Abertay University

Dynamic Web Development 1 Report

Declan Doyle

BSc Ethical Hacking
Abertay University
Dundee, United Kingdom
1600219@abertay.ac.uk

November 18, 2017

Contents

1 Introduction

This report will look at a website created on the town "Linlithgow". The design of each page of the website will be evaluated, considering consistency, accessability, userfriendliness and responsive design. Page functionality will also be considered: static and dynamic content will be explored, along with user interaction and usability. The coursework brief will then be considered, and each aspect will reviewed to determine whether the website adheres to each point of the brief. Finally, the website's security will be considered. The risks of a website will be investigated and the websites mitigations will be reviewed.

2 Page Design

2.1 Index

The index page follows a fairly simple design. All of the content if put into a centre area and the sides of the website are left empty. The background image is fixed and so when scrolling on the website, the image stays the same. However when the page is viewed on a mobile device or a device with a small viewport, the image below the title text does not scale well to that small a size, but the rest of the webpage does. In terms of accisibility, the index page has no difficult to read text, apart from the table, but this has a hover feature so when the user hovers the mouse over an item in the table, it will highlight the item and the background will become white, making the text more readable.

2.2 Location

The location page follows the same design as the index page. A content area in the middle of the page, and two vertical empty spaces at each side. The background image is also the same as the index page. This page also suffers from the same effect as the index page, where when viewed on a mobile device with a small viewport, the image does not scale well. The map on this page would perhaps be more suitable design-wise if there was a border around it, as the green colour on the map can sometimes clash with the green in the background.

2.3 Town History

The town history page has the same design and backrgound as the previous pages, and has the same issues on a mobile screen. The page has three images in a vertical layout, that, when the mouse hovers over them, text appears underneath.

2.4 Town Events

The town events page also has the same design and background as the previous pages, and the same issues on a mobile screeen. The events page has a slightly different design, as it has a two by two layout rather than all the content being in a vertical line. There are alternating headings and pictures, and when the pictures are clicked on, text is revealed about the event. When this is viewed on a small viewport, the images will disappear to either the

left or the right, and the user will have to scroll to the side in order to click them and get them to move back.

2.5 Login Page User Page

Both the login and user page have the same design and background as the rest of the pages. As there is no header image on either of these pages, they do not suffer the same problems that the previous pages do on a mobile device. The Login and user page have several bootstrap forms, that span the width of the middle grid they are in. If the viewport is small then the forms span the width of the page. These pages are the most suitable for mobile viewing as the user experience on these pages does not change when being viewed on a mobile device.

3 Page Functionality

3.1 Index

The index page has several features. The navaigation bar at the top of the page allows the user to go to any page on the website. The navigation bar is dynamic, as it will change to a drop down menu if the viewport is small enough. There is a button that will display an event that is on in Linlithgow. Once clicked the information is displayed below the button. Below this, there is a bootstrap table, that will highlight the row that the mouse hovers over. This table contains information from an online sql database, that can be edited and updated with administrative credentials on the user page. At the bottom of the page there is a weather widget. This will change depending on the weather of Linlithgow.

3.2 Location

The location page contains a map plugin to show the location of linlithgow on a map. The plugin connects to a google maps api and displays the location. As with every page, the location page has a navbar, with the location section showing as the active page.

3.3 Town History

The town history page contains three images of historical events that have occurred in linlithgow. When the user hovers the mouse over one of these images, a description of the image appears underneath. The town history page also contains the navbar, with the town history section showing as the active page.

3.4 Town Events

The town events page shows various pictures of events that happen throughout the year in linlithgow. The user can click on an image and reveal more information about the event. When the viewport is too small for the bootstrap grid system the page uses, the items stack verticallty, however, due to the alternating layout of the page, the images and titles stack in an inconsistent layout. As with other pages, the events page has a nav bar with the town events section marked as active.

3.5 Login Page

The login page has two bootstrap forms. These forms span the length of the 'content space', and if the viewport is adjusted then the forms adapt to the size of the viewport. The login page also has a navbar, however it does not display as active on the navbar. This is an intentional choice as the login page, in terms of user experience, is not considered an active page, as the user will only be on it momentarily, whilst they create a new account, or login. When the user logs in a session is created and their username is stored. If the user clicks the button to login, but is already logged in, then the login page will be bypassed and the user will be redirected to the user page.

3.6 User Page

The user page will display the users name by reading the session variable that was created when the user logged in. There are three bootstrap forms that will adapt if the viewport is adjusted. On this page the user can input data into the first form to add something to the table on the index page, and on the second form they can search for something by the name of the place, and then in the following fields add data that will update that item in the table. In the third form the user can enter the name of an item and it will be deleted. However if the user is logged in with anything other than the administrator account, they will be shown a bootstrap alert that informs them that they do not have sufficient permissions to edit the table. If the user tries to go to this page when they are not logged in they will be redirected to the login page.

4 Coursework Brief

4.1 Clear use of HTML 5

The website makes good use of HTML 5 by using various HTML tags. Headings are used at the start of each page, and sub headings are used for certain sections. Paragraphs are used for sections of text, and data about each page is put in the head section.

4.2 Use of Bootstrap

The Linlitgow website makes extensive use of bootstrap. Each page uses a bootstrap navigation bar for clear navigation to each page. Each page also uses a bootstrap grid to provide a consistent layout across each page. Any piece of content is stored in a bootstrap div, in order to compartmentalise the content of the website. There is a bootstrap table on the index page, and bootstrap forms on the login and user pages. All images used have bootstrap classes, so they are either rounded, or use the thumbnail class. There are also several bootstrap buttons on various pages. There is also a bootstrap alert on the welcome page that displays if the user does not have sufficient permissions and alerts them of this.

4.3 Use of Javascript to manipulate the DOM based on an event

On the town history page, there are three images of historical events in Linlithgow. When the mouse hovers over one of these pictures, a small description of the image appears underneath. A javascript function is called when the mouse hovers over the image which adds a new paragraph element to the DOM. When the mouse leaves the image, the child element - the paragraph - is removed from the parent element - the div.

4.4 Javascript loading dynamically changing information

On every page inside the nav bar there is a clock of the current time at the users locatation. On each page load a javascript function is called that gets the currrent time and edits the inner html of the nav bar.

4.5 Use of jQuery in conjunction with the DOM

On the events page, there is a bootstrap grid that alternates between pictures on one side and text on the other. When the user clicks on the image, it moves across to cover the title, to reveal a description of each event underneath. The page has a jQuery file included that hides all the text underneath the images, incase there is any overlap. It then, when an image is clicked on, animates the image moveing over towards the title text, and if it is already on the title, it animates a move back.

4.6 Use of a jQuery plugin

On the index page, at the bottom of the page, there is a jQuery plugin. A plugin called simple weather is used to get the weather information for Linlithgow. It retrieves the weather information and then prints it to the html of the page. There is also a custom font installed through CSS to display weather icons next to the text on the main page.

4.7 Use of the jQuery AJAX function

On the index page there is a button that when clicked will display an event that is occuring in Linlithgow. AJAX is used to load in a text file containing the information. When the button is clicked a javascript function is called that generates a random number between one and three. It converts that number to a string and adds '.txt' to the end of the string. This is passed to another javascript function that creates an xmlhttp request, getting one of three text files, depending on the number passed by the previous function. The html of the page is edited and the content of the file is added to a div.

4.8 Use of Cookies

The website uses a cookie to store the value true if the user accepts that the website uses cookies. There is a bootstrap modal at the load of the index page alerting the user that the website uses cookies, in order to comply with the cookie law.

4.9 User login functionality

The website has a login page that makes use of php and mySQL. There is a username and password field within a bootstrap form that checks an sql database to verify that the information the user has enetered is correct, and

if it is then the user logged in. The page will display an error if the user does not have the correct credentials.

4.10 Admin section of the website

When the user logs on, they are directed to the login page. They can attempt to add, edit or delete information however if they are not logged in as the admin account, they will get a boostrap alert saying they do not have sufficient permissions to edit the database. This was implimented because of the functionality a user account has, as they cannot do anything a guest account can, so rather than the user not have a page themselves, there is an alert box instead.

4.11 Ability to add, edit and delete information from a database

On the welcome page, accessible once the user has logged in, there are several bootstrap forms. The user can add items to the table on the index page, they can edit items in the table, or they can remove items in the table. An sql insert statement is used to add items to the database, and an sql update statement is used to edit data. A delete statement is used to remove data from the database.

4.12 Appropriate consideration of security measures

On the login and welcome page, all data is sent via the post method, so that no data can be seen in the url. Whenever data is sent, a function is used to make sure the data will not cause any cross-side scripting. The function trims the data of any spaces at the front and end, and then it removes any slashes so that there is no danger that the data entered could be a link. Finally the data converts all html special characters into strings so that no user could enter a script tag. The website also stores the username in a session, so that the data is not stored on the users computer and so is not vulnerable.

5 Security

5.1 Risks

Web applications can suffer many risks if not considered in the development process. The Open Web Application Security Project releases the OWASP top ten, a consensus about the most critical security risks to web applications. Some of these include SQL injection, cross side scripting, and sensitive data exposure. All of which the Linlithgow website will be vulnerable to unless considered during development. As the website uses forms to connect to an sql database, the website is vulnerable to cross side scripting and sql injections. Because the website also uses an sql database, it is vulnerable to sensitive data exposure. The linlithgow wesbite does not hash passwords before sending them to the database. This is a large security risk as the passwords are stored in plain text in the database, meaning that if the database was compromised, then all accounts would immediatly also be compromised. This was not an oversight, as because the website was never intended to be released commercially, the need to protect users accounts was not needed. Also, because the website does not use https, any data sent to the server is at risk, and so modern browsers will warn the user that the connection is insecure. Another reason behind not hashing passwords that was considered, is that the admin account can only edit the "places of interest" database, and so all that is effected is the index page. There is no major damadge that can be done. The linlithgow website does also not use sql prepared statements, for the samre reasons as stated before.

5.2 Mitigations

During the development of the website, several mitigations were put in place to make the website more secure. Cross side scripting and sql injections are prevented as any input from the user is passed to a function that will strip the spaces, slashes, and html characters. This minimises the risk from the affore mentioned threats. By storing the users username in a session, an attacker cannot retreive it from the users computer, as it is not stored locally.

6 Future Work

The linlithgow website as it stands can still undergo a lot of work. The primary concern is the lack of security in certain areas of the website. Even although, as stated earlier, there were decisions made to overlook password hashing and sql prepared statements, these would be immplimented in future releases of the website, to ensure maximum security for the user. Following the securing of the web application, some additional user features would be added. Currently, when the user logs in, they cannot do anything different on the website that they could do as a guest. A future implimentation would be to allow the user to chose their favourite item from the "places of interest" database, and the website would remember this choice, by adding it to a field in the user database. The design of the website would also be changed so that it is more suitable for mobile devices. The images at the top of most pages could be dynamically resized based on the viewport. Cookies could also be made user of better, by implimenting a "dark mode" for the website. There would be an option for the user to switch to dark mode, and the users choice would be saved within a cookie. The darkmode would also help mobile devices, as a lot of modern mobile devices use OLED screens, and so dark modes can help with battery life.