Klausur „Entwicklung webbasierter Anwendungen“
Sommersemester 2024

Hinweise

- Bearbeitungszeit: 5+90+5 Minuten (je 5 min für Download und Upload).
- Insgesamt gibt es 100 Punkte in 6 Aufgaben.
- **Abgabe bis 26.07.24 um 17:40 über GIT in Ihrem persönlichen Repository für die Prüfung**
- Erlaubte **Hilfsmittel**: Ein Notebook mit Internetzugang pro Person, sowie alle auf dem Notebook mitgebrachten Informationen und jede auf Papier gedruckte Quelle.
- Als **Täuschungsversuch** gelten:
    - Jegliche Formen der "Teamarbeit" z.B. mit Chats, GitHub oder Netzlaufwerken
    - abonnierte oder gekaufte Tools wie ChatGPT4 oder AI-Assistant für PHPStorm
- **Lesen** Sie zuerst die gesamte Aufgabenstellung bevor Sie mit der Lösung der Aufgaben beginnen.
- **Bearbeiten** Sie die Aufgabe im Ordner **src/Exam** des Repositories, das ihnen für die Prüfung zugewiesen wurde. Legen Sie keine neuen Dateien an und benennen Sie die Dateien nicht um. Commiten bzw. pushen Sie ihren Code via Git regelmäßig (ca. alle 10 Minuten) in Ihr zugeteiltes Repository.
- Die **Abgabe** erfolgt ausschließlich über Git. Es wird nur bewertet, was sich bis zum Abgabetermin in dem für Sie angelegten Gitlab-Repository in den vorbereiteten Dateien unter src/Exam befindet.
- Schreiben Sie wartbaren und standardkonformen Code, der die **Regeln** der Veranstaltung für ordentlichen Code und die Vorgaben durch die Seitenklassen berücksichtigt. Verwenden Sie Exceptions zur Behandlung von ungewöhnlichen Fehlern. Entwickeln Sie eine barrierefreie, sichere Anwendung. **Achtung: Die Abgabe von größeren, unnötigen Code-Teilen - wie sie KIs gerne erzeugen - wird als nicht-wartbar (= falsch) gewertet, selbst wenn sie korrekte Teile enthalten.**
- Diese Aufgabenblätter werden nicht abgegeben. Schreiben Sie also keine Lösungsteile darauf. Sie dürfen die Seiten natürlich auch trennen.
- In den PHP-Seitenklassen ist die strenge Typüberprüfung aktiviert. Diese Überprüfung darf nicht abgeschaltet oder umgangen werden. Ändern Sie nicht die Signaturen der vordefinierten Methoden.
- Sie können sowohl in Deutsch als auch in Englisch antworten (You may also answer in English).

Zulieferung

- In Ihrem GitLab-Projekt sind die Dateien, die Sie bearbeiten sollen, schon angelegt.
- Als Zulieferung erhalten Sie eine ZIP-Datei mit vorbereiteten Inhalten. Kopieren Sie zunächst diese Dateien in das Verzeichnis src/Exam und überschreiben Sie die vorhandenen Dateien.
- Wenn Sie die Datenbank mit dem SQL-Skript importieren wollen, melden Sie sich über phpMyAdmin mit dem **Benutzer: root** und **Passwort: ganzGeheim** an. Anschließend nutzen Sie die „Importieren“ Funktionalität, wie Sie es kennen.
- **Page.php**: Die bekannte Basisklasse der Seitenklassen (zugeliefert, nicht bearbeiten)
- **Exam.php** – siehe Aufgabe 1
- **Exam.html** – siehe Aufgabe 1 (zugeliefert, nicht bearbeiten)
- **ExamApi.php** – siehe Aufgabe 2
- **Exam.js** – siehe Aufgabe 3
- **Exam.txt** – siehe Aufgabe 4
- **Exam.css** – siehe Aufgabe 5

Die Aufgabe allgemein: „h_da - Quiz“

In dieser Klausur sollen Sie eine Quiz-App mit einem Punkte-Ranking programmieren. Diese beinhaltet

1. einen Kopf-Bereich mit Navigationselementen.
2. einen Abschnitt "Statistik" für den Punktestand und die aktuelle Platzierung im Vergleich zu anderen Spielern. Da parallel andere Personen spielen können, kann sich während des Spiels der eigene Platz in der Rangliste ändern und soll deshalb ständig aktualisiert werden.
3. einen "Quiz-Bereich" für eine vorgegebene Frage und 2 Antwortmöglichkeiten. Mit einem Knopf "Frage abschicken" kann man seine Antwort absenden, und der Punktestand wird aktualisiert. Dann wird die nächste Frage angezeigt.
4. einen Knopf "Quiz fertig" mit dem man das Spiel beenden kann. Erst dann wird der Punktestand in der Ranking-Tabelle abgespeichert und man landet auf einer (zugelieferten) "Quiz fertig"-Seite.

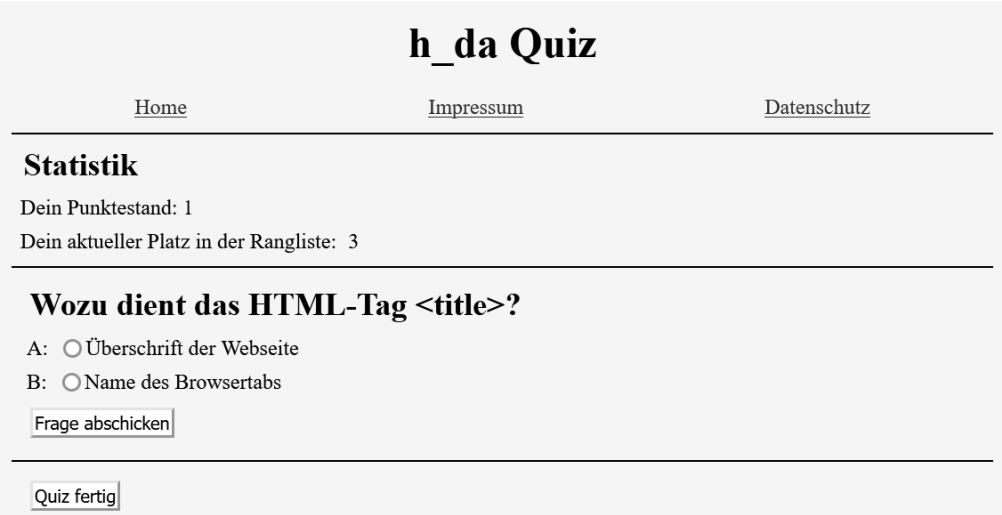

Abbildung 1: h_da-Quiz-Webanwendung mit beispielhafter Ausgabe

Datenbankstruktur

Die Datenbank heißt "*2024_hda_quiz*" und besteht aus den Tabellen *question* und *ranking*. Es existieren schon vordefinierte Fragen und Antworten, sowie Ranglisteneinträge. Weitere Informationen können Sie bei Bedarf der zugelieferten SQL-Datei entnehmen.

id	text	answerOne	answerTwo	answerCorrect
1	Welche Programmiersprache wird hauptsächlich für d...	CSS	PHP	1
2	Wozu dient das HTML-Tag <title>?	Überschrift der Webseite	Name des Browsertabs	2
3	Welches HTTP-Verfahren wird verwendet, um Daten vo...	PUT	GET	2

#	Name	Typ	Kollation
1	id	int(11)	
2	text	varchar(255)	utf8mb4_general_ci
3	answerOne	varchar(255)	utf8mb4_general_ci
4	answerTwo	varchar(255)	utf8mb4_general_ci
5	answerCorrect	int(1)	

#	Name	Typ
1	id	int(11)
2	name	varchar(255)
3	score	int(11)

id	name	score
1	Felix	1
2	Meike	2
3	Felix	3

Aufgabe 1: HTML/PHP (Exam.php) (40 Punkte)

Implementieren Sie in der Datei **Exam.php** unter Verwendungen der **Seitenklassenstruktur** sowie **PHP** und **semantischem HTML,** die in **Abbildung 1** dargestellte Webseite. Folgendes ist zu beachten:

- In dieser Aufgabe soll noch **kein JavaScript** verwendet werden.
- Der Punktestand soll **serverseitig mit Sessions** realisiert werden.

Realisieren Sie folgende Funktionalität:

- Header mit Navigationselementen.
  - Die Links verweisen zur Vereinfachung alle auf die Seite **Exam.php**.
- Statistik Abschnitt.
  - Der Punktestand zeigt an, wie viele Fragen bereits richtig beantwortet wurden.
  - Das Ranking wird später mit AJAX aktualisiert. Setzen Sie den Wert vorerst auf "-".
- Abschnitt mit der Frage und zugehörigen Antwortmöglichkeiten
  - Beim Laden der Seite wird eine zufällige Frage mit Antwortmöglichkeiten aus der Datenbank ausgegeben. Zur Vereinfachung darf die gleiche Frage im Quiz mehrfach auftauchen. Verwenden Sie zum Bestimmen der Frage folgenden SQL-Befehl:
    SELECT id, text, answerOne, answerTwo FROM question ORDER BY RAND() LIMIT 1
  - Bauen Sie die ID der Frage unsichtbar in die Webseite ein, so dass sie zusammen mit der Antwort an den Server übertragen wird.
  - Wird eine Antwort abgeschickt, soll geprüft werden ob diese richtig ist. Falls ja wird der Punktestand um 1 erhöht (Es gibt keine sonstige Rückmeldung). Falls nein, bleibt die Punktzahl unverändert. Danach wird eine zufällige Frage mit dem Punktestand angezeigt.
  - Durch "F5 bzw. Reload" kann man jederzeit eine Frage überspringen (indem man eine neue Frage lädt), ohne eine Antwort abschicken zu müssen.
- „Quiz fertig“ Knopf
  - Durch den Klick auf den Knopf wird das Spiel beendet und der Punktestand in der Datenbanktabelle: Ranking gespeichert. Zur Vereinfachung dürfen Sie immer "Spieler" als Name des Spielers verwenden.
    Hinweis: Die *ranking*-Tabelle kann mehrere Spielstände von einem Spieler abspeichern!
  - Anschließend wird man auf die (zugelieferte) Seite **Exam.html** weitergeleitet.
  - Der Knopf soll erst auftauchen, wenn der Punktestand größer als 0 ist
  - Wenn ein neues Spiel begonnen wird, soll der Punktestand wieder mit 0 beginnen.

Aufgabe 2: PHP (ExamApi.php) (13 Punkte)

Implementieren Sie in der Datei **ExamApi.php** unter Verwendung unserer **Seitenklassenstruktur** ein **PHP-Skript** welches als **Response** einen **JSON-String** zurückliefert. Folgendes ist zu beachten:

- Hierbei handelt es sich um ein API, welches für den aktuellen Punktestand eines Nutzers eine Platzierung berechnet und als JSON-String zurückliefert.
- Der aktuelle Punktestand des Nutzers wurde in Aufgabe 1 seitenübergreifend zwischengespeichert, also steht er auch hier zur Verfügung und muss nicht übergeben werden.
- Um die Platzierung zu erhalten, können Sie in der Datenbanktabelle: Ranking die Einträge zählen, die mehr Punkte haben. Der Rang ist dann um eins größer als diese Zahl.

Hinweis: Passen Sie den zugelieferten Code nach Bedarf an.

Aufgabe 3: JavaScript (Exam.js) (13 Punkte)

Implementieren Sie in der **Exam.js** Datei via JavaScript die Aktualisierung der Platzierung in der Rangliste. Nutzen Sie hierfür die Daten, welche **ExamApi.php** zur Verfügung stellt.

- Registrieren Sie den Aufruf der benötigten Methoden in der JavaScript Datei. Verwenden Sie dafür keine HTML-Tags mit Attributen wie *onload="js_function()"*.
- Sorgen Sie dafür, dass der Rang sofort aktualisiert wird und dann jede Sekunde, ohne dass die Seite neugeladen werden muss.
- Passen Sie den zugelieferten Code nach Bedarf an.

Aufgabe 4: Fragen (Exam.txt) (14 Punkte)

Beantworten Sie in der Datei **Exam.txt,** präzise und ausführlich die folgenden Fragen. Beschreiben Sie ihr Vorgehen und was Sie in welcher Datei anpassen müssten.
Sie sollen diese Aufgaben NICHT IMPLEMENTIEREN, aber ausführlich beschreiben, was Sie tun würden. Pro Teilaufgabe werden 5-10 Zeilen Text mit ca. 80 Zeichen erwartet!

a) Angenommen, die Fragen und Antwortmöglichkeiten sollen sich aktualisieren, ohne dass die Seite jedes Mal neu geladen wird. Wie würden Sie das realisieren? Welche Dateien müssten Sie ändern?
b) Angenommen, Sie wollten den Punktestand (Das Ranking dürfen Sie ignorieren!) clientseitig und ohne Sessions und ohne Cookies verwalten. Wie könnten Sie die Funktionalität trotzdem implementieren? Warum wäre das eine schlechte Idee?

Aufgabe 5: CSS (Exam.css) (10 Punkte)

Implementieren Sie in der **Exam.css** Datei via **CSS** das (grobe) Styling der Webseite. Die **Abbildung 1** sollte Ihnen als Vorlage dienen. Sie müssen nicht alle Abstände pixelgenau realisieren.

Folgendes soll umgesetzt werden:

- Die Links aus der Navigation sollen auf dem Handy untereinander sein, ansonsten nebeneinander. Nutzen Sie zur Anordnung Flexbox und achten Sie auf die Abstände zwischen den Links.
- Setzen Sie die CSS-Attribute für: Hintergrund, Ausrichtung und Linien.
- Achten Sie darauf, dass die verwendeten Farben hinreichend Kontrast haben

Aufgabe 6: Lauffähigkeit/Vollständigkeit (10 Punkte)

Gehen Sie diese Aufgabe erst an, wenn Sie die übrigen Aufgaben nach bestem Wissen gelöst haben.

Prüfen Sie die Lauffähigkeit ihrer Anwendung indem Sie das Docker-Setup starten und im **localhost** ihre Anwendung begutachten. Folgendes soll geprüft werden:

- Prüfen Sie ob Ihr erzeugter HTML-Code valide und barrierefrei ist und den EWA-Regeln für ordentlichen Code entspricht.
- Sieht das Styling ihre Lösung ungefähr so aus wie in Abbildung 1?
- Rufen Sie die **ExamApi.php** im Browser auf und prüfen Sie das Ergebnis
- Vergewissern Sie sich ob die Anwendung funktionsfähig ist, indem Sie das Quiz testen.

Abgabe

Laden Sie Ihr Endergebnis in **Git** hoch und prüfen Sie über **GitLab (https://code.fbi.h-da.de)**, ob die Dateien auch wirklich angekommen sind. Bei Problemen bewahren Sie bitte Ruhe und informieren Ihren Betreuer.