

Leistungsnachweis "Entwicklung Webbasierter Anwendungen" Sommersemester 2021

Aufgabe	1	2	3	4	5	Gesamt
Punkte	29	8	11	6	4	58
Erreichte Punkte						

Vorbereitung:

- Sie haben die Eigenständigkeitserklärung ausgefüllt, unterschrieben und in Git hochgeladen!?
- Ohne ausgefüllte Eigenständigkeitserklärung wird Ihre Abgabe nicht gewertet und kann als Fehlversuch gewertet werden!

Zulieferung:


In Ihrem GitLab-Projekt sind die Dateien, die Sie bearbeiten sollen, schon angelegt und teilweise vorgegeben:

- Page.php: Die bekannte Basisklasse der Seitenklassen mit der Einbindung von CSS und JS und mit der Datenbankbindung. Diese Datei müssen Sie nicht bearbeiten, sondern nur einbinden.
- Exam21.php - siehe Aufgabe 1
- Exam21API.php - siehe Aufgabe 2
- Exam21.js - siehe Aufgabe 3
- Exam21.css - siehe Aufgabe 4

Hinweise:

- Bearbeitungszeit: 90+15 Minuten, **Abgabe bis 31.07.21, 10:50 Uhr über Git in Ihrem persönlichen Repository.**
- Tipp: Stellen Sie sich einen Wecker auf 5 Minuten vor der Abgabezeit!
- Erlaubte Hilfsmittel: Jede elektronische oder auf Papier gedruckte Quelle. Kommunikation mit anderen Menschen außer der Prüfungsaufsicht ist nicht erlaubt!
- Lesen Sie zuerst die gesamte Aufgabenstellung bevor Sie mit der Lösung der Aufgaben beginnen.
- Bearbeiten Sie die Aufgabe im Ordner **src/Exam_21SS** des Repositories, das ihnen zugewiesen wurde. **Committen bzw. pushen Sie ihren Code via *git* regelmäßig (ca. alle 10 Minuten) in Ihr zugeteiltes Repository.** Die Abgabe erfolgt ausschließlich über Git. Commits mit einem Zeitstempel nach Ablauf der Bearbeitungszeit können nicht berücksichtigt werden. Bewertet wird das, was sich dort zum Abgabetermin befindet.
- Schreiben Sie 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. Hinweis: In den PHP-Seitenklassen ist die strenge Typüberprüfung aktiviert. Diese Überprüfung darf nicht abgeschaltet oder umgangen werden.
- Sie können sowohl in Deutsch als auch in Englisch antworten (You may also answer in English).

Wichtig! Schreiben Sie zuerst den Code und versuchen Sie erst am Schluss den Code zum Laufen zu bringen!



Spielplanung

18.08.2021 12:00 Uhr gegen TG 75 Darmstadt

Zusagen Spieler:innen 2

[Planung abschließen](#)

Spiele

Datum	Team	Status
15.06.2021 13:00	Darmstadt 98	vorbei
18.08.2021 12:00	TG 75 Darmstadt	in Planung
25.08.2021 10:00	TSG Darmstadt	zukünftig

Infos zum aktuellen Spiel

Diesen Wert sollen Sie mit AJAX aktualisieren!

Knopf zum Abschließen der Planung

Liste der Spiele:
Das sind tabellarische Daten und Sie dürfen hier eine Tabelle verwenden!

Im Mannschaftssport muss die Trainer:in sicherstellen, dass genügend Spieler:innen zum nächsten Spiel erscheinen. Dazu soll es eine Webseite geben, auf der die Trainer:in sieht, ob für das kommende Spiel genügend Spieler:innen angemeldet sind¹.

Es wird für das nächste anstehende Spiel angezeigt:

- Datum mit Uhrzeit und gegnerische Mannschaft
- Die Summe der Spieler:innen, die für dieses Spiel zugesagt haben (aktualisiert via **AJAX**).

Des Weiteren werden alle vergangenen und zukünftigen Spiele als **Tabelle** angezeigt.

Die zu verwendende Datenbank heißt *2021_TeamSport*. Sie ist in ihrem Repository noch nicht angelegt. Dies ist aber erst ab Aufgabe 5 relevant. Die aufgeführten Beispieldaten passen zum Screenshot. In eckigen Klammern stehen Kommentare, wie die jeweilige Spalte/Tabelle verwendet wird.

id - int (auto)	datetime – datetime [Termin des Spiels]	opposingTeam – varchar(50) [Gegnerische Mannschaft]	status – int [0 zukünftiges Spiel, 1 aktuelles Spiel in Planung, 2 aktuelles Spiel mit abgeschlossener Planung, 3 Spiel vorbei]
1	2021-08-18 12:00:00	TG 75 Darmstadt	1
2	2021-08-25 10:00:00	TSG Darmstadt	0
3	2021-06-15 13:00:00	Darmstadt 98	3

id - int (auto)	gameld – int (Fremdschlüssel)	name – varchar(30) [Name Spieler:in]
1	1	Hanna
2	1	Eva

¹ In dieser Prüfungsleistung beschränken wir uns auf die Ansicht der Trainer:innen – die Seiten zur Spielverwaltung und der Spieler:innen sind nicht Teil der Aufgabenstellung. Die Aufgabenstellung ist insgesamt stark vereinfacht.

Aufgabe 1: PHP/HTML (29 Punkte)

Vervollständigen Sie die Datei Exam21.php.

Aufgabe 1a: Erzeugen Sie das Grundgerüst der Trainer:innen-Seite inklusive Formular in standardkonformen und barrierefreiem *nice* HTML. Schreiben Sie den HTML-Code in den vorgegebenen HEREDOC-Block der Methode *generateView()* der Klasse *Exam21.php*. Beachten Sie folgende Hinweise:

- Die Summe "Zusagen Spieler:innen" wird später mit AJAX eingefügt, fügen Sie hier zunächst nur ein „?“ ein.
- Das Layout und Design der Seite entsteht später mit CSS. Machen Sie keine Formatierungen mit HTML!
- Das Logo liegt in der Datei "Logo.png" vor.
- Die Formulardaten schickt die Klasse an sich selbst ab (Exam21.php) (vgl. Aufgabe 1b).

Aufgabe 1b: Lesen Sie die Spiel-Einträge der Tabelle *games* aus der Datenbank aus.

Aufgabe 1c: Passen Sie den Code aus Aufgabe 1 so an, dass die Daten der Datenbank verwendet werden. Gibt es kein aktuelles Spiel (kein Spiel mit *status* eins oder zwei), so wird statt der Infos zum Spiel der Text „kein aktuelles Spiel“ ausgegeben.

Aufgabe 1d: Wenn ein:e Trainer:in auf „Planung abschließen“ klickt, wird der Status des oben angezeigten Spiels auf 2 gesetzt.

Hinweis: Laden Sie Ihren Zwischenstand mit Git hoch!

Aufgabe 2: PHP/AJAX (8 Punkte)

Vervollständigen Sie die vorbereitete Klasse **Exam21API**: Diese erwartet einen Get-Parameter *gameId* und liefert für das Spiel *gameId* die Anzahl der zugesagten Spieler:innen als JSON zurück. Nutzen Sie folgende SQL-Abfrage:
`$sql = "SELECT count(*) AS playing FROM gameDetails WHERE gameId = $this->gameId";`

Aufgabe 3: ECMAScript / AJAX (11 Punkte)

Die Trainer:innen-Seite soll mit AJAX beim Seitenaufruf und dann alle 10 Sekunden zyklisch die Summe der zugesagten Spieler:innen aktualisieren. Vervollständigen Sie hierzu die vorbereitete Datei Exam21.js.

Aufgabe 3 a: Schreiben Sie eine JavaScript-Funktion **pollData()**, welche die Daten über AJAX abfragt, indem sie die (gegebene **unvollständige!**) Funktion *requestData()* aufruft.

Aufgabe 3b: Rufen Sie die Funktion pollData an geeigneter Stelle auf.

Aufgabe 3c: Aktualisieren Sie die Summe der Spieler:innen mit den erhaltenen Daten.

Aufgabe 4: CSS (6 Punkte)

Fügen Sie in die Datei **Exam21.css** Code ein, der die folgenden Aspekte des Designs der in der Aufgabenstellung dargestellten Seite umsetzt. Designteile nach denen nicht gefragt wird, müssen Sie nicht umsetzen!

- a) Die Tabellenzellen haben einen von der Schriftgröße abhängigen Innenabstand.
- b) Die Summe der zugesagten Spieler:innen hat die Hintergrundfarbe 40785e (RGB) und die Textfarbe ist weiß.
- c) Das Logo wird links vom Titel angeordnet und in seiner Höhe geeignet beschränkt.

Aufgabe 5: Lauffähigkeit / Vollständigkeit (4 Punkte)

Damit der Code laufen kann, müssen Sie die zugeliesserte Datenbank inkl. Tabellen und Beispieldaten importieren: Melden Sie sich hierzu als root in der phpMyAdmin-Oberfläche an und führen Sie im Reiter SQL das mit der Aufgabenstellung zur Verfügung gestellte SQL-Skript aus. Starten Sie nun Ihre Webanwendung und gehen Sie folgende Checkliste durch:

- Vergleichen Sie die angezeigten Informationen mit der Abbildung der Aufgabenstellung.
- Fügen Sie mit phpMyAdmin in der Tabelle gameDetails eine weitere Spielerin ein. Prüfen Sie, ob sich die Anzahl der angezeigten Spieler:innen auf 3 erhöht.
- Klicken Sie auf die Schaltfläche „Planung abschließen“: Prüfen Sie, ob sich der Eintrag in der Tabelle unten auf der Seite nach einem Reload in "Planung abgeschlossen" ändert.

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.

Hinweise für den Notfall

Falls Sie Probleme mit der Netzwerkverbindung haben...

- Ruhe bewahren und erst einmal weiterarbeiten. Oft lösen sich die Probleme nach ein paar Minuten.
- Das Handy für den Netzwerkzugang nutzen (USB-Tethering)
- Bescheid geben / Kontakt aufnehmen über den 2. Kanal (Zoom evtl. mit dem Handy). Die Zugangsdaten erfahren Sie vor der Prüfung per Email.
- Nur wenn die Probleme bis zur Abgabe bestehen, sollten Sie das Endergebnis als ZIP-Datei per Email an Prof. Dr. Ralf Hahn oder Prof. Dr. Ute Trapp senden. In diesem Fall gilt der Zeitstempel des Emailproviders als Abgabezeit.