

**Web Development:**

**Advanced Web Scripting Concepts**

**HL9W 35**

**Assessment**

**LO 2,3**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student** | |  |  | |  | | | |
| **I.D.** | |  |  | |  | | | |
| **Date** | |  |  | |  | | | |
|  | **Pass** | |  | **Fail** |  | **Remediation** |  |  |
| **Tutor** | |  |  | |  | | | |

© Fife College

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

**Assessment task 1**

**Outcomes covered 2, 3**

**Assessment task instructions**

There is one assessment for this Unit. This is an open-book project covering Outcomes 2 and 3. The project requires you to design and implement a dynamic web application, the main purpose of which is to provide a showcase of your skills gained in this unit. You will create a design document for one of three scenarios then implement the design using the technologies specified once your design is approved by your lecturer. The completed web project should be uploaded to the college web server.   
  
**Assessment Brief: Interactive Hobby Website**

**Scenario Overview:**

You are to create an interactive hobby-themed website for a community of enthusiasts. Members of this community share a common interest in **one** of the following topics:

* **Fantasy Game Equipment** (e.g., magical swords and artifacts)
* **Dragons** (e.g., mythical species, habitats, lore)
* **Space Transportation** (e.g., rockets, launchers, and parts)

You will choose **one** of the above scenarios and build a small website that allows users to:

1. **Log in** to a members-only area.
2. **Access a landing page** that introduces the chosen topic and welcomes the user.
3. **View detailed information** about a list of items related to the topic (e.g., weapons, dragons, or space transport artifacts).
4. **Explore a dynamic FAQ section**, where users can click on questions to reveal more or less information for each question

**The requirements of the web application are:**

1. Users will be required to successfully “login” to the application using a given username and password

* 1. In order to test this requirement you will need to implement a simple

server-side database pre-populated with a small number of users

2. Users should be able to “logout” of the application

1. The application will consist of a minimum number of pages as follows:
   1. “Login” screen
   2. “Navigation\Home” screen
   3. “Topic Information” screen
   4. “Frequently Asked Questions” screen
2. The application will demonstrate your knowledge and skills in the following areas:
   1. DOM Programming
   2. AJAX
   3. Design Patterns
   4. Templating Systems
   5. Security in relation to web applications
   6. Deployment of a web application
3. Access to any of the screens in the application (other than “Login”) should not be possible after a user has “logged off”, or if the browser window has been closed and an attempt made to re-open.
4. The application will implement appropriate security measures to ensure that any attempts at SQL Injection at login using the following sample user data are intercepted and safely dealt with:
   1. SELECT \* FROM `Users` WHERE username = 'user1' and password = 'password' or 1=1
5. The application should demonstrate a simple HTML templating system for the topic information page (<template> tag with Fetch API is sufficient)
6. The application should demonstrate a simple implementation of the MVC (Model View Controller, separating data, user interface and control logic
7. Upon completion, your application should be deployed to a live server
8. Finally, your application should be demonstrated to your lecturer for final testing

**Design Document Requirements:**

1. Site structure diagram

1. Wireframe design for each screen\page:

* 1. A textual description of the functionality of the page in terms of (where appropriate):
     1. DOM programming
     2. AJAX
     3. MVC design pattern
     4. Templating system
     5. Security
  2. Description and placement of “static” HTML elements on the page
  3. Description and placement of “dynamic” HTML elements that will be introduced on the page as a result of any DOM programming and\or AJAX functionality

**Additional Information:**

# “Login” screen

Simple client-side validation should be implemented to ensure the presence of a username and password before submission to the server. Prepared statements should be used.

# “Navigation\Home” screen

This screen should receive a page request from the client, and redirect the request to the appropriate page

# “Topic Information” screen

1. Each topic has three information sheets. The user should be presented with the three choices appropriate to your chosen brief. For example, the Fantasy Game Equipment choices are:
   * Swords
   * Apparel
   * Miscellaneous Items

1. The user will select one of the above choices.
2. A Fetch call will retrieve an JSON file on whichever course is chosen – the JSON files will be made available to you on ilearn
3. The JSON file will contain a list of information about each topic. For example the Fantasy Game Equipment titles are:

• Name, Description, Condition

1. Any information lists from previous actions should be removed from the DOM
2. The JSON file returned via Fetch API will be used by JavaScript to modify the DOM to display a list of the information for the chosen topic
3. This task must be accomplished by using JavaScript to modify the DOM – it is not acceptable to use CSS or a library such as jQuery

# “Frequently Asked Questions” screen

1. Present the user with three common questions of your choosing about your topic.
2. For each question:
   * + Display the full question
     + Provide a means for the user to choose to “Display the Answer”
     + The “answer” displayed will give a few sentences in response to the question.
     + The “Display the Answer” option will display additional information on the chosen topic by:
       - Creating DOM Elements
       - Appending DOM Elements
       - Provide a means for the user to “Close Answer” on the chosen question by removing Elements from the DOM