Dynamic Web Development Report – by Connor Duncan

Contents:

1. Background

The website that has been designed is based on the Scotland national Football team. It was created with a navbar that allows the user to find out information about the national team, and navigate the website. The website also makes use of a login feature that allows the user to access their own private account. The website includes many different features; These features make use of many different types of code, including JavaScript, PHP, CSS and more.

This report will be going over the website design, how it was created, and using what coding languages for each page; as well as why certain features are included in the website. Followed by legal issues that websites must comply with, and potential security risks that can occur for websites, and how to mitigate these risks. This report will conclude with how well the website matched the assessment brief.

2. Website Design

The website includes 5 different pages, these pages are as follows:

1 – Home Page

2 – Results and fixtures

3 – Gallery

4 – My Account/ Admin

5 – Login/log out

The pages that the user can access changes depending on whether the user is logged in to an active account or not. The navbar is a feature at the top of every page, that allows the user to easily navigate between the websites pages. It also includes a photograph of Scotland's national stadium – Hamden Park.

Navbar

The navbar dynamically changes depending on where the mouse is located. It uses yellow writing, with a blue background so that the text is clearly visible, and as the user scrolls across each option, the text changes to blue writing on a yellow background. This was created by using CSS, and the hover feature. If the user is not logged in, then there will be a login option displayed. The my Account, and logout options will not be displayed at this point in time. Once the user is logged in, the navbar will change to display the My Account page, and Logout option.

Home page

The Homepage makes use of JavaScript, AJAX, HTML, PHP and CSS. The website displays a current news page with 3 different news articles available. As the user scrolls over the articles with the mouse, the image located on the right uses JavaScript to change to an image related to the news article. If the user clicks on a news article to see the contents, then a JavaScript function is called. A paragraph element is then created to store the contents of the article. The contents to put inside the paragraph are then fetched using AJAX. It uses an XML HTTP request, to obtain a text file from the server. The contents of the text file are then placed inside the paragraph element. The paragraph uses CSS to have a white background with black text. This way the black writing does not contrast with the dark blue background of the website, and allows easy viewing for the user. One of the functions also calls an AJAX jQuery XML HTTTP request, instead of standard AJAX.

Bootstrap is another feature included in the website. By including bootstrap it allows the website to be used on multiple devices, and screens of different sizes. On the home Page, without bootstrap if you were to reduce the screen size then the news articles would overlap with the image, making them unreadable, and you would not be able to fit the whole navbar on one page. With bootstrap, the text moves so that it is alongside the image, rather than overlapping it, meaning the whole news article is still readable. With the navbar, as the screen size decreases, the navbar will place the page buttons on top of each other, instead of vertically, to keep the buttons all displayed on one screen. Therefore, by including bootstrap we avoid everything not fitting on the one screen.

Results and Fixtures

The results and fixtures page shows all of Scotland's major tournament campaigns, dating back to the year 2000. It has another menu bar running parallel to the navbar, and allows the user to pick and choose what campaign they would like to view. Upon choosing a campaign, the current table being displayed is updated. Using jQuery, the DOM is manipulated to update the table with the new content that corresponds to which button was clicked. This was created using two different functions, one of which creates a table

displaying how the campaign finished. The other function creates a table displaying all the results for that campaign, home and away. The table uses abbreviated terms such as GA and GD. For users who are unsure of what these mean, when they hover over the term, it comes up with a title displaying the full term, such as goals against, and goal difference. Once the user has clicked on the campaign, it also updates a text box in the top left corner, to remind the user what campaign they are viewing, this is also changed using jQuery. Sticking with the theme, when the user hovers over a row in either table, the writing changes from yellow writing with a blue background, to blue writing with a yellow background. By including this it enhances the user experience.

Gallery

The gallery page contains several photos and videos, each with captions to explain when the photo was taken, or at what event. The gallery displays the photos by making use of a jQuery plugin. By using this plugin, it allows you to scroll through the photos, and zoom in and out. You are also able to play videos from YouTube through the plugin.

Login

The login page allows the user to login to their account. Upon pressing the login button, it asks the user to register for an account. If the user has already got an account then they press a button, which will call an event listener in a JavaScript file. Once that event listener is triggered, it will manipulate the DOM, to display a new form, that allows the user to enter their existing username and password. Once the user has entered their login details, or created an account, then the PHP file connects to the database, and checks to see if the user's details are correct. If the database finds a match, then it will display an alert saying the user has logged in successfully. If there is no match however, then the user is redirected to the login page, where they must try and log in again. Upon registering for a new account, the user must enter a username and password. They must enter the password twice, to make sure that they have entered the correct password that they wanted, and did not make any typos. If the two passwords do not match up then the user must create the account again, and they are told the reason why it was unsuccessful. Once they have entered the details correctly, the database checks to see if anyone else has the same username. If they do, then the account is not created. However, if there is no match, then a SQL INSERT statement is called, and the database is updated with the new username.

Admin

The admin page is for when the user is logged in to the admin account. The privilege that the admin account has, is that it can delete any users account that it needs to. Once you enter the usernames account that you want to delete, a SQL DELETE statement is called. The query will loop through the database until it finds a match for the username, and subsequently delete the record.

My Account

On the my account page, the user can make multiple changes to their account. They are able to change their username and password, or delete their account. Changing the username works in the same way as creating a new account, however instead of an INSERT statement being used, an UPDATE statement is used, providing the username does not already exist. An UPDATE statement is also used when updating the password. In order to do any of the above stated options, the original password is needed. This way anyone who gained unlawful access, provided they do not have the password, is unable to make changes to the account.

Legality of the Website

Disability and Discrimination act

Under the Equality act 2010, all websites must be made accessible for any user that would need to access the website. Which means that when a website is created, the developer must consider what type of users may need/want to use the website. For this website, a large amount of different types of users may wish to access it. However, the website is not available for all users. For example, if the user is visually impaired, then the website does not offer any features such as screen readers to help the user navigate and 'view' the content of the site.

Cookie law

The cookie law is a piece of privacy legislation. It was implemented by EU countries in May 2011, and was created to make users aware that by using a certain website, their data would be collected and used online. A cookie is a small text file that is downloaded once the user visits a website. The contents of a cookie can vary depending on the developer's intention for it, but usually consist of the users previous preferences, or past actions. In order for most cookies to be used, the user has to consent to the cookie being stored, meaning the website must display clearly that cookies are being used, with the option for the user to disable the cookies. Alongside being told that cookies are being used, the website has to clearly display why cookies are being used, and their intended purpose. Consent can be implied, but the message must be clearly stated to the user. Once the user has given consent once, then the website does not need to gain consent again. The exception to the rule, is that if the cookie is essential to the functionality of the website, for example remembering what the user has stored in their shopping basket, then consent does not need to be given for the cookie. However, the user does still need to be made aware that cookies are being used.

Despite being called the cookie law, not just cookies are covered by it. Any website that either stores information on the user's device, or gains information about the user, is subject to the cookie law. For example, apps on smartphones may be subject to the cookie law, as well as Local shored objects, or flash cookies.

Security Risks

Websites today face many security risks, and so when developing a website you must take all of these risks into account. One way that you can avoid having a security flaw, is by hashing any passwords that your database stores. A password hash is a password that has been taken, and converted into a series of characters, that do not correspond to the original password in any way shape or form. There are many different types of password hashes, such as MD5, SHA-1/256 and MYSQL5. MD5, and SHA-1 are no longer widely used, as they are very much considered 'Solved' password hashes.

MD5 was first created in April 1992, and was designed to be used as a cryptographic hashing function. It would hash a string, and reply with a 128-bit hash. However, it is now mainly used as a checksum algorithm, as it has major security vulnerabilities. The first flaw in MD5 was discovered in 1996, it was at that point that security analysts recommended using other hashing functions, such as SHA-1. However, SHA-1 has also now been deemed vulnerable to attacks. Due to MD5's hashes being so small, they are vulnerable to attacks such as birthday attacks, to crack the hash. It is also susceptible to collision attacks, resulting in the computer cracking the password hash within seconds of attempting to. For these reasons, MD5 is not recommended to be used as a password hash anymore.

SHA-1 was created in 1993, by the NSA. Unlink MD5, it produces a 160-bit hash, instead of a 128-bit hash. SHA-1 hashes are typically hexadecimal numbers, and are 40 digits long. It was only in 2005 that SHA-1 was deemed vulnerable to attacks. In 2010, many organisations moved away from SHA-1, to it's more secure counterparts – SHA-2 and SHA-3. In 2017, many large companies, such as Google, Mozilla and Apple announced that they would no longer be accepting SHA-1 SSL certificates, due to the amount of vulnerabilities found with the hashing algorithm. On February 2017, Google performed a collision attack on SHA-1, to prove that it was an exploitable algorithm.

In addition to password hashes, you can also add salt to make passwords more secure. Salting a password is the process of adding padding to the end of the password, to make the password larger in size. The main reason salting is used, is to protect against dictionary and rainbow attacks. Typically, every password will have it's own unique salting added to the end of it, before the password is then hashed. Another benefit of salting, is that if a user uses the same password for more than one website, then these passwords will not look the same in another websites database, due to a different salt being added.

Another form of attack that hackers may use to target your website is SQL injection. This can be very harmful to a database, and can allow the hacker to gain unauthorised access to all accounts, and can even make the whole database be dropped. SQL injection usually happens when the user must enter some data into a website, such as a username/password, and instead of giving the relevant data, the user gives a line of SQL code, that is then executed on the database. One way that a developer can counter this method of attack, is by using PHP's prepared statements. By using prepared statements, the input given is not executed directly, rather the line of code to be executed is bound with the entered parameters at a later point, resulting in the code using a different

protocol to execute. This means that any harmful code entered by the user is not recognised as code by the database.

Another security flaw is deciding what users can access what web pages. If a user enters the web URL of a site that they are not meant to be able to access, then there must be a countermeasure put in place to block the user accessing the page.

Website vs Assessment Brief

The website contains a large portion of the specification, but not all of it. Some features that the website does not include, are cookies, and some security features. All webpages contain a large amount of HTML, PHP, JavaScript, and bootstrap. The Results and Fixtures page contains a large amount of JQuery. When the user enters input into the login/My Account page, prepared statements are used. By using prepared statements, the website is not vulnerable to SQL Injection. A security flaw with the website is that the passwords are not stored as password hashes, thus meaning if a hacker was to gain access to the database, then they could see all the passwords in plaintext. Another problem with the website is that if the user was to enter the URL, for example to the Admin page, then there is no security measures in place to stop unauthorised users gaining access.

The My Account page is successfully able to send a query to the database, add, edit and delete information. The Login page is also able to add data to the database. Thus showing that the website can successfully communicate with a database.

The home page makes good use of AJAX, and AJAX jQuery, as when a news article is called, AJAX is used to fetch the text file from the server.

Overall, the website successfully meets the majority of the brief, with the only exception being not including cookies.