CAB230 Assignment 1

Website – WifiFinder

*George Delosa & Ari Luangamath*

*(Student No. 9751696 & No. 9446826)*

*Date Due: 28th May 2018*

**Works well in:** Chrome, Firefox?, Edge?, Safari?

**Preferred browser:** Chrome.

Table of Contents

**1 – Statement of Contribution3**

**2 – Test Plan4**

**3 – Web Design Principles21**

**4 – Video link31**

**1 – Statement of Contribution**

**DECLARATION**

You must sign below. By signing this form, you agree to the following:

* I/We declare that all of the work submitted for this assignment is our own original work except for material that is explicitly referenced and for which we have permission, or which is freely available (and also referenced).
* I/We agree that QUT may archive this assignment for an indefinite period of time, and use it in the future for educational purposes including, but not limited to: as an example of previous work; as the basis for assignments, lectures or tutorials; for comparison when scanning for plagiarism, etc.
* I/We agree to indemnify QUT and hold it blameless if copyright infringements are found in this work and the copyright owner takes action against QUT that is not covered by the normal terms of Educational Use

|  |  |  |  |
| --- | --- | --- | --- |
| Team Members Details & Contribution | | | |
| S# | Student Name | Signature | Statement of Contribution |
| N9446826 | Ari Luangamath | Ari | 50% |
| N9751696 | George Delosa | George | 50% |

**2 – Test Plan**

|  |  |
| --- | --- |
| **Test** | **Result** |
| Accessing the home screen | **Input:**  <http://localhost/CAB230/src/index.php>  http://puu.sh/AtceS/e2b9c51d91.jpg**Output:** |
| Registering as a new user | **Valid Input:**   1. Press register button: <http://localhost/CAB230/src/register.php>   http://puu.sh/AtcpO/3d9cc663c9.jpg  **Invalid input:**   1. Press register button: <http://localhost/CAB230/src/register.php>   http://puu.sh/Atczu/90ab0e6739.jpg  **Output (valid):**  http://puu.sh/AtctV/ce9d63902d.jpg |
| Logging in as an existing user | **Valid Input:**   1. Press login button: <http://localhost/CAB230/src/login.php>   http://puu.sh/AtcJh/262915c13e.jpg  **Invalid Input:**   1. Press login button: <http://localhost/CAB230/src/login.php> 2. If client side validation passes but with wrong credentials, “Incorrect credentials” appears. Otherwise, individual field errors appear.   http://puu.sh/AtdjR/658b9de4b4.jpg  **Output:**  http://puu.sh/AtcKV/dce12ea816.jpg |
| Logging out | **Input:**   1. Login first.   http://puu.sh/AtcKV/dce12ea816.jpg   1. Press logout button: <http://localhost/CAB230/src/logout.php>   **Output:**   1. User is logged out by unsetting session variables (menu buttons change to reflect this)   http://puu.sh/AtKEw/5d82aa8688.jpg   1. User is redirected to home page. |
| Adding a review | **Setup:**   1. Login first. 2. Press search on home page. 3. Select a hotspot. 4. Click “Write a review” hyperlink   **Valid Input:**  http://puu.sh/Atgk5/1608c89ac6.jpg  **Invalid Input:**  http://puu.sh/AtggP/b5acbf0df2.jpg  **Output (valid):**  http://puu.sh/Atf3I/5834ea778e.jpg  http://puu.sh/AteY9/4f9eacbf7c.jpg |
| Searching for an item that exists in the database | **Setup:**   1. <http://localhost/CAB230/src/index.php>   **Input (Suburb only):**  http://puu.sh/Atfna/e811c05107.png  **Output (Suburb only):**    **Input (Near me only):**  http://puu.sh/Atfou/453afa9a8f.png  **Output (Near me only): (will vary based on current location)**    **Input (Rating only):**  http://puu.sh/Atfqq/acd8538cc3.png  **Output (Rating only):**    **Input (Name/address text only):**  http://puu.sh/AtfwH/84ae9ad733.png  **Output (Name/address text only):**    **Input (All fields default):**  http://puu.sh/Atfjd/fcd9f32e76.png  **Output (All fields default):**    **Input (Combination of fields):**  http://puu.sh/AtfGf/d85e330f7e.png  **Output (Combination of fields):**  http://puu.sh/AtJOf/b0bd1848bd.jpg |
| Searching for an item that does not exist in the database | **Setup:**  <http://localhost/cab230/src/index.php>  **Input:**  http://puu.sh/AtJR7/aaec322ad2.png  **Output:**  http://puu.sh/AtJSo/68d6767ae9.jpg |
| Accessing an individual item page | **Setup:**   1. Press search on home page. 2. Select a hotspot.   **Input:**  http://puu.sh/AtJV3/2e11744683.jpg  **Output:**  http://puu.sh/AtJXg/4518db5d03.jpg |
| Attempting to use a cross site scripting attack but not being successful | **Setup:**   1. Login first. 2. Press search on home page. 3. Select a hotspot. 4. Click “Write a review” hyperlink   **Input:**  http://puu.sh/AtLsX/061228963b.jpg  **Output:**  http://puu.sh/AtLww/ae46f11c6b.jpg |
| Attempting to use an SQL injection attack but not being successful | All SQL queries use prepared statements with sanitized inputs, thus an SQL injection attack is not possible. |
| Unregistered user not being able to log in | **Input:**  <http://localhost/CAB230/src/login.php>  http://puu.sh/Atgxr/41e4e2ebf4.jpg  **Output:**  http://puu.sh/Atgvy/58a3574d80.jpg |
| Operating gracefully in multiple resolutions | **Output:**  http://puu.sh/AtK5x/b9b3ff21f5.png  http://puu.sh/AtK8O/26fac6c7fa.jpg  http://puu.sh/AtKaB/767ba556bb.jpg  http://puu.sh/AtSRW/7e0fd88e3b.jpg |
| An Example of a SQL Query that has been implemented in code and a description of where this Query is located (for example the file and method names). | **Query (and associated code):**  $stmt = $pdo->prepare("SELECT \* FROM members WHERE email = :email");  $stmt->bindValue(":email", $email);  if($stmt->execute()) {  if($stmt->rowCount() > 0){  $row = $stmt->fetch();  if(password\_verify($password, $row['password'])) {  return $row['ID'];  }  }  }  **File:** login\_server\_validation.php  **Method:** TryLogin($email, $password) |
| On the search results page: a map showing markers for all search results. (Add on #1) | **Input:**  http://puu.sh/AtK1r/60b0930563.png  **Output:**  http://puu.sh/AtK2s/aa306ffd5a.jpg |
| On the individual item page: a map showing the item. (Add on #1) | **Input:**  http://puu.sh/AtJV3/2e11744683.jpg  **Output:**  http://puu.sh/AtJXg/4518db5d03.jpg |
| Evidence that the geographic microdata is valid as reported by Google’s structured data validator (Add on #2) | **Output:** |
| Evidence that the microdata is valid as reported by Google’s structured data validator (Add on #2) | **Output:** |
| Evidence that the site icon displays on a mobile phone (iOS or Android) home screen (Add on #3) | **Output:**  https://cdn.discordapp.com/attachments/297391247083962369/450933980502425600/Screenshot_20180529-180900.jpg |
| Evidence that the site pages adjust well to a mobile phone screen (Add on #3) | **Output:**  https://cdn.discordapp.com/attachments/297391247083962369/450934185578725376/Screenshot_20180528-111235.jpghttps://cdn.discordapp.com/attachments/297391247083962369/450934185578725377/Screenshot_20180528-111249.jpg  https://cdn.discordapp.com/attachments/297391247083962369/450934102372384768/Screenshot_20180528-111326.jpghttps://cdn.discordapp.com/attachments/297391247083962369/450934045791092736/Screenshot_20180529-180734.jpg |

**3 – Web Design Principles**

1) User Experience

The website ensures that it meets the standards of good user experience by keeping content simple and obvious. The home page of the site displays a search bar on it’s own, with the line “A place to discover and review WiFi Hotspots near you”, indicating exactly what the search bar is used for, alongside placeholder text to communicate example inputs to the user (Figure 1). The menu buttons at the top-right of the page are labelled common functionality expected of a website, allowing users to immediately feel comfortable and aware of how to navigate content on the site. By taking into consideration the kinds of users the site should expect to receive traffic from, design choices were made to ensure these users are able to get the information they need. As the site pertains to discovering where wifi hotspots are located, a user may decide that the only thing they want to know is where the **nearest** hotspots are. As such, the first dropdown option in the select bar is “Near me”, allowing the user to quickly achieve their goal on the site. The implementation of Google Maps provides familiar and trusted functionality for the user as well. Additionally, it may be beneficial for the user to know not just where a hotspot is relative to them, but how effective it is. As such, immediately after selecting a hotspot, they are able to see reviews from other users. Speed of finding information is essential for this website, and is essential for a good user experience overall.

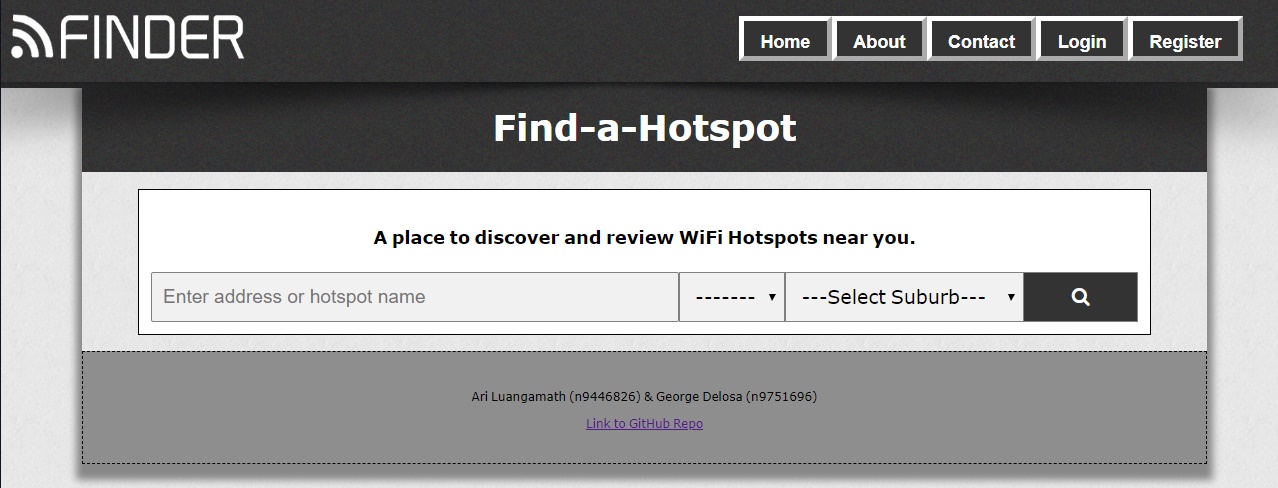


Figure 1 - Easy to understand, following good user experience

2) Visual Design

Utilising a consistent visual style gives the site an appealing look. The primary colour of the site, grey, was chosen as it reinforces a neutral, balanced, and informative theme, which is suitable for the content of the site. Pages have been designed to be minimal in how much content is displayed at once, and utilise whitespace efficiently to create clear visuals. This can be seen most notably in the register, login, and review creation forms (Figure 2). In addition, the visuals of many smaller components have been taken into consideration as well, such as font consistency, colour contrasts, and familiar icons, all to further improve the readability of the site.

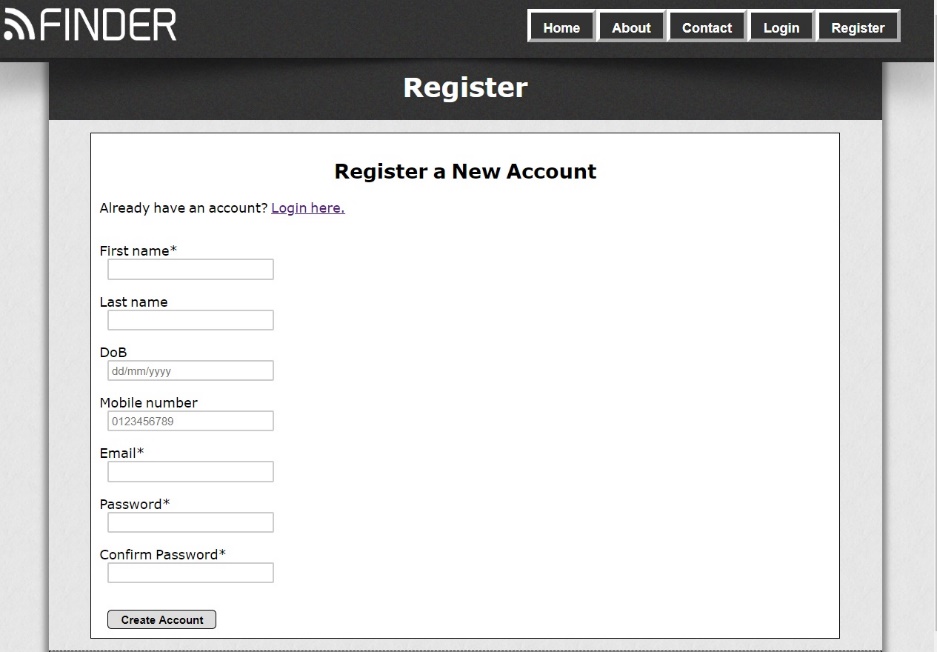


Figure 2 – Minimalistic and simple visual design, taking advantage of whitespace

The site is also capable of running effectively on multiple browsers and different versions of each, due to enforcing backwards compatible code architecture. Importantly, the website also dynamically adjusts for smaller resolutions, whether it be due to a small browser window or a mobile device. This allows users to continue to utilise the site even under space limitations.

3) Page Layout

It is important that users don’t begin to feel lost in a website design because it is too foreign, so care has been taken to ensure a familiar visual design. Many sites utilise an ordered page structure, usually as some form of block layout, consisting of a header, footer, sidebars, menus and main content areas. This structure is advantageous in creating content that is easily understood based on a section’s position, size, and other attributes. By lifting these proven to be effective page layouts, the user is not only able to benefit from the structure, but also gain a quick understanding of a completely new site based on previous experience. The page layouts described is visible in the WifiFinder website, as can be seen in the figure below:

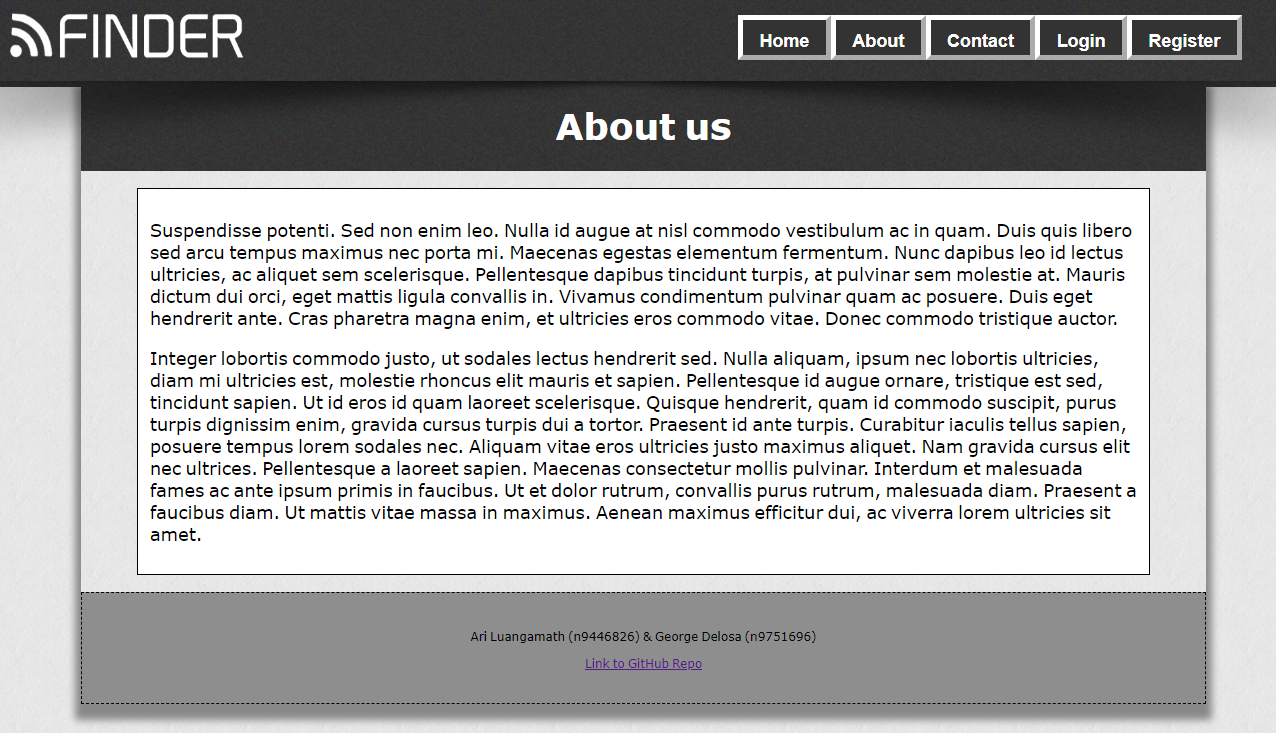


Figure 3 - Block page layout

A clear header, navigational menu, central content area and footer can be discerned from the site. The entire site is also centered no matter how large or small the resolution may be, ensuring that the sections of the page retain their positional value. Additionally, at very small resolutions such as on mobile, most elements on the page shrink, and the menu bar is transformed into the typical three-horizontal-bars icon, which displays the menu buttons once again in a vertical ordering when clicked. These implementations make the WifiFinder site follow a flexible page layout. (Figure 4)

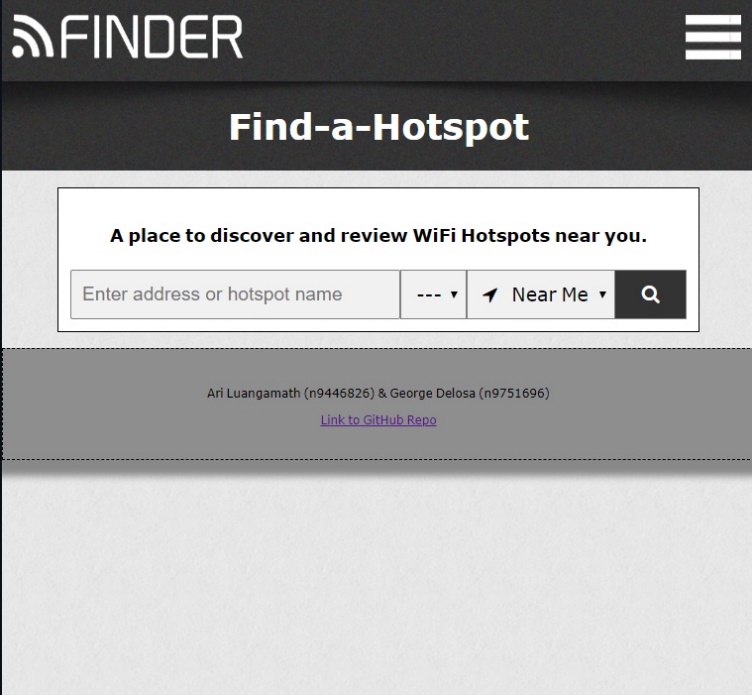


Figure 4 - Flexible page layout design

4) Standards

The HTML and CSS code of all sites have been validated against a formal set of rules and technical specifications maintained by W3C. Compliance with W3C standards ensures maintainability and portability between browsers as well as different devices. In addition, the Web Content Accessibility Guidelines (WCAG) have been undertaken, and in doing so have assured that users with special needs can still understand and interact with the website. Namely, the key points are that of being perceivable, operable, understandable and robust.

In achieving perceivability, the WifiFinder site demonstrates this through the adaptable nature of the site’s content such as resolution or font size changes, while still remaining visually and functionally effective. Non-text content such as logos and other imagery have text alternatives to ensure that the site maintains flow even resources are unavailable.

The site is operable because it is designed to be simple to navigate, patient with users by allowing enough time for content to be read, and keyboard navigational in the event that a mouse is inoperable. The site also is devoid of any content that may induce seizures or other negative responses that may affect the health of users.

The WifiFinder site utilises javascript to help the user avoid and correct mistakes quickly and effectively, making the site more understandable (Figure 5). Text is readable, and content appears in a predictable way by reusing the same page layout on every page.

Finally, robustness is assured by utilizing metadata, microdata and code compatibility architecture, so that a wide variety of user agents can understand the site’s content, including assistive technologies for special needs users.

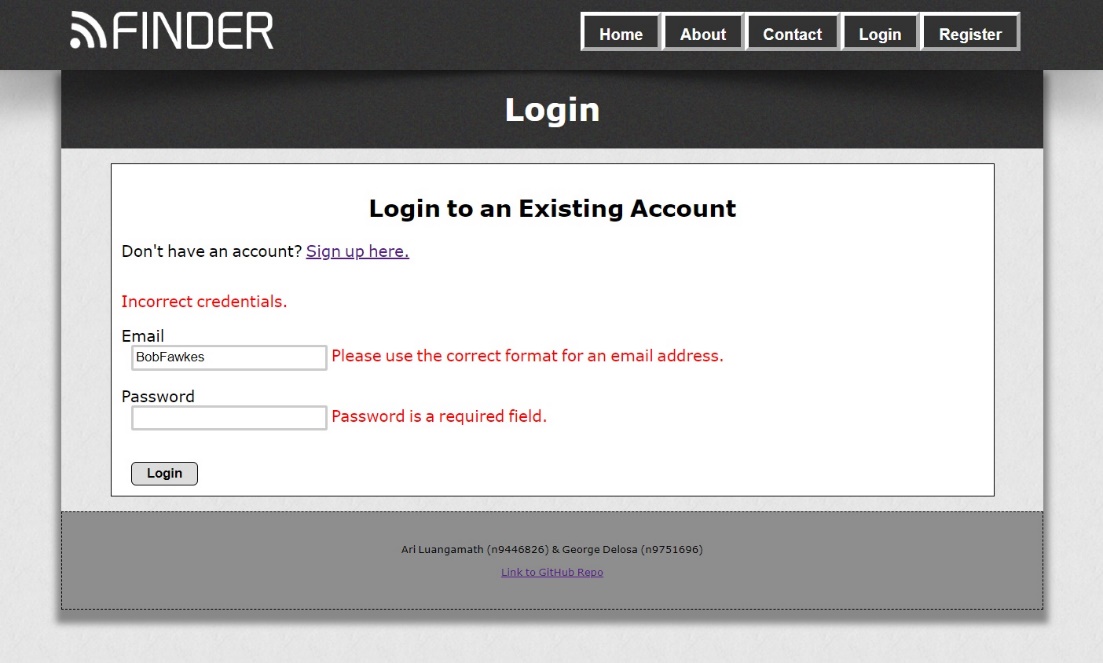


Figure 5 - Responsive error reporting