**EXPLORE- Problem Analysis**

The requirements of this task as stated by the task sheet is to design, develop, and evaluate the process of creating an ‘interactive learning object’. The learning object has to be built with the intended audience in mind as listed in the task sheet. As stated by The University of Windsor, an interactive learning object is a tool comprised of different interactive elements such as videos and activities that assist and engage students in their learning. An interpretation of the interactive learning object model are browser-based learning games.

While browser games are generally single-player or multiplayer games played through a web browser on a device (Desktop or Mobile) for entertainment purposes. An article by Science Direct states that browser-based learning games can very positively affect students learning and can assist students in deepening their knowledge of certain subject areas. Web-based learning games are built off standard web development tools such as HTML (and CSS), and PHP for server-side programming. This project will utilise the standard web technologies, HTML, CSS, and PHP to create a single-player browser-based learning game.

**EXPLORE- Rationale**

This project will be used to create a practical browser-based interactive learning object that will serve as a functional product to assist students in their education. With browser games quickly growing and being accepted as practical educational tools, the skills learnt and applied for this project have transferability to real-life applications.

**EXPLORE- Specifications/ Objectives**

There will be multiple individual games, each will be based specifically around a subject that requires further teaching, all hosted on one primary platform (Website).

The main platform will be called ‘The Learning Hub’ and will have a simple and easily expandable UI. Each ‘Game’ will have two sections, a ‘Learning’ and a ‘Quiz’ section. The learning section will provide students with the appropriate knowledge surrounding the topic. The quiz section will test the students on their interpretation and memory abilities through a variety of simple and complex questions on the topic.

**EXPLORE- Specifications/ Audience**

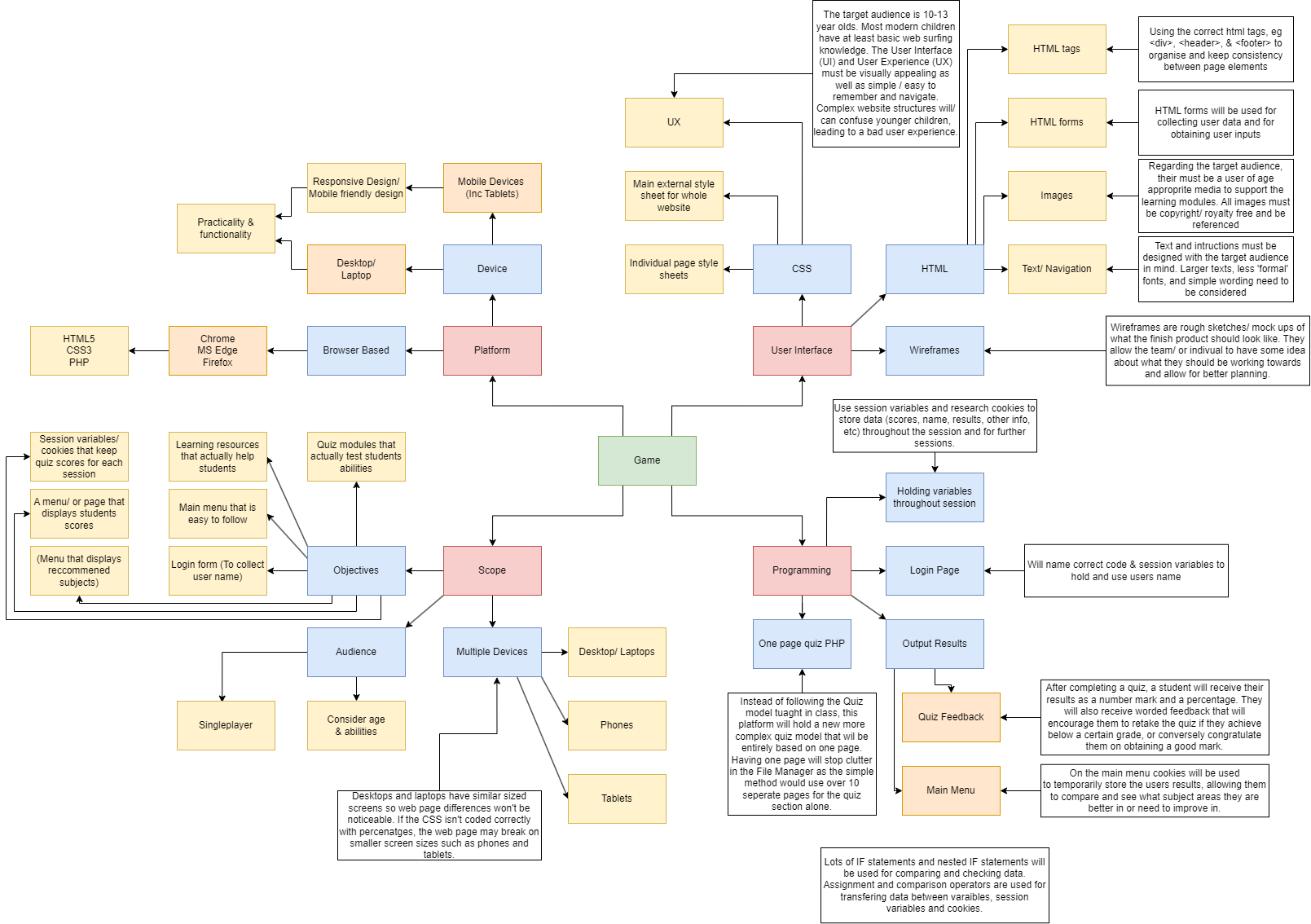
The audience for this game is children aged 10 to 13, grades 5-7. In Queensland state schools, students in grade 6 are taught the fundamentals of all maths, this includes the concept of using fractions & decimals, converting between different units, percentages (with money), interest, angles, and most importantly being capable in 4 main areas, addition, subtraction, multiplication, and division without a calculator. On the English department side, students are required to be able to use coherent text structures, understand the effects & how to use language features, understand language features & patterns, and be able to use correct and consistent grammar for better cohesion in texts. Considering that grades 5 & 6 are still in primary school, the platform design will be aimed more towards the younger children while maintaining a professional feel for the slightly older students. This may entail the use of larger fonts, bright/contrasting colours, age-standard wording, and ‘less formal’ fonts.

**EXPLORE- Usability Principles**

Usability principles will be considered throughout this project include:

1. Accessibility must be considered during the development and evaluation of the prototype and final platform. To accommodate for the target audience, specific mainline design choices have been made, these include increasing the overall text size, and considering learnability.
2. Effectiveness will be judged when children are encouraged to retake quizzes to improve on their previous results, as well as by providing educational resources that cover the appropriate topics they are being assessed on.
3. The safety of the website/ platform is strictly upheld with regular backups produced by Digital Solutions Online as well as by the developer locally.
4. Utility is achieved by providing useful learning resources and simple user-end quizzes that operate efficiently and effectively in the background.
5. Learnability is assumed that users already have general ‘web surfing’ abilities such as navigating through a web platform, entering data into a text form, and using radio buttons to select data choices.

**EXPLORE- Mind Map**



**EXPLORE- Criteria/ Prescribed**

By the due date, 8:30am 5th June 2023:

1. A functioning prototype of ‘The Learning Hub’ platform will be coded in Digital Solutions Online.
2. Users will be able to enter their name so they can be addressed by it for a more ‘personal feel’.
3. Users will have access to at least 3 modules:

* A module including 1 ‘learning resource section’ and 1 ‘Quiz section’.

1. Develop a new PHP Quiz model that is based entirely on 1 page, and that allows for more ‘add-ons’ or extra features in the future.
2. Users will receive encouraging feedback depending on their score achieved on each quiz.
3. User marks/ results will be displayed on the main menu/ homepage for ease of access.

**EXPLORE- Criteria/ Self-determined**

By before the due date, on Friday 2nd June 2023:

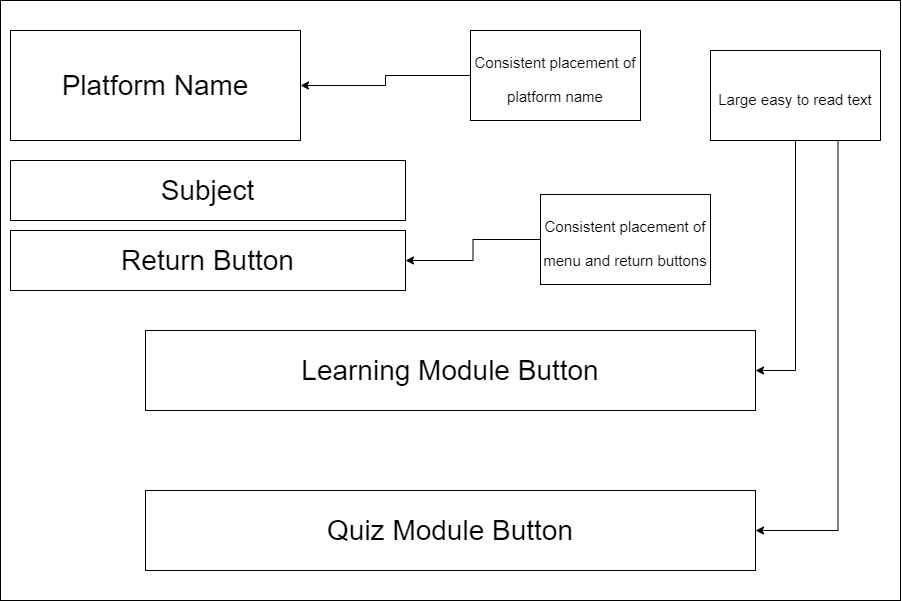
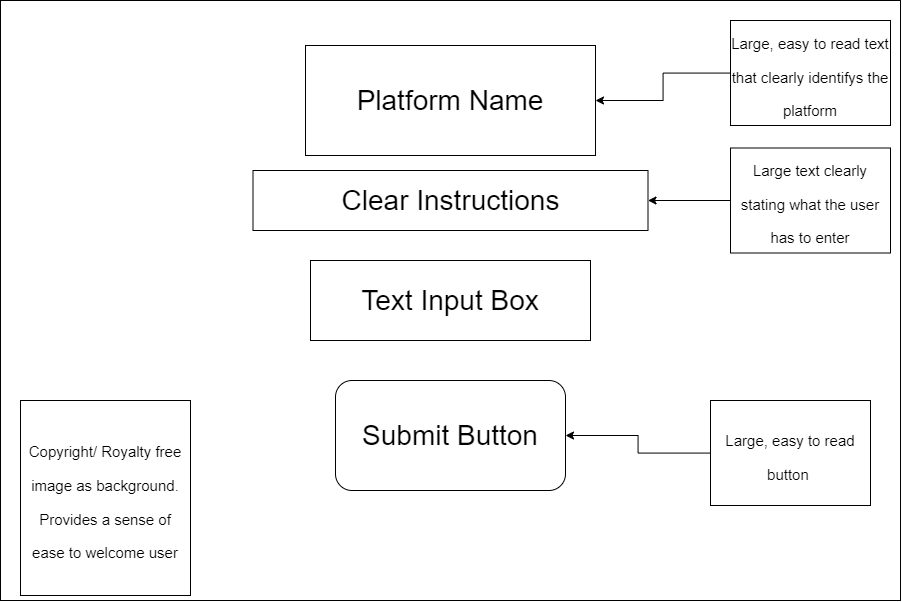
1. (All 6 of the above)
2. Have completed building an entirely new PHP quiz program that works entirely on 1 page.

* This allows for easy copy/ pasting for the next modules on ’The Learning Hub’ Platform.

1. Personally understand and be familiarised with PHP, HTML, & CSS coding rules/ systems.
2. Have commented, indented, and appropriately used whitespace on all pages, specifically pages with complex PHP systems.
3. Have the basic UI (User Interface) built with room to improve it later.
4. Have the UX (User Experience) design understood and built to be improved later.
5. Be designed and built with the target audience in mind.
6. Only used variables, cookies and other coding systems where needed in the most efficient manner possible.

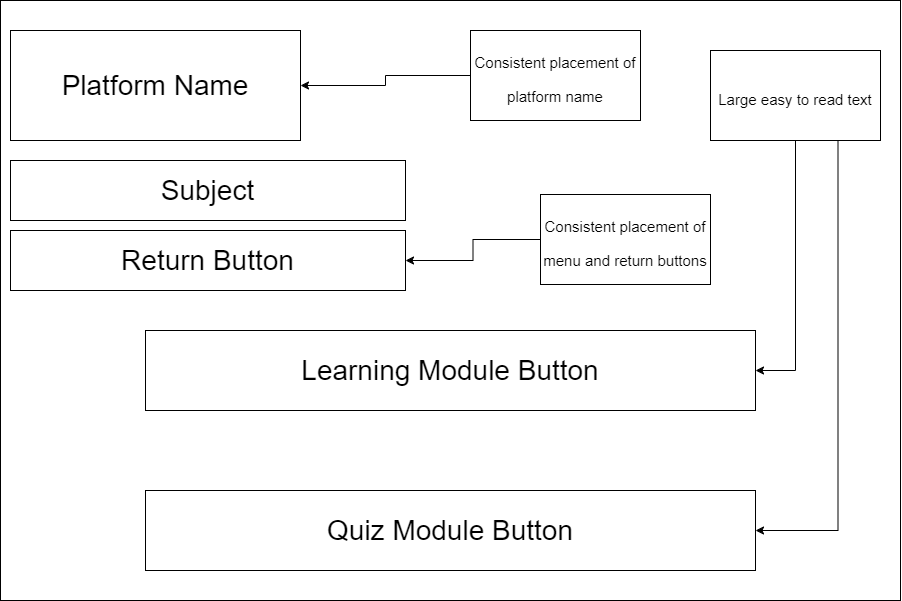
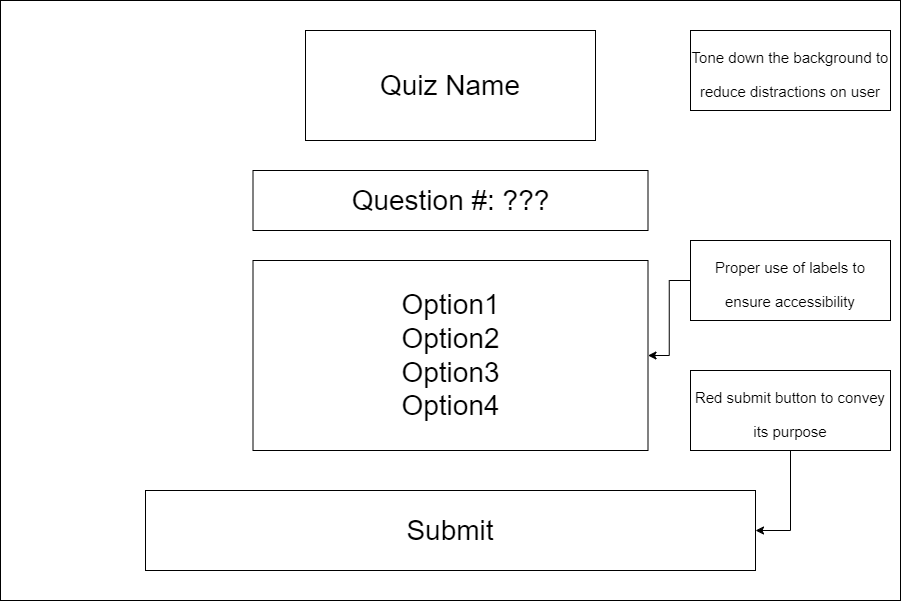
**DEVELOP- Wireframe**

A wireframe is a basic/ plain schematic blueprint of what the physical website will look like. They are used to assist in deciding what looks/ feels the best when it comes to UI and UX. Simple wireframes contain rough positioning of all the elements that will be on each page. Even simple wireframes are useful tools that assist when applying cascading style sheets (CSS) to a webpage to improve the aesthetic view of the webpage or platform which helps improve user experience.



All pages have a mainly white with a partial blue art background. There is a consistent theme of mostly white for the background, with black text and blue backgrounds for buttons. This creates contrast on the webpage.

All text is in Comic Sans, this is to appeal to the target audience. The platform name is consistently in a large font for accessibility. All text is considerably large.



**DEVELOP- Algorithm**

**login.php**

SET session var question to 1

SET session var Score to 0

SET session var MATHscore to 0

SET session var GEOscore to 0

SET session var ans1-11 to “”

**Page1\_quiz.php & page2\_quiz.php**

IF session var question equals 1

ECHO (1st Quiz Form)

The Quiz code is fundamentally the same from the 2nd IF statement onwards. Each consists of echoing/ printing the form for the current question while increasing the question variable which allows the user to move on to the next question. It most importantly assigns the POST value of the previous form into its corresponding session variable.

SET session var question to 2

ENDIF

IF session var question equals 2

ECHO (2nd Quiz Form)

SET session var question to 3

session var ans1 = POST 1

ENDIF

IF session var question equals 3

ECHO (3rd Quiz Form)

SET session var question to 4

Despite the quiz actually only being 10 questions long, the variable ticks over to 11 to process the POST information from question 10. It also serves as a convenient spot to place all of the marking code, and consequently the score page.

session var ans2 = POST 2

ENDIF

IF session var question equals 11

session var ans10 = POST 11

IF session var ans1 equals A

session var score +1

ENDIF

IF session var ans2 equals B

session var score +1

The question 11 IF statement also allows for scores to be moved from the score session variable into their own personal ‘SUBJECTscore’ variable. This is important as the score variable is universal, being used for all quiz pages. In turn meaning the current scores would be lost when the user starts a new quiz.

ENDIF

session var SUBJECTpercent equals session var Score divided by 10, multiped by 100

session var SUBJECTscore equals session var Score

Print (Ending Screen)

IF session var SUBJECTscore equal to or higher than 9

ECHO (Very Positive message) & Main menu button

ENDIF

Too add complexity to the code, the SUBJECTscore was broken apart, and sperate messages were produced depending on the user’s score. When scoring low, the user is recommended to take the quiz again, conversely, if they are sufficiently competent, they are only given a Main menu button

IF session var SUBJECTscore between 6 & 8

ECHO (Positive message) & Main menu button \* Retry button

ENDIF

IF session var SUBJECTscore equals Less than 6

ECHO (Encouraging message) & Main menu button \* Retry button

ENDIF

ENDIF

**GENERATE- Code (The code in this section is snippets that contain important elements such as php and forms)**

**Login.php**

<?php

session\_start();

$\_SESSION["Score"] = 0; //Score Var

//These variables must be delcared on this page to stop it messing it up on the Quiz page!

$\_SESSION["question"] = 1;

//Subject Scores

$\_SESSION["MATHscore"] = 0;

$\_SESSION["GEOscore"] = 0;

There are 11 ‘ans’, 1 ‘Score’, and 2 ‘(SUBJECT)score’ session variables, declared with blank values.

There is also 1 ‘question’ session variable set to 1. This is so that when the quiz page is opened, the quiz runs.

These variables are declared on a separate page for the simplicity of stopping them from getting reset every time the quiz script is run.

//Answer Scores

$\_SESSION["ans1"] = "";

$\_SESSION["ans2"] = "";

$\_SESSION["ans3"] = "";

$\_SESSION["ans4"] = "";

$\_SESSION["ans5"] = "";

$\_SESSION["ans6"] = "";

$\_SESSION["ans7"] = "";

$\_SESSION["ans8"] = "";

$\_SESSION["ans9"] = "";

$\_SESSION["ans10"] = "";

$\_SESSION["ans11"] = "";

?>

<html lang="en-AU">

<head>

<title>Login | TLH</title>

<link rel="stylesheet" href="stylesheet.css">

</head>

<body>

<!--Login Form-->

<form name="login" method="post" action="home.php" class="center" id="style2"> <!--Main Form-->

<div style="size">

<h2 class="style2" for="login">Please enter your name</h2>

<input type="text" id="login" name="name" required="true" align="center"><br><br>

<input type="submit" value="Submit" class="btn">

</form>

</body>

</html>

**Home.php**

<?php

session\_start();

//Sets User name and stores in cookie

if (isset($\_POST['name'])) {

$\_SESSION["SESname"] = $\_POST["name"]; //Assigns POST to SES VAR (Message references SES VAR, Not the cookie!)

setcookie("NAME", $\_SESSION["SESname"], time() + 86400\*5);

}

This if statement checks if the user has entered their name on the login page. It then assigns the POST variable to the Name session variable. It then stores the name inside of a 5-day cookie.

//If session VAR is killed, this checks cookie for data, that then re assigns the cookie data to session VAR.

if (!isset($\_SESSION["SESname"])) {

if (isset($\_COOKIE["NAME"])) {

$\_SESSION["SESname"] = $\_COOKIE["NAME"];

}

If the session is terminated, this if statement checks if the Name session variable is set, if it isn’t, it checks if the Name cookie is set. If the cookie is set, its sets the value of the Name variable to the Name cookie

}

//If A user comes back after 1 day, The cookie keeps the data, but the session Variable score will need to be set if they dont go through the login page.

if (!isset($\_SESSION["Score"])) {

$\_SESSION["Score"] = 0;

}

//Cookie //86400 is 1 day in Seconds

if ($\_SESSION["Score"] > 1) {

if ($\_SESSION["GEOscore"] > 1) {

setcookie("GEOscore", $\_SESSION["GEOscore"], time() + 86400\*5); // Sets GEO Cookie

}

if ($\_SESSION["MATHscore"] > 1) {

setcookie("MATHscore", $\_SESSION["MATHscore"], time() + 86400\*5); // Sets MATH cookie

}

This If statement checks if the session variable Score has a value, this meaning the user has completed one of the quizzes (Assuming they get at least 1 question correct). The nested If’s check each individual SUBJECTscore variable, then assigns their values to their respective cookies (5 day expiry time).

}

//Following 2 Reassign scores from cookies if cookies are set.

if (!isset($\_SESSION["GEOscore"])) { //GEOGRAPHY

if (isset($\_COOKIE["GEOscore"]) > 1) {

$\_SESSION["GEOscore"] = $\_COOKIE["GEOscore"];

}

}

if (!isset($\_SESSION["MATHscore"])) { //MATHS

if (isset($\_COOKIE["MATHscore"]) > 1) {

$\_SESSION["MATHscore"] = $\_COOKIE["MATHscore"];

}

These IF statements check if the SUBJECTscore’s variables have been set. If they haven’t been, another IF checks if their respective cookies have been. If the respective cookies have been set, the respective session variables for each SUBJECTscore are reassigned a value from the cookies stored data.

}

?>

<!DOCTYPE html>

<html lang="en-AU">

<head>

<title>The Learning Hub</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="stylesheet.css?v=3">

<style>

body {

background-image: url('https://wallpapercave.com/wp/wp3404269.jpg');

background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;

}

Inside the head tag any important metadata is stored. The CSS stylesheet is also referenced here. The style tag is used to change CSS elements only on this page, separate from the main stylesheet, in this case, it is a home screen unique background image.

</style>

</head>

<body>

<header>

<div class="style1">

<h1 class="style1">The Learning Hub</h1>

<h2 class="style1">Welcome, <?php echo $\_SESSION['SESname'];?>! </h2>

</div>

Simple echo statement used within a H2 tag to print the Users name.

<nav>

<div class="btn-group">

<a href="page1.php"><button class="button">GEOGRAPHY</button></a>

<a href="page2.php"><button class="button">MATH</button></a>

<a href=""><button class="button" disabled="true" id="dis">English!</button></a>

<a href=""><button class="button" disabled="true" id="dis">Coming soon!</button></a><br><br><br>

</div>

The Nav tag is used to state the navigation panel for the webpage.

</nav>

</header>

<br><br><br><br>

<?php

if (isset($\_COOKIE["GEOscore"]) || isset($\_COOKIE["MATHscore"])) { //Add each possible cookie here

if (isset($\_COOKIE["GEOscore"]) || isset($\_COOKIE["MATHscore"])) {

echo '<p>Return to The Learning Hub within 5 days to keep your scores!</p>';

}

These If statements check if any of the SUBJECTscore cookies have been set. If they have it means the user has completed at least one quiz. This then changes the starting message for a new one. It echoes a warning to the user that their scores will remain if they return to the platform within 5 days.

if (isset($\_COOKIE["GEOscore"])) { //Write score for specific cookie

echo '<p>GEOGRAPHY: '. $\_COOKIE["GEOscore"]/10\*100 . '% </p>';

}

if (isset($\_COOKIE["MATHscore"])) {

echo '<p>MATHS: '. $\_COOKIE["MATHscore"]/10\*100 . '% </p>';

}

These If statements check which individual cookies have been set. For each respective cookie it echoes out its score along with some text. This text displays what percentage the user got on each respective quiz.

} else {

echo ("

<p>Welcome to The Learning Hub! <br> Select one of the modules above to begin!</p> <br>

<p>Inside each module will be a learning resource called 'LEARN!' <br> Once you finish the learning resource you can check your knowledge with the 'QUIZ!'</p> ");

}

This else statement is run first as initially none of the quizzes have been completed (So none of the cookies have been set). This echo statement is a brief overview of how The Learning Hub Works. This message is replaced with completed quiz scores once the user completes a quiz.

?>

</body>

</html>

**Page1.php &**

**Page2.php etc**

<?php

session\_start();

//Ensures that the Quiz is set to 1

$\_SESSION["question"] = 1;

?>

The question session variable is set to 1 to ensure when the user continues to the Quiz page everything works as intended.

<?php

if (isset($\_COOKIE["GEOscore"])) {

if ($\_COOKIE["GEOscore"] > 1) {

echo ("<h4>You have have already taken this Quiz!</h4>");

}

}

?>

This If statement checks if the cookie for each respective quiz prepage. If the cookie has been set, it echoes out a message notifying the user that they have already attempted this quiz.

**Page1\_quiz.php &**

**Page2\_quiz.php etc**

<?php

session\_start();

$\_SESSION["Score"] = 0; // Sets Score to 0

?>

To begin the “Quiz Code” the session variable Score must be reset. This ensures there is no further issues with the code.

<!DOCTYPE html>

<html>

<head>

<title>Continents Quiz!</title>

<link rel="stylesheet" href="quiz\_style.css">

</head>

<body>

<h1>Continents Quiz!</h1>

<br><br>

<?php

if ($\_SESSION["question"] == 11) {

$\_SESSION["ans10"] = $\_POST["quiz10"]; //Posts Data

Once the user completes question 10, the question variable becomes 11. This If statement commits the final POST variable to the ans10 session variable. Inside the main IF statement (Variable question equals 11) it begins the final process, which includes marking, and echoing scores and their respective messages, and echoing the button options.

//Checks each answer

if ($\_SESSION["ans1"] == "B") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans2"] == "C") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans3"] == "D") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans4"] == "A") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans5"] == "A") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans6"] == "D") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans7"] == "B") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans8"] == "C") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans9"] == "A") {

$\_SESSION["Score"]++;

}

if ($\_SESSION["ans10"] == "B") {

$\_SESSION["Score"]++;

}

These if statements check whether the answer the user chose on the forms is correct or not. If the answer is correct, it increases the score session variable.

//Creates session Variable that hold GEO Score

$\_SESSION['GEOpercent'] = $\_SESSION['Score']/10\*100;

$\_SESSION['GEOscore'] = $\_SESSION['Score'];

This IF statement commits the final Score session variable to its own “SUBJECTscore” variable. Giving each subject its own session variable removes the issue of using the Score variable over multiple quiz pages and lots of over issues when it comes to the cookies and the homepage.

// Screen items (Score)

printf ("<h1>Score</h1>");

echo '<h2>You scored '. $\_SESSION['Score'] . ' out of a possible 10 ('. $\_SESSION['GEOpercent'] . '%)</h2>';

This section of code echoes the top half of the ‘score’ page. It provides the user with their mark out of 10 and the equivalent percentage.

if ($\_SESSION['GEOscore'] >= 9 ) {

echo '<h2> Great job!</h2>';

//Main Menu

printf("

<br><br>

<!--Button asking if the user wants to return main menu.-->

<form name='Return Main Menu' method='post' action='home.php' id='style1'>

<input align='center' name='Return Main Menu' type='Submit' value='Return Main Menu' id='btn'>

</form> ");

This IF statement (and the following two) provides the user with varying end-screen messages and button options. If the user scores 9 or above, it congratulates them and only provides them with a return to the main menu button. At 90% and above the student is well above competent in the chosen area of study., needing no further assistance.

} elseif ($\_SESSION['GEOscore'] >= 6 && $\_SESSION['GEOscore'] <= 8) {

echo '<h2>Good job! Remember there is always room for improvment!<h2>';

//Retry

printf("

<!--Button asking if the user wants to retry the quiz.-->

<form name='RestartQuiz' method='post' action='page1.php' id='style1'>

<input align='center' name='RestartQuiz' type='Submit' value='Try Again?' id='btn'>

</form>");

//Main Menu

printf("

<br>

<!--Button asking if the user wants to return main menu.-->

<form name='Return Main Menu' method='post' action='home.php' id='style1'>

<input align='center' name='Return Main Menu' type='Submit' value='Return Main Menu' id='btn'>

</form> ");

This If statement is run if the user scores 6, 7, or 8. It gives them an encouraging message that they are doing good, but they can do better. At this level they are given a return to the main menu button as well as a button to retry the Quiz.

} else {

echo '<h2>(╯°□°)╯You might want to try again! ノ( º \_ ºノ)</h2>';

//Retry

printf("

<!--Button asking if the user wants to retry the quiz.-->

<form name='RestartQuiz' method='post' action='page1.php' id='style1'>

<input align='center' name='RestartQuiz' type='Submit' value='Try Again?' id='btn'>

</form>");

//Main Menu

printf("

<br><br>

<!--Button asking if the user wants to return main menu.-->

<form name='Return Main Menu' method='post' action='home.php' id='style1'>

<input align='center' name='Return Main Menu' type='Submit' value='Return Main Menu' id='btn'>

</form> ");

}

}

?>

This last else statement is run if the user scores 5 or less. The provided message informs the user that their score wasn’t too good and encourages them to retake the quiz. Similar to the previous level, they are provided with a return to the main menu and a retry button.

<!--Form ---------------------------------------------------------------------------------------------------------------------------------->

<?php

// Question 1

if ($\_SESSION["question"] == 1) {

echo('

<form action="page1\_quiz.php" method="post" id="style1">

<h3>Question 1: How many continents are there?</h3>

<input type="radio" id="A" name="quiz1" value="A" required>

<label for="A"> A) 6 </label><br>

<input type="radio" id="B" name="quiz1" value="B">

<label for="B"> B) 7 </label><br>

<input type="radio" id="C" name="quiz1" value="C">

<label for="C"> C) 8 </label><br>

<input type="radio" id="D" name="quiz1" value="D">

<label for="D"> D) 9 </label>

<br><br>

<input type="submit" value="Submit" class="submitbtn" id="btn">

</form>

');

$\_SESSION["question"] = 2;

This If statement is the first one run when begging the quiz. It echoes out a form which contains a question and 4 possible answers. The session variable question is set to 2. When the user submits the form, it will echo the next form instead of this one.

// Question 2

} elseif ($\_SESSION["question"] == 2) {

echo('

<form action="page1\_quiz.php" method="post" id="style1">

<h3>Question 2: What continent starts with an "E"?</h3>

<input type="radio" id="A" name="quiz2" value="A" required>

<label for="A"> A) Eurip </label><br>

<input type="radio" id="B" name="quiz2" value="B">

<label for="B"> B) Entarctica </label><br>

<input type="radio" id="C" name="quiz2" value="C">

<label for="C"> C) Europe </label><br>

<input type="radio" id="D" name="quiz2" value="D">

<label for="D"> D) Esia </label>

<br><br>

<input type="submit" value="Submit" class="submitbtn" id="btn">

</form>

');

$\_SESSION["question"] = 3;

$\_SESSION["ans1"] = $\_POST["quiz1"];

The If statement for the 2nd question is identical to the one for the first question. Except at the end, it also contains an assignment operator that takes the POST variable data from the previous question and assigns it to its respective ans variable. This is because the form data can only be collected after the form is submitted, creating a 1 behind delay in commenting the post data to its ans variable. This is the reason behind having question 11. It is convenient to have it as an ending screen and to use it to commit the last POST variable to ans10.

// Question 3

} elseif ($\_SESSION["question"] == 3) {

echo('

<form action="page1\_quiz.php" method="post" id="style1">

<h3>Question 3: What continents have North & South at the begnning?</h3>

<input type="radio" id="A" name="quiz3" value="A" required>

<label for="A"> A) Australia </label><br>

<input type="radio" id="B" name="quiz3" value="B">

<label for="B"> B) Asia </label><br>

<input type="radio" id="C" name="quiz3" value="C">

<label for="C"> C) Europe </label><br>

<input type="radio" id="D" name="quiz3" value="D">

<label for="D"> D) America </label>

<br><br>

<input type="submit" value="Submit" class="submitbtn" id="btn">

</form>

');

$\_SESSION["question"] = 4;

$\_SESSION["ans2"] = $\_POST["quiz2"];

//Question 10

} elseif ($\_SESSION["question"] == 10) {

echo('

<form action="page1\_quiz.php" method="post" id="style1">

<h3>Question 10: What continent has the country China?</h3>

<input type="radio" id="A" name="quiz10" value="A" required>

<label for="A"> A) Australia </label><br>

<input type="radio" id="B" name="quiz10" value="B">

<label for="B"> B) Asia </label><br>

<input type="radio" id="C" name="quiz10" value="C">

<label for="C"> C) America </label><br>

<input type="radio" id="D" name="quiz10" value="D">

<label for="D"> D) Europe </label>

<br><br>

<input type="submit" value="Submit" class="submitbtn" id="btn">

</form>

');

$\_SESSION["question"] = 11;

$\_SESSION["ans9"] = $\_POST["quiz9"];

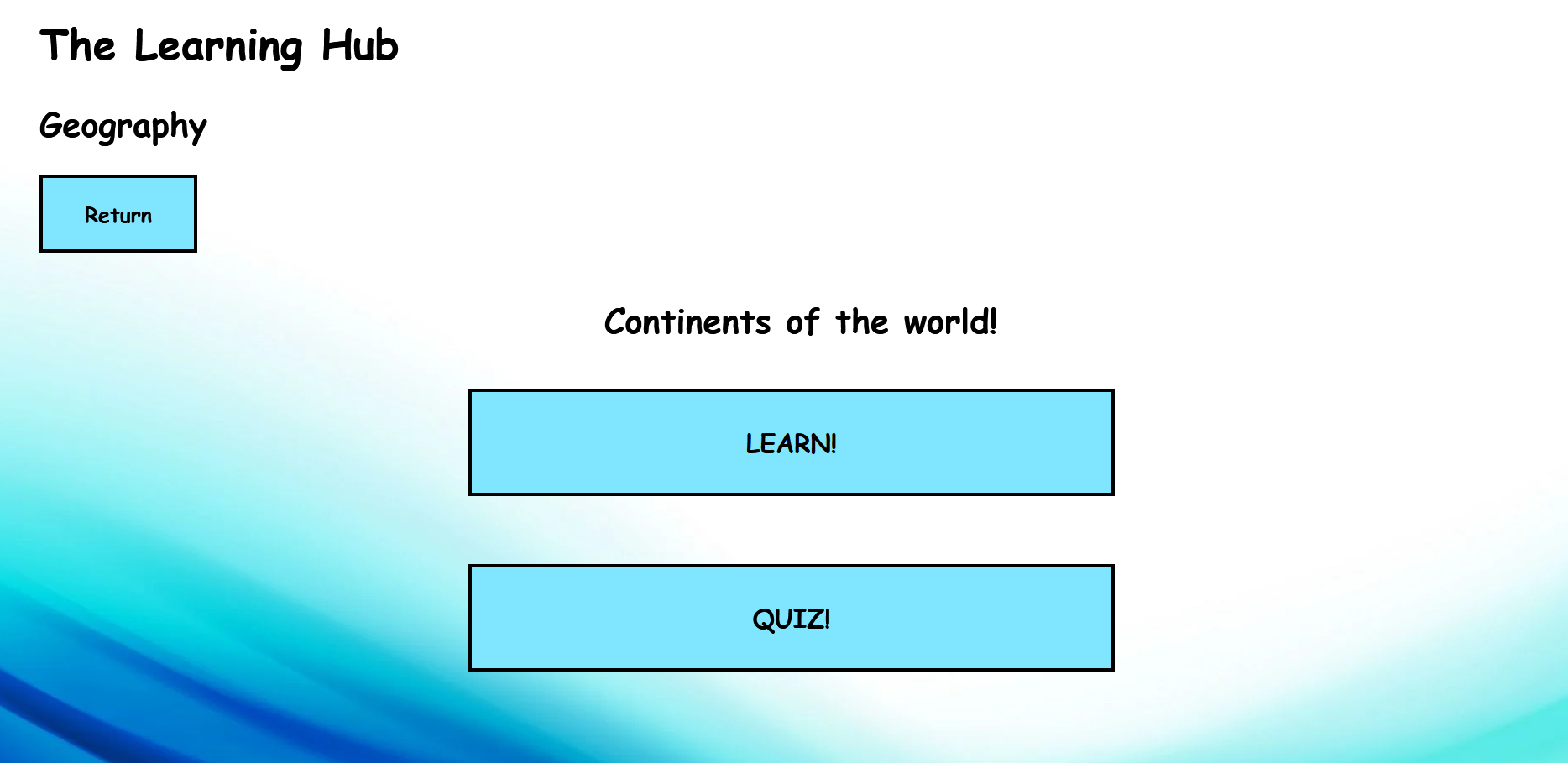
}

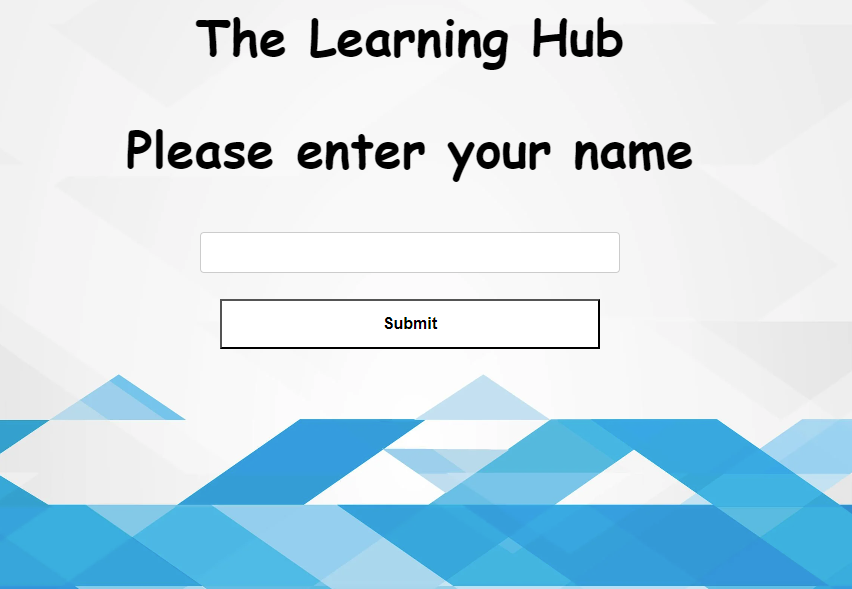
?>

</body>

</html>

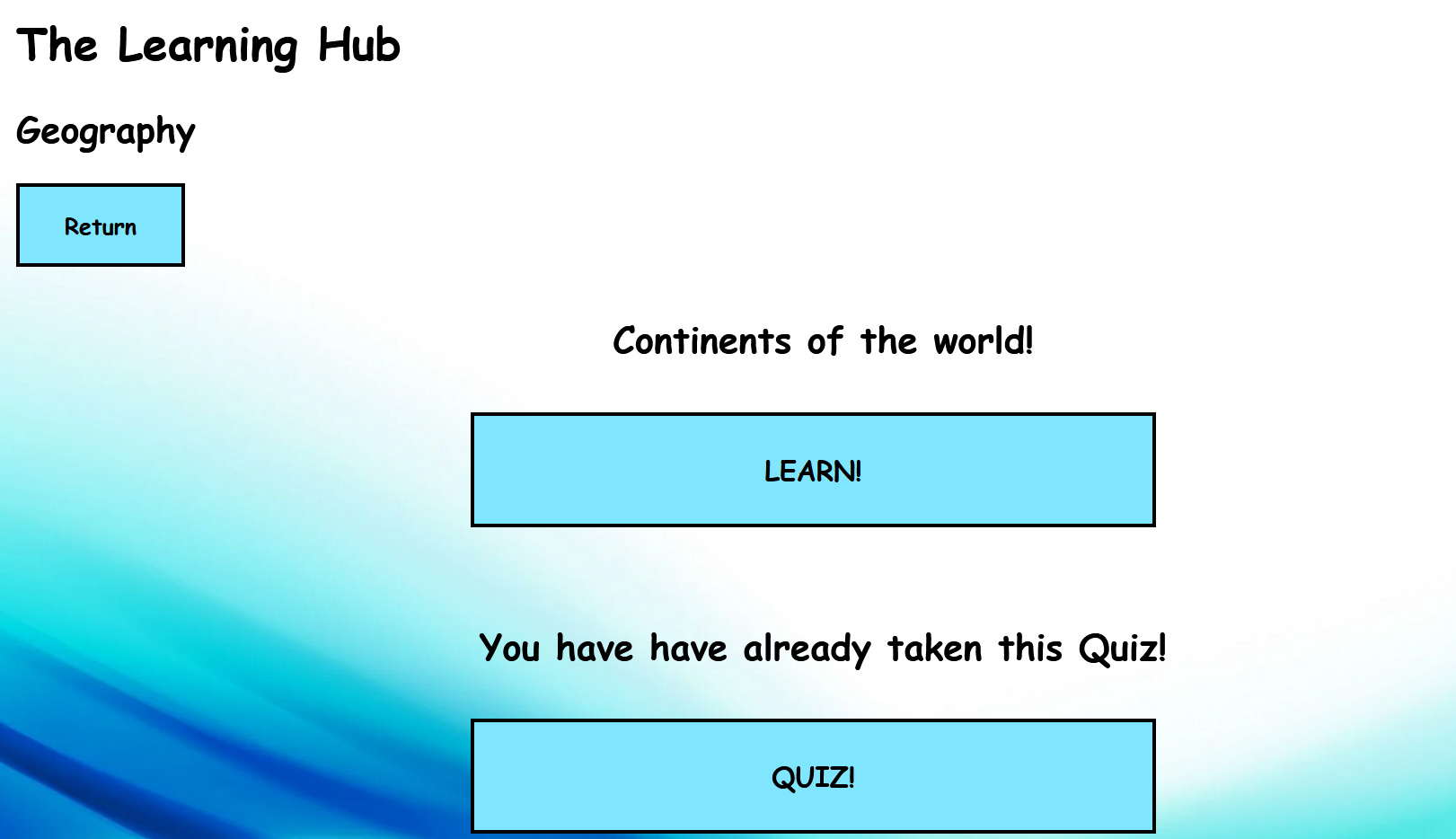
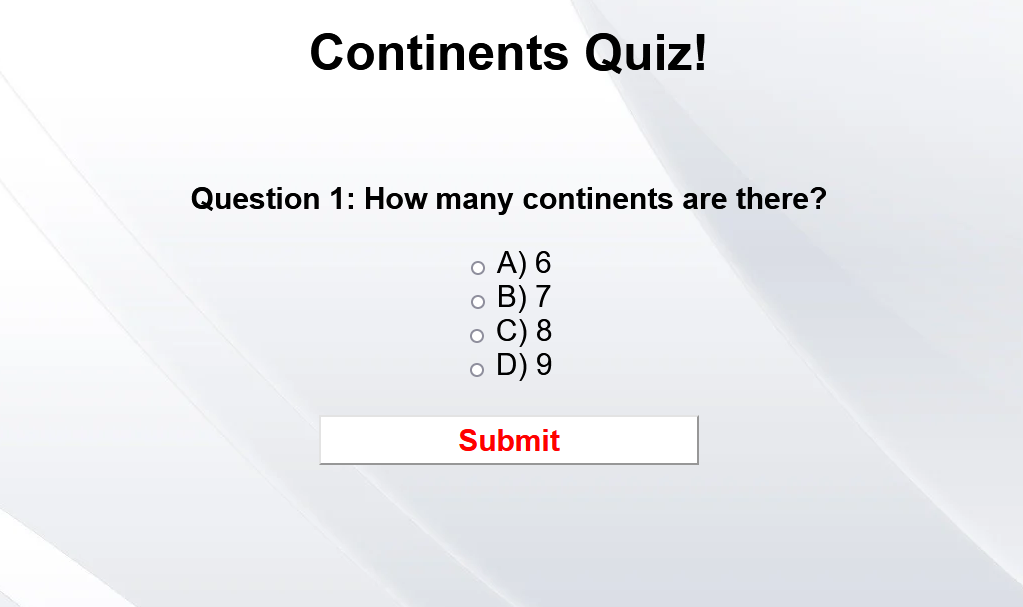
**GENERATE- User Interfaces**





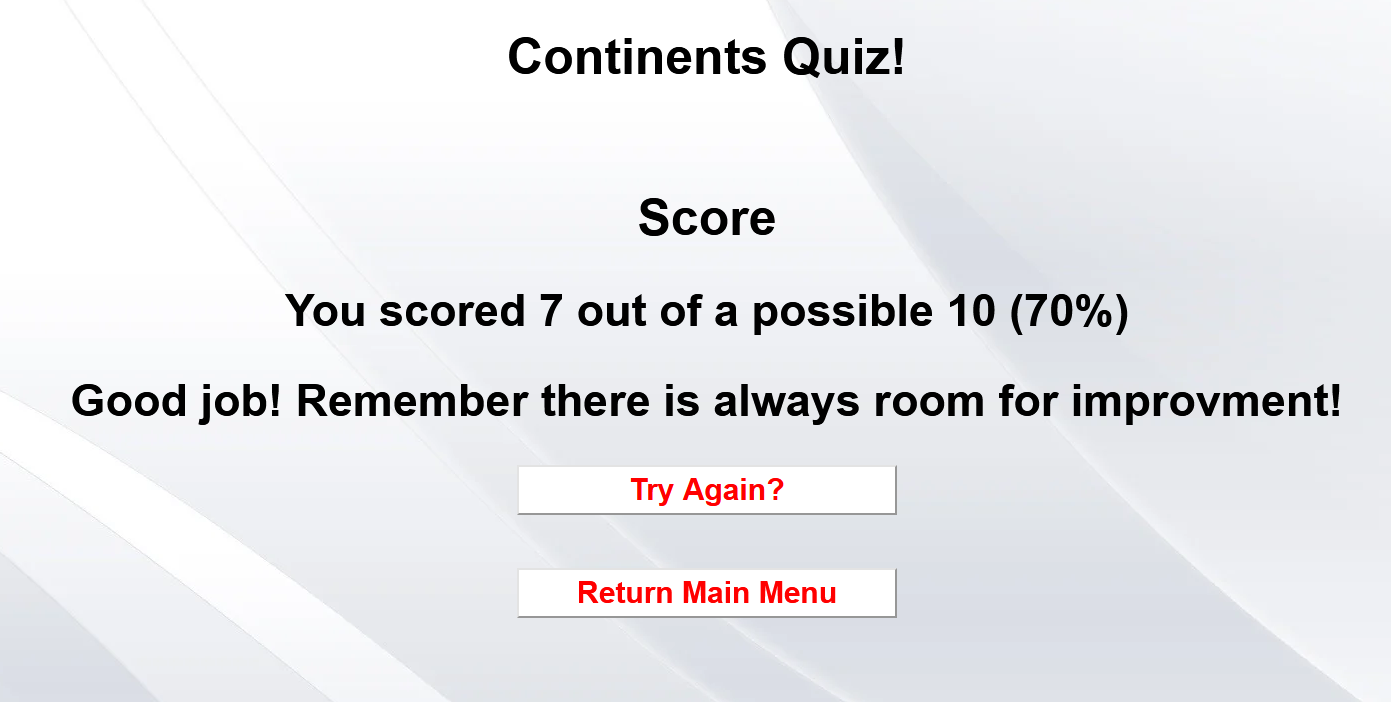
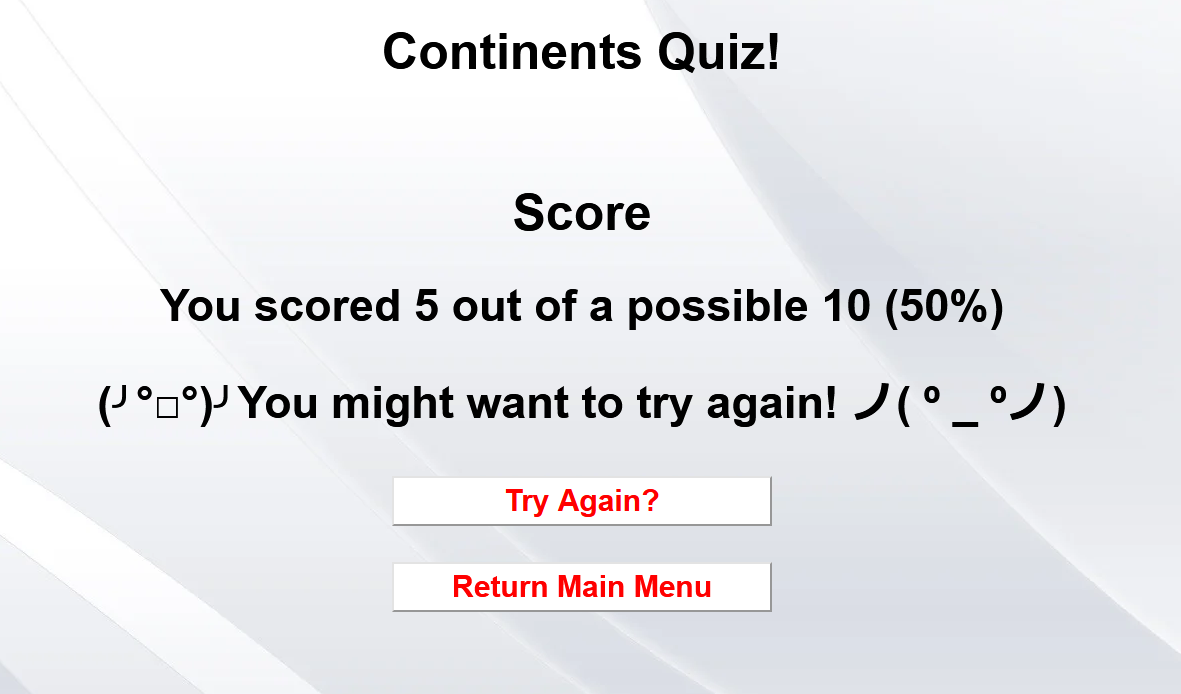
Simple ‘in between page’. Too many cause unnecessary confusion regarding navigation, having too little makes the site feel small and empty.

Open and friendly design. Image used to calm and welcome users. Big text identifies the platform.

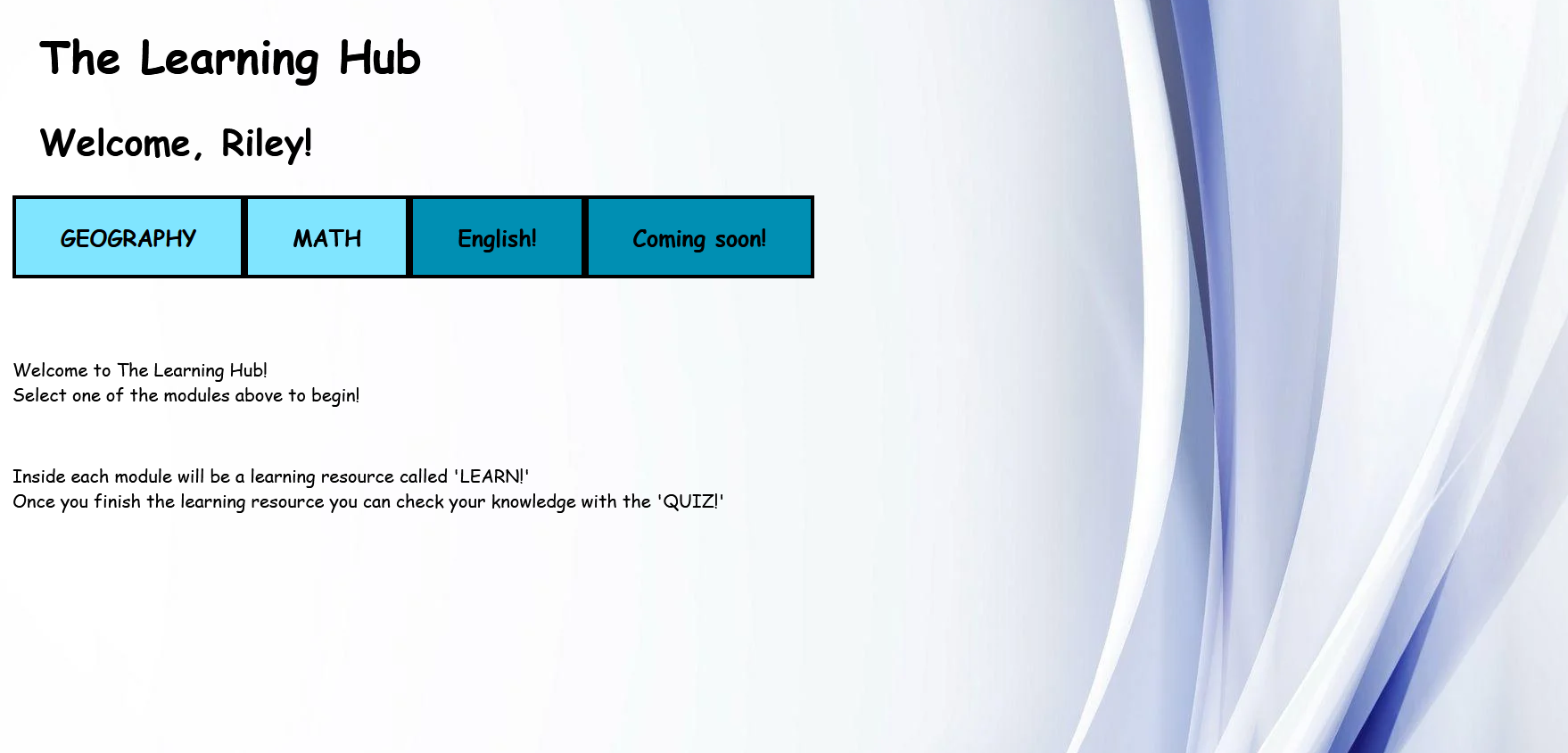
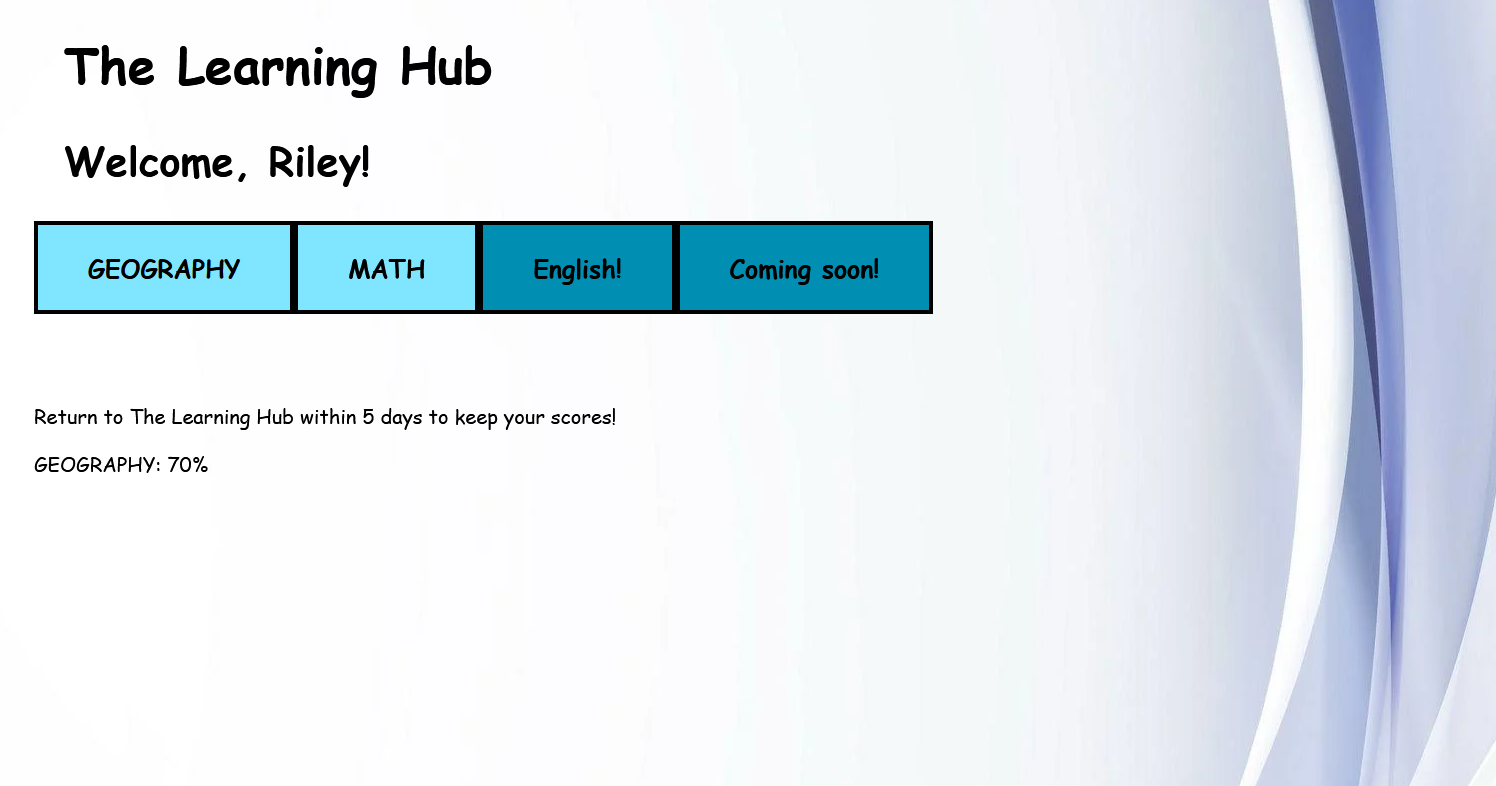


Simple and repetitive design that is easily learnable for younger children. While working functionally and practically.

Once the user attempts a quiz, a message will pop up notifying them that they have already attempted this quiz. This can stop confusion if the user clicks on the wrong subject.



Varying messages are displayed for achieving different scores on the test. Different button options are also presented if the User scores high enough. This way it congratulates higher achievers and encourages less abled students.



When the user completes a quiz, the bottom half of the home page displays their quiz scores. This allows the user to easily track their progress from the homepage.

Personalised message to welcome/ greet users. Easy to understand UI with simple instructions below for those who need them. Colourful background to fill in large blank space.

**EVALUATION- Evaluation against criteria**

To finalise this digital solution, I ineptly followed the problem-solving phases of explore, develop, generate, and evaluate/refine. The explore phase was used to fully understand the issue in full and assisted in generating relevant criteria that guided me throughout this solution.

The prescribed criteria have been met and a fully functioning and working prototype/draft platform (The Learning Hub) have been produced.

The interactive learning object, the browser game works as intended. Users have to login, where they are greeted with a personal greeting message. Students are then allowed to independently choose from a variety of learning modules (Currently only the geography module is available). From there students go through an interactive learning section that contains images and contrasting colours, as well as the design choices selected for the target audience. Students are then able to take a quick quiz on the subject they just studied to test their knowledge. The questions progressively become more difficult with the final two not having been clearly stated on the learning resource. This prompts students to use their own intuition to solve the question, similar to an A-level question on an exam that is ‘complex unfamiliar’.

To achieve all this, I planned out the User Interface (UI) and User Experience (UX) with wireframes. This allowed me to easily choose a design that aligns with the goals of this solution that meet the requirements (such as the target audience). After the planning/ design phase, I used HTML, CSS, and then further PHP to program the Platform mechanics and systems. Using decomposition, I was able to break the problem into multiple steps. The first step was to redesign the quiz program (Honestly the most taxing and time-confusing process). Reprogramming the quiz program allowed it to be more streamlined and easier to integrate into other processes and systems and allowed for easy ‘add-ons’. The next few steps were minor tasks compared to reprogramming the quiz code, these included building a login page, an interactive home page, and the learning resources/ sections.

Furthermore, my self-determined criteria required that I designed/ built the UI & UX to be friendly to the target audience (10-13-year-olds). For this, I ensured that the theme, colours (contrast), font size and styles, and placement of items were appropriate and familiar. The design was ensured to be open to everyone, as grades 5 & 6 are leading into high school and 7 are in high school, it is required to have a child-friendly feel, but have a more independent feel for those who are further along in social and mental development. The overall UI serves as a good learning tool for the slightly younger users about how modules and sections on web platforms work.

Lastly, efficient PHP code was produced to ensure that the entire platform works as intended and withholds the function stated in the criteria such as being modular and easy to add on to. By meeting this criterion, it ensures the code is easy to work on in future and easy for other developers to look at and pick up.

**EVALUATION- Impacts (Personal, Social, Economic)**

Personal impacts are those that would impact the user of the platform at a personal level. Users who spend too much time playing games, such as this can become socially withdrawn and lead to mental health issues. Just like any game, time spent using the platform should be used within judgment.

Social impacts are those that will be caused by the solution’s influence on society. This solution is designed to assist young students to firm up their knowledge before they enter high school. Students that go into high school with a good base knowledge will be able to progress faster than their peers, thus leading to more successful people in industries after education.

Economic impacts are any monetary restrictions that could inhibit users from accessing the learning platform. Although the game is currently free, the user is required to have a computer or mobile device with an internet connection which may limit low socioeconomic people from accessing this resource, further creating a gap in education (as mentioned in social impacts.)

**EVALUATION- Recommendations**

Currently, the most needed improvements would be to the overall UI, such as fixing colouring, positioning, margins and spacing. The platform also needs further ‘fool’ proofing to prevent any necessary errors, such as being able to click enter on the quiz without any option being selected.

To create a proper finished model of ‘The Learning Hub’ there would need to be the introduction of Databases to store students’ results and create classes for teachers so that student improvement can be tracked over time. This would allow teachers to better understand what subject areas specific students are struggling in.

Finally, this game is accessible to any users with an appropriate device and an internet connection. While there are social and personal impacts to be considered, simple moderation can easily balance out the negatives to create a positive learning tool to assist children in learning, and further prepare them for entering high school.

**REFERENCES**

H5P n.d., www.uwindsor.ca, viewed 26 May 2023, <https://www.uwindsor.ca/openlearning/509/interactive-learning-objects>.

Coştu, S, Aydın, S & Filiz, M 2009, ‘Students’ conceptions about browser-game-based learning in mathematics education: TTNetvitamin case’, Procedia - Social and Behavioral Sciences, vol. 1, no. 1, pp. 1848–1852.

Mathematics n.d., [www.australiancurriculum.edu.au](http://www.australiancurriculum.edu.au).