Fun Quizzes For You

And Your Whole Family And Friends

About This Project

Technologies used: PHP, MySQL, HTML, CSS, Bootstrap and JavaScript

Introduction

This is a PHP project to create a website with Entertainment and Educational purposes, targeting a broad range of audiences (pretty much anyone that can read and write), and it consists of the following parts:

1. Database

The database has 2 tables: quiz and quiz_topic which have a one-to-many relationship. There are 2 functions in the PHP code that query the database, one queries all the available topics to be displayed on the main page, and the other queries a certain number of questions and answers for a specific topic by joining the two tables.

Security and Error Handling: If the query result is empty, the user is redirected to the Home page. All queries are sanitize by Meekro DB.

2. PHP

The code uses a TWIG template to separate the logic from the view. Also all the functions are well documented and placed in a separate file.

The database is queried only once per game to fetch all the necessary questions and answers for that game. A session variable is used to carry these data as well as the status of each question (answered/un-answered) between different pages. Other session variables are used to keep track of the time left and the status of the game (Not Started / Active / Ended).

The quiz UI is very flexible, users can click on the Next or Prev buttons or used the provided thumbnails to navigate between pages. Furthermore, the thumbnails provide visual cues about the status of each question as well as the currently active question.

Security and Error Handling: The user's answers are not saved to the DB, nor displayed on any HTML page, once the user answers a question correctly, that question will be marked as answered, and the correct answer which was fetched from the Database earlier, will be displayed on the page. GET and POST requests are validated once submitted to the server, and default pages will load if required values are not provided or incorrect.

3. HTML, CSS And Bootstrap

The design uses valid HTML-5 and CSS-3 code and is fully responsive. The title, description and keywords meta tags are customized for each page and various other SEO best practices are followed.

4. JavaScript

Since this is a PHP project, JavaScript is used only to add extra features to the quiz pages. The quizzes rely only on the back-end PHP code for functioning and the game is perfectly playable without the JavaScript code. Here is what JS brings to the table:

- Once the page loads, the input field will gain focus.
- JS makes the timer show the real time left, and when the time is up, it will simulate a click on the Next button to force a page refresh, so that PHP can take over and stop the game. Without this feature, the game only ends when the user clicks Next after the timeout. Furthermore, PHP is in charge of keeping the time, and JS only takes the time from PHP and makes it's display live on the page.

- If the JavaScript Assist variable is set to "true" in the PHP code, at each page load, PHP passes the correct answer to JS, and JS will change the text-color of the user input to green if the answer is correct. This helps users to avoid typos before submitting their answer. Since the answer is passed in the HTML source code, it is ciphered by PHP and later deciphered by JS, so that the users cannot find the answer by looking at the source code.

5. API

The same project has also been implemented as an API and is used by a **REACT client application < https://www.aminamani.net/projects/react-quiz-api.html>** . The API version uses the Fat Free Framework (F3), and implements a third party OAuth authentication service (Auth0) which allows users to sign-up and login. Moreover, it defines different roles for users, such that all logged in users can participate in quizzes, but only those users with an "admin" role in their authentication token can view the admin settings. Instructions for using the API version can be found on the **API home page < https://www.aminamani.net/assets/demo/php-quiz-api/>** .

Future Work

The Admin panel can be extended to allow for adding new quiz topics and quiz questions to the website. Plus full CRUD functionality on the database values.

Users could be able to register, to save their scores, and the high scores can be displayed for each quiz.

When taking a quiz, if the user presses the Next button, instead of displaying the next question, which might already have been answered, the UI can jump over to the next un-answered question.

A "Hint" button can be added to provide a hint message to the user for each question (these hints would be saved in th database as well). Another option would be having a "Hint" button that types the first letter of the answer for the user.

Related information about each question can be displayed on the page during the game, or after the game ends. For example general information about the movie plots or a short biography of famous people.

Check out the code on GitHub < https://github.com/AminAmani83/PHP-QuizWebSite>