BRITTS COLLEGE

ABN 87 056 634 541 | CRICOS Code: 04061K
Provider Code: 40330

STUDENT GUIDE

ICTWEB514 Create dynamic web pages

CONTENTS

Overview

Application of the unit

This unit describes the skills and knowledge required to produce both server and client-side content for web pages.

It applies to individuals working as web developers, who are responsible for creating dynamic pages to provide interaction between the user and the website. They use highly developed technical and analytical skills when developing the user-website interface.

No licensing, legislative or certification requirements apply to this unit at the time of publication.

Learning goals

Learning goals include:

- preparing to create dynamic web pages
- creating dynamic content
- testing and finalising dynamic web pages.

Topic 1: Creating dynamic web pages

Introduction

Before getting started we will explore some background concepts that relate to creating dynamic web pages.

A web browser's function is to act as a client to contact a web server and request information. But not all browsers act the same.

The difference between dynamic web site and a static websites

In basic terms a static website consists of a number of web pages with fixed content where each page displays the same information to every visitor. HTML code stays the same and the content of each page does not change unless manually updated by the web designer.

Static sites are the most basic type of website and are built by creating a few HTML pages and uploading these to a web server. This can work well for smaller websites or sites with a short life span and few content updates.

Interactive websites (or dynamic websites) on the other hand, are used for more complex websites, or ones where content is updated regularly.

A dynamic website with a content management system is used to allow users to interact with the website such as using a shopping cart or using an input form.

The code used to produce websites will determine whether it is dynamic or static. The dynamic functions of a web page can determine whether or not you use client-side or server-side scripting.

In order to develop a dynamic website, you would require:

- a server-side scripting language such as PHP
- a server environment such as Apache
- software for development such as an editor.
- a web browser.

Activity: Research and discuss

Look at the following website:

http://www.janewilliams.com/tuit.htm

- Is this website a static or dynamic site?

Compare the site to this one:

https://www.evoenergy.co.uk/uk-energy-guide/

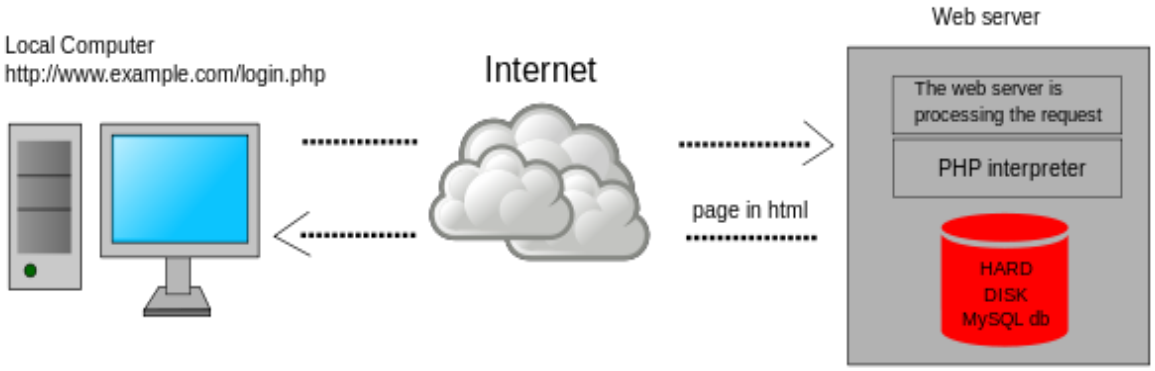
Client-side scripting –v- Server-side scripting

Client-side scripting is the code put into a standard HTML page and processed through the client to produce the output. The code can be viewed via code view window and a browser.

Client-side webpages can only contain client-side scripting for developing web pages.

Server-side scripting is the code put into a server side webpage such as a PHP page. The web browser will request a certain webpage and the page is then processed on the web server using an interpreter and then supplies data, images or any other code and merges it with your code that has been inserted within your web design of either CSS or HTML to produce the finished page.

Server-side code can't be viewed because the code is executed on the server and then delivered as HTML to the client.



Activity: Read

See how client-side and server-side scripting relates to creating dynamic web pages in the following wiki article:

https://en.wikipedia.org/wiki/Dynamic_web_page

Take any notes to summarise what you have read and keep for future reference.

Technical requirements for creating dynamic web pages

Whether you are dealing directly with a client or part of a web development team, you will need to obtain and review the technical requirements for creating dynamic web pages. This could include front end or user interface, coding requirements or storage and access.

The technical requirements may be obtained from a client by conducting a fact finding mission or these may just be provided by a project manager. This may be in the form of a report including the web server, database and scripting language requirements.

Undertaking website design should encompass the 3-tier architectural approach, which modularises the interface, business logic and data storage layers.

The three tiers are:

- Presentation (front end/user interface)
- Application (functions written in code such as JavaScript, PHP)
- Data (database or storage system and data access layer such as MySQL).

Let's look further at some of the technical requirements and how this relates to building dynamic web pages including hardware, software and languages.

Development server

To build a dynamic website you would need a web server consisting of an application server and database software.

Apache web server software

Apache is a widely used web server software application which is open source (available free of charge). The server is fast, reliable and secure and customisable to meet the needs of different environments. Most WordPress hosting providers use Apache.

Activity: Read

Look at the following development environments for PC and MAC:

XAMPP is a free install Apache distribution:

https://www.apachefriends.org/index.html

MAMP - local server for MAC:

https://www.mamp.info/en/

The following is a tutorial for setting up the PHP environment:

https://www.jetbrains.com/help/phpstorm/configuring-php-development-environment.html

Integrated Development Environment (IDE)

These environments offer features such as debugging, testing and integration. Dreamweaver is an example of an IDE, providing previews of changes for the front end and enabling visualisation of web content while coding.

Legislative and organisational standards and procedures

When designing and developing websites, there are a number of standards and legislation that an organisation will include in their organisational technical requirements for web development. For example:

- Copyright and intellectual property
- Standards for dynamic and static content
- Website accessibility
- Accuracy of content
- Website standards and principles
- Privacy and confidentiality
- Australian consumer laws
- Cyber security.

Activity: Watch

Watch the following video that explains the legal issues involved in developing a web application.

Video: https://legal123.com.au/how-to-guide/how-to-develop-an-app-infographic/

Make notes for future reference.

The organisational requirements, standards and procedures that a web developer follows will be dependent upon who you work for. The following are some general requirements:

Following a software development lifecycle such as SDLC for web development

A methodology adopted for the design and development of software projects. The methodology provides clearly defined steps and processes from the beginning to the end of a project. The process is meant to produce software with the highest quality and lowest cost in the shortest time. It includes a detailed plan on how to develop alter maintain and/or replace a software system. The SDLC can be roughly followed to achieve best practices in software design and development.

Work through the tutorial to develop your understanding:

https://www.guru99.com/software-development-life-cycle-tutorial.html

Word Wide Web Consortium (W3C) Standards

W3C develop the standards for the world wide web.

Complying with web standards can provide accessibility to software/machines and people, along with providing stability, consistency and compatibility.

The following link to their web design and applications standard page provides a number of further links for HTML and CSS and Accessibility:

https://www.w3.org/standards/webdesign/.

Accessibility

When designing and creating web pages, it is important that they are accessible to everyone. It should allow a user to navigate and interact with a site easily. The code must be as semantic as possible (using the correct HTML elements for their correct purpose) so that it is easy to understand for other users or screen readers.

In particular, due to the visual and dynamic nature of webpages, it is important to make sure that the website will also make sense to visually-impaired or blind users.

Read the following information from W3schools explaining HTML accessibility:

https://www.w3schools.com/html/html_accessibility.asp

CSS and JavaScript

These languages do not have the same immediate importance for accessibility as HTML, but they are still able to help or damage accessibility, depending on how they are used.

In other words, it is important that you consider best practices to make sure that your use of CSS and JavaScript doesn't interfere with the accessibility of your documents.

Following rules and standards for website design and functionality

For example:

- Easy navigation
- Customisable
- Site navigation made easy for the user
- Ability to work on multiple browsers and devices
- Consistency
- Quick load time
- Embedded privacy and security.

Using standard organisational software development processes

- Such as the waterfall model following the software development life cycle phases.
- Standards for coding.
- Templates, documentation and styles to provide consistency and standard practices.

Following coding standards for web applications

- Syntax and algorithm standards.
- Structure of code and commenting.

Activity: Read

Further information of legislation relating to web development:

https://legal123.com.au/how-to-guide/legal-guide-web-developers/

Information on website development and the law:

https://www.artslaw.com.au/information-sheet/website-development/

Website standards, guidelines (published by the Queensland government):

https://www.forgov.qld.gov.au/web-publishing-standards-and-guidelines

Standards W3C:

https://www.w3.org/standards/

Take any notes to summarise what you have read and keep for future reference.

Identifying dynamic content

As part of the web development, you need to identify what sections of the website will require client-side dynamic content and which require server-side dynamic content.

Client-side dynamic content

Web pages that change in response to an action within that web page, such as a mouse or a keyboard action, use client-side scripting.

Client-side scripts generate client-side content. Client-side content is content that's generated on the user's computer rather than the server. In these cases, the user's web browser would download the web page content from the server, process the code that's embedded in the web page, and then display the updated content to the user.

Scripting languages such as JavaScript and Flash allow a web page to respond to client-side events.

Server-side dynamic content

Web pages that change when a web page is loaded or visited use server-side scripting. Server-side content is content that's generated when a web page is loaded. For example, login pages, forums, submission forms, and shopping carts, all use server-side scripting since those web pages change according to what is submitted to it.

Scripting languages such as PHP, ASP, ASP.NET, JSP, ColdFusion and Perl allow a web page to respond to submission events.

Source: https://www.doteasy.com/web-hosting-articles/what-is-a-dynamic-web-page.cfm

Which language and technology should be selected?

In most cases the language and technology will be dependent upon:

- purpose of the website
- the client request
- skill, knowledge and expertise of the web dev team
- resources available
- equipment, hardware and software availability/cost
- budget.

Languages and structures

Programming control structure

Program control is how a program organises its activities and makes decisions. This involves executing particular code based on a user input or the outcome of a prior operation.

The flow of control through any given function is implemented through three basic types of control structures:

Sequential:
Default mode. Sequential execution of code statements, one line after another is like following a recipe.

Selection:
This is used for decisions, branching, which is choosing between 2 or more alternative paths. In C++, these are the following types of selection statements: IF IF/ELSE SWITCH

Repetition:
This is used for looping, which is repetition of a piece of code multiple times in a row. In C++, there are the following three types of loops: WHILE DO/WHILE FOR

Programming Design Structure

There is a standard way of organising the component structure, data structure, and control structure for a single-user application (used by one person at a time), which normally reads data, saves it in a data structure, computes on the data, and writes the results. This is the following:

- Design the program's component structure with three components. This must be organised in a model-view-controller pattern.
- Decide what form of data structure (array, table, set, list, tree, etc.) the program's data will be held with. The data structure will be inserted into the program's model component.
- Write the algorithm that defines the execution steps: which is the control structure. The algorithm will then be placed inside the program's controller.
- Determine the form of input and output (disk file, typed text in a command window, dialogs, a graphical-user interface, etc.) that the program uses. This will be embedded in the program's view.

Object-Oriented Programming

This is a programming technique constructed around objects rather than procedures, and these objects can work with each other.

Attributes and behaviours can be given to the objects. This means that developers can then focus on programming how the objects work with each other. It is used for making websites dynamic.

Activity: Tutorial

The following clip explains the four pillars of object-oriented programming.

Video: https://www.youtube.com/watch?v=pTB0EiLXUC8 (07:00)

Object-Oriented Programming Languages

The best languages used for dynamic website development include:

- Java.
- Python.
- Ruby.
- C++.
- Smalltalk.
- Visual Basic .NET.
- Objective-C.

Some concepts:

Object:

An object is a software bundle of related state and behaviour. Software objects are often used to model the real-world objects that you find in everyday life. This lesson explains how state and behaviour are represented within an object, introduces the concept of data encapsulation, and explains the benefits of designing your software in this manner.

Class:

A class is a blueprint or prototype from which objects are created. This section defines a class that models the state and behaviour of a real-world object. It intentionally focuses on the basics, showing how even a simple class can cleanly model state and behaviour.

Inheritance:

Inheritance provides a powerful and natural mechanism for organizing and structuring your software. This section explains how classes inherit state and behaviour from their superclasses, and explains how to derive one class from another using the simple syntax provided by the Java programming language.

Interface:

An interface is a contract between a class and the outside world. When a class implements an interface, it promises to provide the behaviour published by that interface. This section defines a simple interface and explains the necessary changes for any class that implements it.

Package:

A package is a namespace for organizing classes and interfaces in a logical manner. Placing your code into packages makes large software projects easier to manage. This section explains why this is useful, and introduces you to the Application Programming Interface (API) provided by the Java platform.

Source and further information from: https://docs.oracle.com/javase/tutorial/java/concepts/

Stateless programming

With stateless programming, the operations that are implemented are not sensitive to the state of the computation. All the data used in an operation are passed as inputs to the operation, and all the data used by whatever operations invoked that operation is passed back as outputs.

The program must have value semantics as it is not permitted to modify shared/aliased data structures, and objects do not have an identity.

They must not use global or class variables, and all of the input s and outputs must be handled specially, through monads or by threading an I/O state through any parts of the computation that perform I/O.

This means that when a program is stateless it cannot take the information about the last session into the next, something that a dynamic web pages requires.

Dynamic Content (Client-side)

DHTML (Dynamic HTML) is required to create interactive websites on the client side. It encompasses:

- HTML
- CSS
- SCRIPTS (JavaScript being the most common).

Note: JavaScript is a client-side script, and PHP is a server-side script.

PHP: Hypertext Pre-processor

PHP is an open source server-side scripting language that can:

- generate dynamic page content
- create, open, read, write, delete, and close files on a server
- collect form data
- send and receive cookies
- add, delete, modify data in a database
- used to control user-access
- encrypt data.

It runs on various platforms and supports a range of databases.

Activity: Practical

Use the tutorial from W3Schools to learn more about PHP.

In particular, look at input forms like a contact form that can be completed by a user and submitted:

https://www.w3schools.com/php/php_intro.asp

The trainer/assessor will assist with demonstrations and tutorials.

MySQL (SQL: Structured Query Language)

SQL is a standard query language used for interacting with databases. MySQL is a relational database management system. The database software is a client/server system consisting of a multithreaded SQL server that supports:

- different back ends
- different client programs and libraries
- admin tools
- application Programming Interfaces (API).

For creating dynamic web pages, a database is required: PHP complements MySQL. PHP collects the data and MySQL stores the information. For example, PHP can be used to create a shopping cart and then MySQL used to keep the data in a format PHP can use to generate an order status.

Editors

Text editors can be used for coding in HTML, PHP and JavaScript, however there are program editors with features to support the coding such as Notepad++ and HTML Kit.

Activity: Read

Use the tutorial from W3Schools to learn more about SQL and data storage and manipulation:

https://www.w3schools.com/sql/

Activity: Tutorials

There are a number of tutorials for using HTML along with JavaScript and CSS.

PHP: https://www.w3schools.com/pHP/default.asp

HTML: http://w3schools.sinsixx.com/html/html_intro.asp.htm

HTML5: https://www.w3schools.com/html/html5_intro.asp

CSS: http://w3schools.sinsixx.com/css/css_intro.asp.htm

HTML DOM: https://www.w3schools.com/js/js_htmldom.asp

There is also a tutorial for DHTML if you already have a basic understanding of HTML, JavaScript and CSS: http://w3schools.sinsixx.com/dhtml/default.asp.htm

Activity: Practical

For the purpose of this unit, it is important that you have an understanding of the fundamentals, in particular control structures used for sequence, selection and iteration.

Combining the languages, scripts and code will enable the development of the dynamic web pages.

Make sure that you understand the following:

- Understanding relational database structures
- PHP

- MySQL
- JavaScript
- DOM
- Testing and validation.

You will need to source tutorials that cover:

- Create a PHP script
- Creating a database table
- Connect the PHP file to the database
- Test the connection for errors

Ensure you cover:

- Arrays
- Printing data
- WHILE loops

The trainer/assessor will assist with demonstrations and tutorials using the software that the college has available.

Topic 2: Creating dynamic content

Creating boilerplate templates

As discussed in the previous topics, standardisation is common practice in web development. It provides consistency, structure and can link to compliance with legislation and organisational requirements.

Boilerplate templates are common practice used for coding, such as in HTML. You can identify what is required for a boilerplate according to the technical requirements and create code that will save time with repeating sections.

Activity: Read

Read further about the HTML5 Boilerplate:

https://html5boilerplate.com

Boilerplate in practice: A practical walk through:

https://www.unitily.com/articles/boilerplate.html

Take any notes to summarise what you have read and keep for future reference.

For your assessment you will be required to create dynamic web pages for a client. Within the website will be a form that users will have to complete contact details, along with a calendar to select the best day and time for an appointment, and an input field for comments.

- The form must perform validation for the user
- Contain a form handler script
- Redirect the user to another page when the form has been validated and submitted

Therefore, you will need to understand how to create a database that will store the information.

Activity: Research and discuss

What hardware and software do you have access to?

What language, script and platforms will you use?

Data Storage

Most websites store data that can be accessed at a later date, most commonly using a relational database to store the data. This can include using:

- Data Access Objects (DAO)
- PHP Data Objects (PDO).

It's the integration of the database with the website design that is the tricky part. Data can be simple in structure or complex.

Activity: Read

Objects -v- data structure: The following article could be used to explain the differences:

https://stackoverflow.com/questions/23406307/whats-the-difference-between-objects-and-data-structures

Take any notes to summarise what you have read and keep for future reference.

Activity: Create

Create a simple database that will hold contact information for a form, with CustomerFirstName, Email, Telephone and UserName and login details.

Create at least two tables.

The trainer/assessor will facilitate a tutorial.

Creating dynamic web pages

HTML5 is the new standard and will most probably be the language used for your assessment.

Activity: Research and discuss

Now create a simple page using HTML or your chosen script language.

The page should instruct a user to enter information that needs to be captured. For example, username and login details.

Use any tutorials to support your learning.

The trainer/assessor will facilitate a tutorial on using HTML (or your chosen script language) to create the page. You can look up the tutorial for HTML using the links provided earlier.

Now let's look at some of the design and functionality that you will need to consider when developing dynamic web pages.

Design features

Good design and a good user experience go a very long way toward making a site more accessible.

For example, many websites are overly complicated, and most people find the web difficult to use, at least sometimes, you've probably filled out online forms that have several fields, many of which seem unnecessary. A shorter and simpler form is not only a better user experience for all your users, it's also less functionality that needs to be made accessible.

Some further design features include:

- Navigational menus
- Navigation bars
- Quality web content
- Physical file structure
- Page design including dimensions, headers/footers, text-v-image layout, media used, colour and style, balance and emphasis.

Website functionality

Website functionality is the interactive aspect of the site – allowing the visitor to respond in some way. Types of functionality include:

- using a shopping cart
- contact form/appointment booking
- database form
- credit card processing
- email link
- private client access areas
- social media integration
- file uploads/downloads.

Some basic rules of website design and functionality:

Easy navigation	Customisable	Site navigation made easy for the user	Ability to work on multiple browsers and devices
Consistent	Quick load time	Embedded privacy and security	

Interface

This refers to the way in which the user interacts with the website and how the control structures in the code will function within the website to assist the interaction.

These elements could be:

- navigational components such as search fields, anchors, tags or sliders.
- input controls such as drop-down menus, action buttons or check boxes.
- information components such as alt text, tooltips or progress bars.
- containers used for layout and structure.

The interface could be functional or non-functional. Before the site is to be designed, it is important to know the amount of user-interaction required with the website and how the control structures will be used.

The design of the interface should create consistency, commonality, ease of use and be user-friendly. There are a number of 'best practices' out there, but here are a few common tips:

- The interface should be consistent
- Common user interface elements should be used
- Page layout should be structured for placement of important information and readability
- Colour and changes in text can direct attention to important areas on the page
- Use of varying font sizes, styles and arrangement can increase readability if used in the right way
- Communicate to the user using altext or tips for example.
- Input forms should use validation to support the user interaction (also can fall under usability).

Usability

This is the measure of the quality of a user's experience in using a website. The components of usability is variable on the type of website, however some common aspects are:

- Is the site easy to use?
- Is it easy to perform tasks if required, i.e., like completing a form.
- How memorable is the site if a user returns after a period of time?
- Does the site have a pleasing design?
- Does the site require a user to learn its functions?
- Who is the audience?

Usability is part of the planning, analysis and design of a website and should be vigorously tested and feedback used to improve the user experience.

Security

Web server security protects information assets that can be accessed from a web server. It is important for anyone who has a physical or virtual web server that is connected to the Internet. It requires defence that has several layers.

The purpose of website security is of course to prevent any attacks or damage, destruction, unauthorised access or major disruption for an organisation hosting a website.

It includes:

- Configuration of the web server,
- Policies for creating and renewing passwords,
- The client-side code.

Common threats include cross-site scripting, SQL injections, Denial of service (DoS), Clickjacking.

Authentication

Authentication is used by a server when it needs to know who is accessing their information or site. It is used by a client when they need to know that the server is the system that it claims to be.

With authentication the user, or the computer, has to prove its identity to the server or client. Authentication by a server usually entails the use of a username and password. Authenticate can also be done through voice recognition, retina scans, cards, and fingerprints.

Authentication usually involves the server giving a certificate to the client which states that a trusted third party (such as Verisign or Thawte) has verified that the server belongs to the entity that the client expects it to.

Authentication does not determine what files the individual can see or what tasks the individual can do. Authentication merely identifies and verifies who the person or system is.

Hypertext transfer protocol (HTTP)

Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, for example HTML. It was originally designed for communication between web browsers and web servers, but it is also used for other purposes. HTTP follows a standard client-server model: the client opens a connection to make a request, then waits until a response is received. HTTP is a stateless protocol, which means that the server does not keep any data (state) between two requests.

Session Management

A web session is a sequence of network HTTP request and response transactions that are associated to the same user. Complex web applications require that information or status about each user is retained for the duration of multiple requests. Sessions provide the ability to establish variables, such as localization settings and access rights, which apply to every interaction the user has with the web application for the duration of that session.

Web applications can create sessions to keep track of anonymous users after the first user request, such as the user language preference.

Web applications will make use of sessions once the user has authenticated, which ensures the ability to identify the user on any subsequent requests.

They make it possible to apply security access controls and authorised access to the user private data. Current web applications can provide session capabilities both prior to, and following, authentication.

Activity: Research and discuss

Using the HTML page, created earlier, design a simple form that could be used to capture the user information such as contact details or login information using HTML and CSS.

Ensure to consider interface design, usability, accessibility, standards and incorporate validation on the forms, for example do not allow blank fields or integers in a text field.

How does Alt-text provide accessibility? Use an example to illustrate your understanding.

How is ARIA used for accessibility? Use an example to illustrate your understanding.

The trainer/assessor will facilitate tutorials and demonstrations.

Topic 3: Testing and finalising

One of the main activities of web page design and development is testing. This is carried out throughout development and can be for front end elements (i.e., content, links) and back end elements (database and functions used in the code). It includes testing for errors, using debugging software, validating and browser testing and validating functionality of the web pages.

Debugging

Debugging code is a cyclic process and requires an understanding of the debugging software being used as this helps to work through any issues with the code and is used to determine the cause of any errors.

Activity: Research and discuss

Read through the following tutorial on JavaScript debugging:

https://www.w3schools.com/js/js_debugging.asp

What is the debugging tool used for the software you are using to develop your web pages?

Run the debugger and identify any errors and how they can be fixed.

The trainer/assessor will assist with the tutorials and help with the debugging codes and errors.

Cyber security protocols and procedures

Ensuring that the cyber security protocols and procedures are in place and working can help to ensure that the organisation is safe from any attacks, breaches or other critical incidents.

Some examples of these are:

- Firewalls
- Encryption
- Authentication
- Using an incident response protocol
- Training and education

The primary reason for testing the security of a web is to identify potential vulnerabilities and subsequently repair them.

- Network Scanning
- Vulnerability Scanning

- Password Cracking
- Log Review
- Integrity Checkers
- Virus Detection.

Following are some of the test cases for web security testing:

- Test by pasting internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing URL options directly. I.e. If you are checking some publisher site statistics with publisher site ID= 123. Try directly changing the URL site ID parameter to different site ID which is not related to the logged in user. Access should be denied for this user to view others stats.
- Try some invalid inputs in input fields like login username, password, input text boxes etc. Check the system's reaction to all invalid inputs.
- Web directories or files should not be accessible directly unless they are given download option.
- Test the CAPTCHA for automating script logins.
- Test if SSL is used for security measures. If it is used, the proper message should get displayed when user switch from non-secure HTTP:// pages to secure HTTPS:// pages and vice versa.

All transactions, error messages, security breach attempts should get logged in log files somewhere on the web server. This can be kept as confirmation of the security tests that have been carried out.

Testing web page functionality and content

Testing a website can include:

Testing code is a continual and cyclic process that occurs both as the programmer is coding and also to finalise sections of code and then finally interacting with the front-end development.

As mentioned, testing can be for front end elements and back end elements. The testing therefore should include the interaction between these two elements.

A component test can include:

- whether a search function works
- if a shopping cart works
- a database query.

Component testing tests each component in isolation (also known as module or program testing).

When the component is tested, sample data is used to validate, using various testing variables, upper and lower limits, and ranges.

It will also include colours, styles, formats and creativity of these elements.

Dynamic page testing

This involves the process of connecting all the components of the web project and testing that they integrate and work. This can include:

- integration into internal and external systems and applications
- integration of the interface.

Activity: Read

Refer to the following article which explains how the backend and front end interact:

https://hackernoon.com/in-simple-terms-backend-code-frontend-code-and-how-they-interact-2485c5a1bbd2

Take any notes to summarise what you have read and keep for future reference.

Functionality Testing

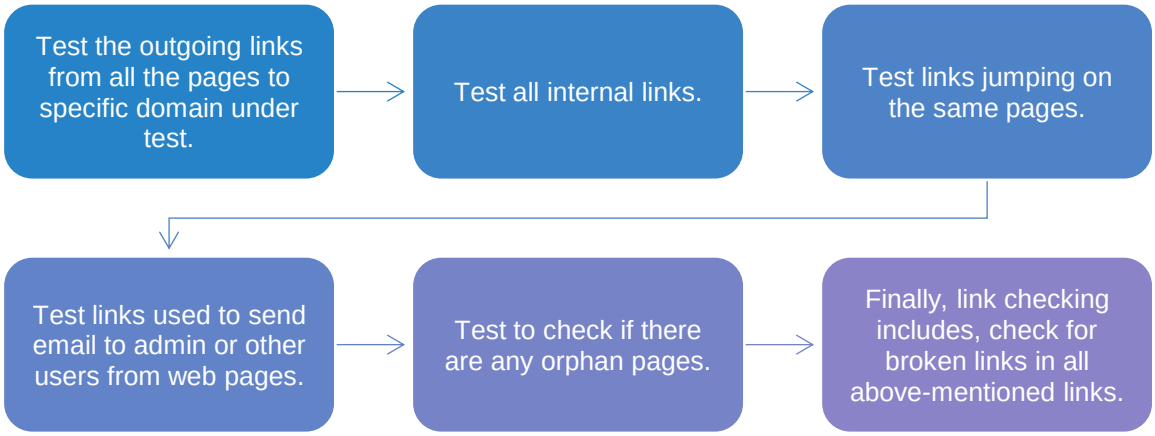
Functionality testing can be quite complex, particularly for large websites. It can include tests for all the links in web pages, database connections, forms used for submitting or getting information from the user in the web pages, cookie testing etc.

In testing the functionality of web pages, the following should be tested:

Links:

- Internal Links
- External Links
- Mail Links
- Broken Links.

For example, checking all the links can involve:



Forms:

Forms are an integral part of any website. Forms are used for receiving information from users and to interact with them. So, what should be checked in these forms?

- First, check all the validations on each field.
- Check for default values of the fields.
- Wrong inputs in the forms to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.

Database:

Testing for database integrity. Data consistency is also very important in a web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality.

Check if all the database queries are executing correctly, data is retrieved and also updated correctly. More on database testing could be a load on DB, we will address this in web load or performance testing below.

Note: You will not be required to carry out all of the testing outlined below, however it is good to have an understanding of all the various tests available and how it would affect your site if it went live.

Cookies Testing

Cookies are small files stored on the user machine. These are basically used to maintain the session-mainly the login sessions. Test the application by enabling or disabling the cookies in your browser options.

Test if the cookies are encrypted before writing to the user machine. If you are testing the session cookies (i.e., cookies that expire after the session ends) check for login sessions and user stats after the session ends. Check effect on application security by deleting the cookies.

Usability Testing

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

Usability testing includes the following:

- The website should be easy to use.
- Instructions provided should be very clear.
- Check if the instructions provided are perfect to satisfy its purpose.
- The main menu should be provided on each page.
- It should be consistent enough.

Test for navigation

Navigation means how a user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.

Content checking

Content should be logical and easy to understand. Check for spelling errors. Usage of dark colours annoys the users and should not be used in the site theme.

You can follow some standard colours that are used for web page and content building. These are the commonly accepted standards like what I mentioned above about annoying colours, fonts, frames etc.

Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.

Interface Testing

In web testing, the server-side interface should be tested. This is done by verifying that communication is done properly. Compatibility of the server with software, hardware, network, and the database should be tested.

The main interfaces are:

- web server and application server interface
- application server and Database server interface.

Compatibility Testing

Compatibility of your website is a very important testing aspect. This includes:

- browser compatibility
- operating system compatibility
- mobile browsing.

Browser compatibility

Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with.

Your website coding should be a cross-browser platform compatible. If you are using java scripts or AJAX calls for UI functionality, performing security checks or validations then give more stress on browser compatibility testing of your web application.

Test web application on different browsers like Internet Explorer, Firefox, Netscape Navigator, AOL, Safari, Opera browsers with different versions.

Regardless of the languages that you use to develop dynamic web pages, you should still make sure that they are validated across all browsers.

OS compatibility

Some functionality in your web application is that it may not be compatible with all operating systems. All new technologies used in web development like graphic designs, interface calls like different API's may not be available in all Operating Systems.

Mobile browsing

Test your web pages on mobile devices for any compatibility issues.

Performance testing

The web application should sustain to heavy load. Web performance testing should include:

- Web Load Testing
- Web Stress Testing.

In web performance, testing website functionality on different operating systems and different hardware platforms is checked for software and hardware memory leakage errors.

Performance testing can be applied to understand the web site's scalability or to benchmark the performance in the environment of third-party products such as servers and middleware for potential purchase.

Validation

Validation is also a testing tool and an important step to ensuring the technical quality of web pages. W3C validation is the process of checking a websites code to determine if it follows the formatting standards.

If you fail to validate the pages of your website based on W3C standards, your website will most likely suffer from errors or poor traffic owing to poor formatting and readability.

Why a website should be validated:

- It helps to improve search engine rankings
- Teaches best practices in website design

- Improves user experiences
- Website is browser friendly (across a number of browsers)
- Available over devices
- Ensures code is properly formatted
- It is used as a debugging tool.

Validating inputs

Validation will also occur when designing user inputs. For example, a form with date of birth should only be a date in the format DD/MM/YYY.

Types of validation can be:

- data type validation
- email address verification
- checking for blank fields
- using event handlers to flag incorrect submissions.

Activity: Read

See the W3Schools tutorial on form validation:

https://www.w3schools.com/js/js_validation.asp

HTML validator checks the markup validity of web documents produced in HTML:

http://validator.w3.org/

CSS Validator checks the CSS validity of web documents in HTML:

http://jigsaw.w3.org/css-validator/

Take any notes to summarise what you have read and keep for future reference.

Points to be considered while testing a website

The websites are essentially client/server applications – with web servers and 'browser' clients.

Consideration should be given to the interactions between HTML pages, TCP/IP communications, Internet connections, firewalls, applications that run in web pages (such as applets, JavaScript, plug-in applications) and applications that run on the server side (such as CGI scripts, database interfaces, logging applications, dynamic page generators, asp, etc.).

Additionally, there are a wide variety of servers and browsers with various versions of each. They include small but sometimes significant differences between them in terms of variations in connection speeds, rapidly changing technologies, and multiple standards and protocols. The end result of which testing for websites can become a major ongoing effort.

Activity: Practical

Using the HTML page, created earlier, create a test plan that could be used to test the form you created. The test plan should provide a list of tests and the outcome. Remember you will need to test a number of factors, within the form as well as user testing.

Carry out tests that will check:

- Form layout – is it easy to understand/use?
- Form fields – does the validation work on a number of entry variations? For example if you enter the wrong date range or date format what happens? What happens if you leave a field blank? What happens if a field contains integers instead of characters?
- Required fields – check if mandatory fields are completed and then the submit button works.
- Validation (input, messages and any other elements such as email function works)
- Tool tips
- Confirmation of page submission
- Browser testing in at least three different browsers.

Document all tests and results.

Send the outcomes in a draft email to the project manager/client (trainer/assessor) for feedback. The text of the email should be in grammatically correct English, written in an appropriate (polite, business-like) style.

Completion

After testing and validation, you need to evaluate the web page functionality results and make any amendments or modifications as required.

This may be in the form of a demonstration to show that the functions have been implemented according to the initial requirements; or further tests carried out by the project manager or client for further feedback.

Feedback could be from the client or within the software development team.

As in all projects, milestones should be used so that the client can sign off on various aspects of the project. If no feedback is received till the very end this could lead to a project failure.

The feedback obtained is used to make any changes. Any amendments or modifications should then be incorporated and the testing and documentation updated. This may be a cyclic event that may take several iterations until the web page is fully functional and approved by the appropriate people for deployment.

Once testing and validation is complete the web page should be finalised and sign-off obtained from relevant personnel such as a project manager or client.

This may include completing project documentation created during web page development, as well as any standard organisational documentation. It could include testing plans, user documentation, user interface changes, end-user documentation, and lesson's learned reporting.

Once this has been carried out you may also need to perform any project finalisation activities for the close of the project.

Activity: Practical

Refer back to your testing and if you receive any feedback that requires changes, incorporate these into the final web design. Include changes to any related documentation.

You will then submit to the client via an email (the trainer/assessor) to obtain approval for the completion of the project.

The text of the email should be in grammatically correct English, written in an appropriate (polite, business-like) style.