

Meilenstein 5 – E-Mensa (mit Anmeldung)

Ziel ist die Weiterentwicklung der E-Mensa. Wir ermöglichen eine Anmeldung durch Benutzer:innen, erweitern die Darstellung von Gerichten um Bilder und berechnen zentrale Statistiken zur Verwendung der E-Mensa mit Views in SQL.

Aufgabe 1

E-Mensa: Anmeldung. Die E-Mensa soll registrierten Benutzer:innen die Möglichkeit für eine Anmeldung bieten. Erst nach einer Anmeldung können Benutzer:innen zukünftig Gerichte vorab bestellen und Bewertungen zu Gerichten abgeben.

- 1) Legen Sie die folgende Struktur in der Datenbank an.

benutzer

Attribut	Typ	Kommentar
id	int8, PKEY	Eindeutige ID, Auto-Inkrement.
name	varchar(200), not null	Name, der auch auf der Oberfläche dargestellt wird.
email	varchar(100), not null, unique	Eindeutige E-Mail der Benutzer:in. Teil der Anmeldung.
passwort	varchar(200), not null	Speicherung des Passwort-Hashs mit SHA-1.
admin	boolean, not null	Markierung, ob es sich um einen Administrator handelt oder nicht. Standard: false.
anzahlfehler	int, not null	Zähler, wie oft hintereinander eine Anmeldung erfolglos durchgeführt wurde. Standard: 0.
anzahlanmeldungen	int, not null	Zähler, wie oft eine Anmeldung insgesamt erfolgreich durchgeführt wurde.
letzteammeldung	datetime	Zeitpunkt, an dem sich der/die Benutzer:in zuletzt erfolgreich angemeldet hat.
letzterfehler	datetime	Zeitpunkt, an dem sich der/die Benutzer:in zuletzt erfolglos angemeldet hat.

- 2) Stellen Sie sicher, dass zentral die Verwendung von Sessions für die gesamte Anwendung eingeschaltet ist.
- 3) Legen Sie eine/n Benutzer:in für die Administration an. Verwenden Sie die E-Mail „admin@emensa.example“. Wählen Sie selbst ein Passwort und berechnen Sie den zu speichernden Hash selbst in einem eigenen PHP Skript, das Sie unter `beispiele/passwort.php` ablegen. Verwenden Sie einen Salt mit mindestens 4 Zeichen Ihrer Wahl für die gesamte Anwendung (nicht pro Benutzer:in).
- 4) Zeigen Sie unter dem Endpunkt `/anmeldung` eine Anmeldemaske mit den Eingabefeldern E-Mail und Passwort an. Wählen Sie passende

Eingabefelder. Nach dem Betätigen der Schaltfläche „Anmeldung“ wird eine Überprüfung in `/anmeldung_verifizieren` durchgeführt. Bei einer erfolgreichen Anmeldung gelangen Benutzer:innen zurück auf die Hauptseite. Bei einer misslungenen Anmeldung sehen Benutzer:innen die Anmeldemaske unter `/anmeldung` und erhalten eine entsprechende Fehlermeldung.

- 5) Bei einer erfolgreichen Anmeldung soll der Zähler *anzahlanmeldungen* in der Datenbank um eins inkrementiert werden.
- 6) Bei einer erfolgreichen Anmeldung soll das Attribut *letzteanmeldung* auf den aktuellen Zeitpunkt gesetzt werden. Bei einer fehlerhaften Anmeldung soll der aktuelle Zeitpunkt im Attribut *letzterfehler* gesetzt werden, wenn ein/e Benutzer:in mit dem verwendeten Namen existiert.
- 7) Zeigen Sie einem angemeldeten Benutzer auf der Hauptseite, dass dieser angemeldet ist. Dies können Sie z.B. oben rechts mit einem Text „Angemeldet als <Name>“ umsetzen.
- 8) (Optional) Bei einem Klick auf den Namen bei „Angemeldet als <Name>“ gelangen Benutzer:innen auf `/profil`, was die Profildaten (E-Mail, Anzahl Anmeldungen, Adminkonto oder nicht) anzeigt. Ist der/die Benutzer:in nicht angemeldet und navigiert auf `/profil`, so wird diese/r auf die Anmeldeseite weitergeleitet.
- 9) Führen Sie alle SQL-Anweisungen der Anmeldung in einer Transaktion durch, wobei Sie die Funktionen `mysqli_begin_transaction()`⁶ und `mysqli_commit()` verwenden.
- 10) Implementieren Sie unter `/abmeldung` eine sofortige Abmeldung von Benutzer:innen von der Webseite. Benutzer:innen sollen nach der Abmeldung die Hauptseite sehen. Fügen Sie für angemeldete Benutzer:innen einen Link auf die Hauptseite zur Abmeldung ein.
- 11) (Optional) Sperren Sie Benutzer:innen, wenn diese 3 mal direkt hintereinander das Passwort falsch eingegeben hat. Sperren Sie Benutzer:innen nicht, falls diese Administrator:innen sind.
- 12) (Optional) Nach einer fehlerhaften Anmeldung müssen Benutzer:innen 5 Sekunden warten, bevor dieser die nächste Anmeldung durchführen kann. Es erscheint eine Warnung, falls Benutzer:innen zu schnell eine Anmeldung versuchen.
- 13) (Optional) Implementieren Sie die Anmeldung mit den Funktionen `password_hash()`⁷ und `password_verify()`.

⁶ <https://www.php.net/manual/en/mysqli.begin-transaction.php>

⁷ <https://www.php.net/manual/de/function.password-hash.php>

Aufgabe 2

E-Mensa: Bilder von Gerichten. Besucher:innen der E-Mensa sollen die Gerichte über passende Bilder schneller erkennen und besser einschätzen können, ob dieses Gericht ihnen wohl schmeckt. Dazu erhalten Sie die Datei **gerichtebilder.zip**. Lösen Sie die folgenden Aufgaben:

- 1) Erweitern Sie die Struktur der Tabelle *gericht* via ALTER Statement:

Attribut	Typ	Kommentar
bildname	varchar(200)	Name der Bilddatei, die das Gericht darstellt. Standard: null.

- 2) Speichern Sie die Bilder im Verzeichnis `/public/img/gerichte/` ab. Aktualisieren Sie die Tabelle *gericht* indem Sie den passenden Bildnamen unter dem Attribut *bildname* abspeichern. Die Bildnamen sind nach dem folgenden Namensschema aufgebaut:

`<id>_beliebigerName.{jpg, png, ...}`

`<id>` entspricht dem Primärschlüssel in der Tabelle *gericht*.

- 3) Erweitern Sie die Übersicht der Gerichte auf der Hauptseite (unter /) um die Darstellung der Bilder. Zeigen Sie die Bilder passend in der Übersicht an (z.B. durch Anwendung der ``-Attribute `width` und `height`).
- 4) Ist für ein Gericht kein Bild gepflegt (ist also *bildname* gleich NULL oder die Bilddatei existiert nicht bei dem hinterlegten Namen), so soll das Bild `00_image_missing.jpg` dargestellt werden.

Woche 2

Empfehlung VL: PHP Namespaces, Paketverwaltung, SQL Views Stored Procedures

Aufgabe 3

E-Mensa: Logging. Die E-Mensa schreibt aktuell noch keine Lognachrichten. Lognachrichten sind für Anwendungen von hoher Bedeutung, da diese die durchgeführten Operationen einer Anwendung dokumentieren und so unter anderem die Nachvollziehbarkeit verbessern.

- 1) Installieren Sie Monolog über Composer. Sie finden Monolog unter:
<https://packagist.org/packages/monolog/monolog>
- 2) Erweitern Sie die `index.php` (den Frontcontroller), damit Monolog in den Controllerklassen komfortabel verwendet werden kann. Schreiben Sie dazu eine Funktion mit dem Namen `logger()`, die einen einfachen Zugriff auf das Logger-Objekt von Monolog bereitstellt. Geben Sie das konfigurierte Logger-Objekt zurück. Orientieren Sie sich dabei an der

bestehenden Funktion `view()` sowie an dem Anwendungsbeispiel der Dokumentation von Monolog.

- 3) Legen Sie das Verzeichnis **/storage/logs** an und schreiben Sie bei einem Aufruf der Hauptseite eine Nachricht mit Monolog in die Logdatei.
- 4) Dokumentieren Sie ...
 - a) mit einer `info()`-Nachricht die An- und Abmeldungen
 - b) mit einer `warn()`-Nachricht die fehlgeschlagenen Anmeldungen
 - c) mit einer `info()`-Nachricht den Zugriff auf die Hauptseite

Aufgabe 4

E-Mensa: Auswertung mit SQL-Sichten. Die E-Mensa GmbH möchte einen schnellen Überblick über alle verfügbaren Gerichte pro Kategorie aufgeteilt nach vegetarisch und nicht vegetarisch erhalten. Zusätzlich soll es eine einfache Abfrage auf der Datenbank von allen Suppengerichten geben.

- a) Erstellen Sie eine SQL-Sicht `view_suppengerichte`, die alle Suppen-Gerichte (die ein `*suppe*` im Namen tragen) darstellt.
- b) Erzeugen Sie eine SQL-Sicht `view_anmeldungen`, die die Anzahl der Anmeldungen pro Benutzer absteigend sortiert nach Anzahl der Anmeldungen darstellt.
- c) Erzeugen Sie die SQL-Sicht `view_kategoriegerichte_vegetarisch`, die alle vegetarischen Gerichte sowie die zugehörigen Kategorien zeigt. Stellen Sie immer alle Kategorien dar, auch wenn dieser Kategorie kein Gericht zugeordnet ist.
Hinweis: Doppelte Nennungen der Kategorie sind in Ordnung.
- d) (Optional) Zeigen Sie die Übersicht der Ergebnisse der SQL-Sicht `view_kategoriegerichte_vegetarisch` unter `/berichtkategorien` in der Webanwendung.
- e) Geben Sie die verwendeten CREATE VIEW Statements in der Datei **/beispiele/m5_4.sql** ab.

Aufgabe 5

E-Mensa: Anmeldungszähler als Prozedur. Ein Teil der Geschäftslogik der Anmeldung, genauer das Zählen der Anmeldungen pro Benutzer:in, soll in die Datenbank verlegt werden.

- a) Schreiben Sie eine Datenbank-Prozedur, die den Zähler bei einer Anmeldung `anzahlanmeldungen` in der Tabelle **benutzer** inkrementiert. Übergeben Sie die notwendige id des betreffenden Datensatzes aus **benutzer** an die Prozedur. Geben Sie die SQL-Anweisung zur Erzeugung der Prozedur unter **/beispiele/m5_5.sql** ab.

- b) Rufen Sie die Prozedur während der Anmeldung aus der Anwendung heraus auf.
- c) Welche weiteren Prozeduren sind aus Ihrer Sicht bei der E-Mensa denkbar? Nennen Sie zwei mögliche Beispiele und begründen Sie, warum sich eine Prozedur anbietet.

Aufgabe

Abgabe. Laden Sie Ihre Ergebnisse als ZIP in unter

<https://git.fh-aachen.de/praktika/dbwt/ws24>

in den Ordner Ihres Teams in den Branch „main“ hoch. Der Name des ZIPs soll sein:

M5.zip

Das Zip umfasst mindestens die Verzeichnisse und Dateien:

- /beispiele
- /werbeseite
- dossier.{txt, pdf, html, ...}
- README

Das ZIP soll alle Ergebnisse des Meilensteins (Antworten zu Freitextaufgaben, Grafiken, Quelltexte, ...) enthalten. Fügen Sie zusätzlich ein README ein, in der Sie Vor-, Nachname und Matrikelnummer der beteiligten Teammitglieder hinterlegen. Eine Abgabe einer Person für das Team reicht aus. Die letzte Abgabe vor dem Abgabezeitpunkt (Freitag, 22:00 Uhr) zählt.