## Analysis

### Description of problem

I will be developing an online quiz website. On this website, registered users will be able to compete against each other in a leaderboard on one multiple-choice quiz.

The website will have:

* Account facilities - allowing users to register, log into and out of accounts
* A leaderboard - a table which will display the ranking of each user based on their score and the time completed
* Error pages for when the user tries to access the quiz without an account, tries to access an account unsuccessfully and tries to register/log into an account when an account is already in use.
* The quiz page

The quiz will have:

* 10 questions with 4 possible answers.
* A set time limit of 90 seconds to complete the quiz
* A leaderboard, which will display each player’s score and the time they completed the quiz

The points allocated will depend on how many questions the user answers correctly and how quick the user is to finish the quiz. If the user finishes the quiz in:

* 70 seconds, multiply number of correct answers by 2
* 69-31 seconds, multiply number of correct answers by 1.5
* 30-10 seconds, multiply number of correct answers by 1.25
* 9 seconds or less, multiply number of correct answers by 1

All points will be rounded to omit any decimal places in the final score that is calculated.

Registered users will have their total score and the time for them to answer each question on the leaderboard. Users will be able to see how their rank compares to others who have taken the quiz.

The end-users of this quiz will be targeted towards S1-S6 students and their parents/guardians, as well as teachers who want to test their general knowledge. End-users should be relatively skilled in using and understanding the typical website.

My project should meet the advanced higher computing requirements as:

* When a user is making or logging into an account, a form will be used to post the inputs to be processed. PHP will be used to validate inputs.
* Session variables will be used to store the account that is in use so that it can be validated whether or not a registered user is attempting to take the quiz (and if the user is not on an account and is trying to take the quiz, they can be redirected to an error page)
* External CSS will be used to make a intuitive and clear user interface for users on both desktop and mobile devices. Media queries within the external CSS for the mobile and desktop interfaces.
* A database will be used to store question, leaderboard and user details.
* SQL queries will be used to retrieve questions and distractors, validate user details when registering/logging in and update and sort the leaderboard.

#### Scope

The project’s scope should cover and include:

1. A completed design to show the user interface for the website. This design will include pseudocode and wireframes in order to illustrate what the interface should look like when finalised. The design will also include a data dictionary and query designs for the database features.
2. A functional website that integrates database connection and queries successfully.
3. A completed test plan so the website can be tested to see if requirements have been fulfilled. Each test will be described and carried out using personas and test cases. Results will be given and described for each test.
4. An evaluation of the website based on its fitness for purpose, maintainability and robustness.

#### Constraints

Technical:

* PHP, Javascript and CSS will be used in order to create the quiz as I have built up my experience and skills in these programming languages throughout the computing science course
* XAAMP’s MySQL will store the database with the user, leaderboard and question details.
* The website will run on the XAAMP server so that server-side processing can be enabled
* The website will be developed and tested on a Windows operating system

Economic:

* All software that is being used to develop the solution is fully licensed and free-of-charge - meaning that there will be no financial costs when developing this project

Time:

* The project should before the SQA submission deadline so it can be checked and sent off in adequate time

#### Boundaries

The website should be able to:

* Register users with a user-inputted username and password
  + Users will not be able to share the same username. If a user tries to sign up with an already existing username, they will be redirected back to the register page with an error message on the page
* Log in already registered users when they submit their username and password
  + If the user tries to log in with an account that is not existing on the database, they will be redirected to an error page
* User inputs for the login and register facilities will be validated to make sure that they are filled out and only contain alphanumeric characters. If the data is unvalid, the user will be redirected back to the login/register page with an error message in the header
* Users who are already using an account will not be able to access the login and register facilities again until they log out of their account.
* Allow registered users to take the quiz
  + Unregistered users who try and enter the quiz page will be redirected to an error page
* Allow users to view the leaderboard at any time
* Do the quiz
  + The quiz will contain a countdown of 90 seconds - when the countdown ends, the form used to collect the user’s answers will be processed. A PHP session variable will be used to store the time it takes for the user to complete the quiz. The user cannot alter the countdown’s time.
  + The quiz will contain 10 questions with 4 possible choices. These questions will be shown individually to the user. Their visibility will be toggled by a button (javascript). Questions and the multiple choice answers will be shown in the same order for each iteration of the quiz.
  + Users will have their answers submitted either when the countdown reaches 0 or when they click the submit button to complete the quiz. The form storing the users’ answers will not be validated. The form does not have to be completed when submitted.
* Calculate the total score of the user using PHP to compare the users’ answers to the actual answers stored on the database. Use the session variable storing the users’ time in order to calculate the total score.
* Update the leaderboard:
  + If the registered user has not already taken the quiz, insert them into the leaderboard with their time and score.
  + If the registered user has already taken the quiz, update their existing time and score. This will be done regardless, even if the user has scored worse than they did last time.
* Sort the leaderboard:
  + The leaderboard will be ordered by the highest to lowest scores. If users share the same score, the user who completed the quiz the quickest will be shown first.
  + Assign a rank to each user - with the user at the top being assigned 1 and so on.
  + Show the user’s rank, their username, their total score and the time it took for them to complete the quiz
* Log out users:
  + If no account is in use, inform the user on the page that there is no account in use and do not perform the logout

### UML Use Case Diagram

### 

### Requirements Specification

#### Purpose

The purpose of this website is for registered users to be able to take a timed, multiple-choice quiz and see how their score compares to other users who have done the quiz on a leaderboard.

#### Functional Requirements

The website must be able to:

* Store user, leaderboard, and question details in an external database which can be queried and sorted
* Validate user inputs for the account facilities
* Provide a working navigation bar so users can easily navigate themselves around the website
* Show a different user interface based on whether the user is using a desktop or mobile device
* Authenticate that users:
  + If logging in, exist within the external database
  + If registering, the username they have inputted does not exist within the external database
  + If either of these conditions are not met, redirect to the appropriate error page
* Format the user interface based on the CSS in an external stylesheet
* Use session variables to store the current account in use
* Logout users from the account they are using. If they are not using an account, do not perform the logout.
* Enable registered users to take the quiz:
  + Redirect to the appropriate error page when users who are not registered try to access the quiz
  + Query the external database so that the questions and the choices display on the quiz page within a form
  + Include a PHP countdown to store the session variable for the time in which the user completes the quiz which will use elements of XML to access PHP code and a javascript countdown to display the countdown
  + When the countdown times out or when the user submits the quiz, calculate the score and update the leaderboard by querying the external database
  + Redirect the player to the leaderboard page when they have completed the quiz
* Enable users to view the leaderboard:
  + Sort the leaderboard details so that it is ordered by each user’s score and the time they completed it (both in descending order)
  + Display the username, score and time completed for each user on the leaderboard
  + For each user, give them a numbered rank in ascending order

#### End-user requirements

* Use a clear and legible font family, colour and size for readability
* Make use of white space for further readability
* Make each page similar in layout and design for visual clarity
* Use a pleasing and consistent colour palette for visual clarity
* Implement a intuitive navigation bar so the website is easy-to-navigate
* Make the functions of buttons, links, etc clear so users know what to expect when they use them

### 

#### Inputs:

For the quizzes:

* The user’s selected answers
* The user’s time to complete the quiz
* The question that the user has toggled visible

For the login:

* The user’s username
* The user’s password

#### Outputs:

Show the rank, total score, time completed and the username for each user on the leaderboard

* Error pages/messages when applicable
* Show the questions and the choices for answers on the quiz page
* Show the question that the user has toggled visible
* Show the time the user has left to complete the quiz on the quiz page
* Show the user’s username on the home page to show which account is in use

#### Processes:

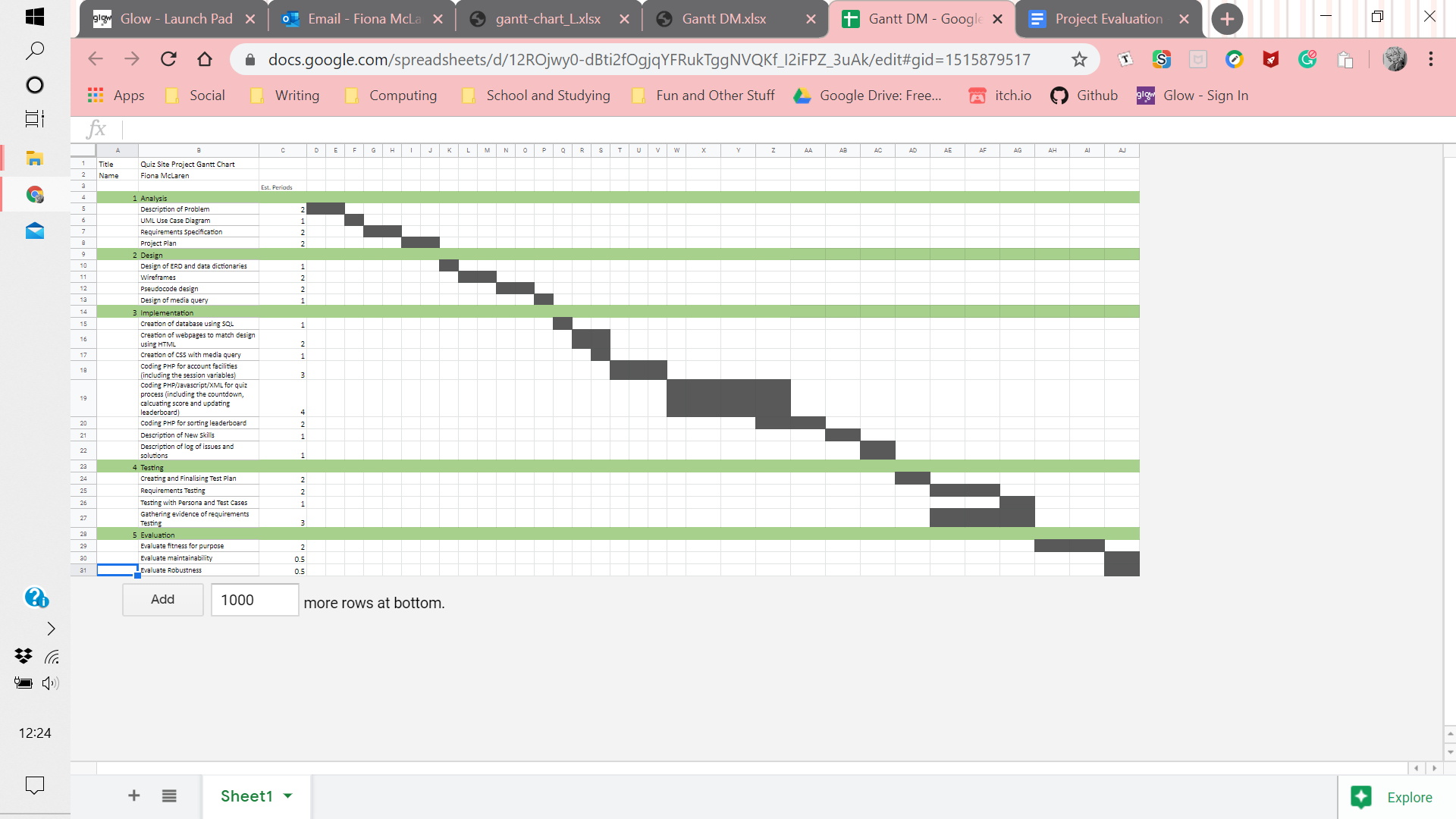
* Calculate time for user to complete the quiz

o Award points based on the time it took for the user to complete the quiz and how many answers they got correct

* Get all the questions and the choices for answers via a query so it can then be displayed on the page
* For each question, place within a <div> element which can be toggled visible when the user clicks on the corresponding button
* Submit the form when either the user has submitted it themselves or when the countdown times out
* Calculate the total score of the user and add it to the leaderboard via update or insert
* Sort all the total scores and the time completed on the leaderboard in descending order. Number the position of each score.
* Validate the username and password of registered users logging in
* Validate the username and password of unregistered users registering

### PROJECT PLAN

#### Tasks and Estimate of Timings



|  |  |
| --- | --- |
| Task | Est. Periods |
| Analysis | 7 |
| Design | 7 |
| Implementation | 15 |
| Testing | 8 |
| Evaluation | 3 |

### Resources

Computer: Windows Operating System

Software:

* XAMPP (Apache and MySQL)
* Notepad++ (for PHP, Javascript, HTML, CSS and XML)
* Google Chrome (for viewing an testing the website)