousel of images  Animated rotation of images  Enlarged image box  Data form validation (Fill form, JavaScript, Errors) | **30**  7  8  8  7 |
| **Task 6**  Visual appeal  Easy navigation | **5**  3  2 |
| **TOTAL** | **/100** |

**6. Assignment Instructions**

# Task 1: Create an XML file…………………………………….. (5)

For this task you must create an extensible mark-up language (XML) file to store the data that will be used in the web page.

The XML must have the following structure:

− Class

− Person

− Username

− Password

− Name

− Surname

− Image Path

− Age

− Degree

− Email

− History

-All additional Details Required for CV

The XML must adhere to the following requirements:

* XML must be valid and well-formed.
* Dummy data for at least 5 People must be entered within the structure.
* The XML document will be static and its contents will only be accessed by web pages.

# Task 2: Login Page, XML Parsing and Objects …………………. (15)

For this task you need to create a login page with the file name ‘index.html’. The page must be linked to its necessary style and script files.

The page must adhere to the following requirements:

* A form which contains input fields for a username and password must be provided.
* The characters in the password input field must be masked.
* Once the user clicks on submit, JavaScript must be used to authenticate the student trying to log in, i.e. to determine whether the provided username exists within the static XML file and to determine the validity of the provided password. This can be done as follows:
  1. Load the XML document into JavaScript using this code:

if (window.XMLHttpRequest)

{

xhttp=new XMLHttpRequest();

}

else // code for IE5 and IE6

{

xhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

xhttp.open("GET","nameOfFile.xml",false);

xhttp.send();

xmlDoc=xhttp.responseXML;

* 1. ‘XmlDoc’ is a variable in JavaScript that stores a representation of the data from the XML file. Using this ‘xmlDoc’ variable created in the code above, traverse through the XML structure to find the relevant student.
  2. If the person is not found or the password is incorrect, provide a descriptive error message. If the person is found, create a JavaScript object to store the details of the person. These details include the Name, Surname, Image Path, Age, Degree and Email
  3. Store the created object in a JavaScript cookie. For this, the object needs to be represented in the form of a string. To do this you might have to do some string manipulation, or use the easier alternative: JavaScript Object Notation (JSON).
  4. Navigate automatically to the home page: ‘home.html’.

# Task 3: Home Page, Cookies & JavaScript DOM ……………….. (15)

For this task you need to create a home page with the file name ‘home.html’. The page must be linked to its necessary style and script files.

The page must adhere to the following requirements:

* When loading, the page should retrieve the data stored in the JavaScript cookie created during Task 2 and convert the data back into a Student object.
* If no cookie data exists, the user should automatically be redirected to the login page.
* Once the Person object has been created, some of its general data must be displayed on the home page in a visually appealing manner. This data includes: Name, Surname, Age, Degree, Email and an Image.
* The HTML elements used to display this data must be created dynamically using JavaScript and appended to the ‘contents area’ of the home page. If these elements are hard coded into the HTML page rather than generated by the JavaScript, marks will be lost.
* The Image Path attribute must not be displayed in text, but should be used to display the image to which it is pointing.

# Task 4: Edit Details Page.………………………….. (30)

For this task you need to create an Edit Details Page with the file name ‘EditDetails.html’. The purpose of this page is to allow the user to change the details of their CV. Read all of the requirements before starting with this task as some of the requirements depend on prior requirements.

The page must adhere to the following requirements:

* When loading, the page should retrieve the data stored in the JavaScript cookie created during Task 2 and convert the data back into a Student object.
* Position all the blocks created above as a vertical stack in the contents area of the page.
* Provide two buttons, namely ‘Sort Ascending’ and ‘Sort Descending’.

# Task 5: View People page, Data forms validation, Carousel & Dynamic styling ……………………………………………….…..... (30)

For this task you need to create a page with the file name ‘class.html’, which should provide the functionality to view all the people using your website, as stored in the XML file, as well as adding a new person to the class.

The page must adhere to the following requirements:

* After the array has been created, it should be used to populate a carousel of the people images as specified by their respective Image Path attributes.
* The carousel should be set up as follows:

− One large image in the middle, and two slightly smaller images on either side of the main image.

− A left arrow and a right arrow.

− On load, the main image should display the logged in persons image.

− Only three of the people images will be displayed, so the other people’s images must be hidden.

− When an arrow is clicked, the carousel should gradually revolve (i.e. animate) into the correct direction. The image at the one end must gradually become hidden and at the other end, a previously invisible image must gradually become visible.

* Additionally, the general details of the person whose image is displayed as the main image must be loaded into editable input fields. These input fields must be part of a form which should be below the carousel. The form should also have a button to submit.
* The general details include Name, Surname, Age, Degree and Email.
* When the submit button is clicked, the values entered into the input fields must be validated using JavaScript. E.g. Email correct format or age must be a number.
* Empty or invalid fields should be indicated in some way.
* If the edited values are valid, the changes should be made to the XML Dom temporarily stored as a variable. In other words these changes do not need to be made to the XML file, but only to the internally stored XML structure.
* When the main image of the carousel is clicked, the entire page should be dimmed out and a larger sized image box must be displayed over the dimmed out contents. It should be possible to close the enlarged image to continue viewing the contents of the page.

# Task 6: Overall look, feel and ease-of-use ……………………… (5)

This task requires you to ensure that your website is easy to use and visually appealing.

You should focus on the following but you are not limited in what you may do:

* Web pages should be styled consistently.
* Color choices, font styles, and font sizes should take human visual perception into account (i.e. avoid poor contrasts like brown text on a green background, etc.)
* Ensure that your labels and buttons are either self-explanatory or employ some mechanism to provide more details (e.g. popup hints).
* Provide a navigation bar, menu, or some other tool of your choice to facilitate the navigation of your web pages.

**Enjoy!**

|  |  |
| --- | --- |
| **Total for assessment** | **100\** |
| **Total percentage for assessment** | **100\** |

**Competency:**

**Competent**

**Not Yet Competent**

**Final Comment**