

**Web Development:**

**Advanced Web Scripting Concepts**

**HL9W 35**

**Assessment**

**LO 2,3**

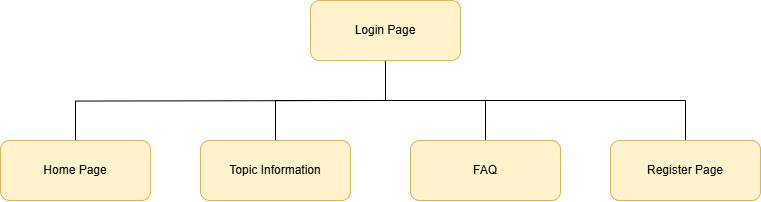
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student** | |  | Kamil Milej | |  | | | |
| **I.D.** | |  | 2264629 | |  | | | |
| **Date** | |  | 16/05/2025 | |  | | | |
|  | **Pass** | |  | **Fail** |  | **Remediation** |  |  |
| **Tutor** | |  |  | |  | | | |

© Fife College

These materials may not be reproduced without the copyright holder’s permission.

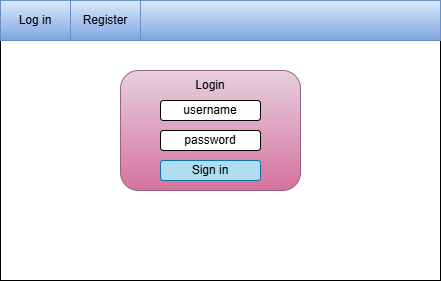
***Design Document***

1. Site Structure Diagram

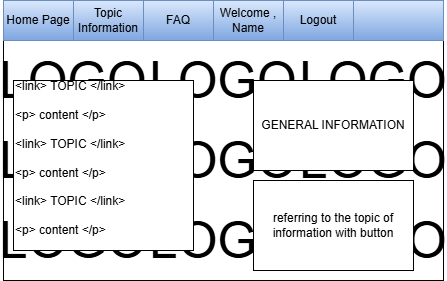
  
  
This diagram shows the structure of my website, which includes the following:

* The login page is the first page the user sees.
* From the login page, the only available action is to navigate to the registration page.
* After logging in, the navigation bar displays the remaining page options along with a Logout button.

1. Wireframe design for each screen
2. Login Page

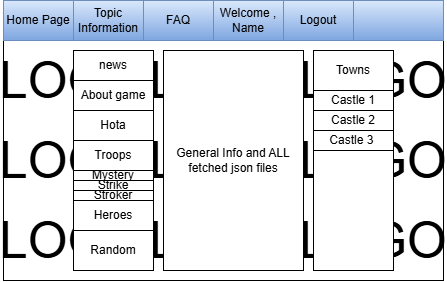
The login page will consist of a navbar that will appear on every page and will include two options: "Log In" and "Register". Additionally, there will be a window that allows the user to log in to the site. According to the requirements, the user will not be able to perform any actions on the site without being logged in. After logging in, the user will be automatically redirected to the homepage, which serves as the main page after login. The login process will use the following prepared statement: $stmt = $dbConnect->prepare("SELECT user\_id, username, password, role FROM Users WHERE username = ?"); which means it uses placeholders, which protects against SQL injection.

1. Home Page



The homepage is the main landing page that contains links to other pages and some general information. It also includes a navbar, which changes dynamically after logging in, additional options appear, while the "Register" and "Login" buttons disappear and are replaced with a "Logout" option. Additionally, the logo is part of the visual design. The page is divided into two sections, left and right, which contain general content and a button that redirects to the "Topic Information" section.

1. Topic Information Page



Topic Information is our main page where many features have been added, such as DOM Programming, AJAX, and a Templating System.

After entering the page, we also have a navbar, a logo, and the main section which is divided into three columns. In the left column, there will be buttons that change the content in the middle column, and one of the buttons will have a dropdown that expands and shows 3 categories which must be added to the assignment - these are 3 different pieces of information about dragons.

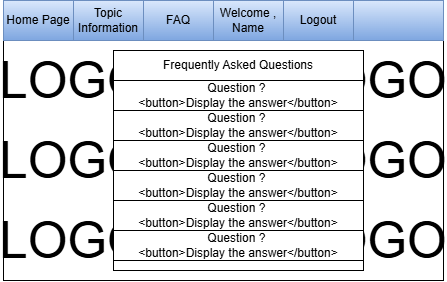
I already tried to use the method from lesson : <a href="employees.json" id="loadEmployeess">Click here to load employees</a> and const file = loadEmployees.getAttribute("href"); to put href into te variable but from my attempts, I found out that this cannot be used with buttons. That’s why in the left column I will use data-json, which perfectly shows the data even with buttons. The school example I will use in the right column. Also, in this column, AJAX with JSON is used, that is, asynchronous JavaScript, whose main purpose is usually to handle the page without reloading it and to fetch or send data in the background, such as JSON or XML. Whenever I use fetch() to asynchronously communicate with a server without reloading the page, I am using the AJAX technique, because AJAX is a concept, not a specific technology.

In the middle column, all data from both side columns will be displayed.

In the right column, there will be "Towns" and a "Show More" button, which will trigger a function when clicked that displays additional information below, in this case, about castles in Heroes 3.

After clicking "Show More", the castles will appear in the centre using: <li><a href="json/Towns/castle.json" id="loadCastle"> Castle</a></li> and thanks to getAttribute("href"), it is assigned to a variable which is then used to fetch the JSON file. The data is then displayed using appendChild. Also, <template id="castleTemplate"> </template> will be used here to define a reusable HTML structure. It is not rendered by default, but allows JavaScript to clone and populate it dynamically

1. FAQ Page



The FAQ page also includes a background, a navigation bar, and a central content section. In the center section, the Frequently Asked Questions are displayed.

Each question is shown individually, with the option to expand or collapse the answer.

That’s why the following are involved in this page:

* Creating DOM elements
* Appending DOM elements
* Removing elements from the DOM