

DBWT Dossier

Autoren: Muhammad Zulfahmi bin Zaid, Hristomir Dimov

Meilenstein 1

Aufgabe	geschätzter Aufwand (min)	benötigte Zeit (min)	Kommentare
1	10	6	-
2	2	2	-
3	10	4	-
4	20	30	Abweichungsgrund -weiß nicht, wie ich ein Grid-Cell in zwei Reihen/Spalten einsetzen kann.
5	30	45	Abweichungsgrund -hab Probleme mit dem radio-type-buttons
6	15	20	Abweichungsgrund -Eigentlich relativ einfach, weil die Sachen schon in der Vorlesung behandelt wurden; nur bei manchen HTTP-Attribute muss ich nach der Bedeutung suchen.
7	15	20	Abweichungsgrund -bin mir nicht sicher, ob alle Datei komplett ohne Warnungen sein sollen oder man darf die Warnung von noch nicht vorhandener Datei(formdata.html) bei Aufgabe 5 ignorieren.
8	20	45	Abweichungsgrund -nach font-type suchen -CSS verstehen -verschiedene Einstellungen von Attributes rumspielen, damit alles zu dem gegebenen Bild passt.
9	30	60	Abweichungsgrund -nach der möglich nah richtiger Farbe

			suchen -wie man die Länge des Borders verändern kann. -Wie kann man auf eine spezielle Reihe/ Spalte in einer Tabelle mittels CSS zugreifen.
10	90	120	Abweichungsgrund Erste Erfahrungen mit HTML, CSS
Gesamte Zeit Meilenstein 1	5.86 Stunden(352 Min)		

Aufgabe 6

6.1)

Anfrage Header

The screenshot shows the 'Request Headers' section of a web browser's developer tools. The request is to 'bibliothek/'. The headers are as follows:

- Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
- Accept-Encoding:** gzip, deflate, br
- Accept-Language:** de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
- Cache-Control:** max-age=0
- Connection:** keep-alive
- Cookie:** fhac_cookiemodal-selection=[%22essential%22]; _pk_id.4.19d4=f47b19ac8bbcb0a6.1664697332.; _pk_ref.4.19d4=%5B%22%2C%22%22%2C1664887724%22%22https%3A%2F%2Fwww.google.com%2F%22%5D; _pk_ses.4.19d4=1
- Host:** www.fh-aachen.de
- sec-ch-ua:** "Chromium";v="106", "Google Chrome";v="106", "Not;A=Brand";v="99"
- sec-ch-ua-mobile:** ?0
- sec-ch-ua-platform:** "Windows"
- Sec-Fetch-Dest:** document

At the bottom left, it shows '3 / 123 requests' and '14.2 kB / 23.3 kB'.

5 Attribute des Headers

- 1) **Accept:** gibt die Information an, welche Formate der Client in der Antwort unterstützt
- 2) **Accept-Language:** gibt die Information an, welche Sprache der Client bevorzugt. (in diesem Fall de-De mit q = 1)
- 3) **Cache-Control:** überträgt einzelne Direktiven für die Steuerung des Caches. (hier Request-Direktiven)

- 4) **Connection:** gibt die Information an, ob die Verbindung persistent oder nicht persistent ist.
- 5) **Cookie:** enthält gespeicherte Cookies der Server.

Antwort Header

Name	×	Headers	Preview	Response	Initiator	Timing	Cookies
bibliothek/		▼ Response Headers					View source
<input type="checkbox"/> matomo.php?action_name=Bi...		Cache-Control: private, no-store					
<input type="checkbox"/> favicon.ico		Connection: keep-alive					
		Content-Encoding: gzip					
		Content-Language: de					
		Content-Length: 13502					
		Content-Type: text/html; charset=utf-8					
		Date: Tue, 04 Oct 2022 12:49:18 GMT					
		Permissions-Policy: geolocation=(self), microphone=()					
		Referrer-Policy:					
		Server: nginx/1.18.0 (Ubuntu)					
		Strict-Transport-Security: max-age=31536000					
		Vary: Accept-Encoding					
		X-Content-Type-Options: nosniff					
		X-Frame-Options: SAMEORIGIN					
		X-XSS-Protection: 1; mode=block					

5 Attribute des Headers

- 6) **Cache-Control:** überträgt einzelne Direktiven für die Steuerung des Caches. (hier Response -Direktiven)
- 7) **Connection:** gibt die Information an, ob die Verbindung persistent oder nicht persistent ist.
- 8) **Content-Encoding:** Es enthält Informationen darüber, welches Encoding auf der Datei angewendet wird.
- 9) **Content-Language:** Welche Sprache die Datei enthält
- 10) **Content-Length:** der Länge des Message Bodys im Bytes

6.2)

```
▼ Request Headers (820 B)
? Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
? Accept-Encoding: gzip, deflate, br
? Accept-Language: en-US,en;q=0.5
? Connection: keep-alive
? Content-Length: 87
? Content-Type: application/x-www-form-urlencoded
? Cookie: Phpstorm-eb7997c=fdeb2802-d0e5-4ecf-a268-3cbeebfdee11
? Host: localhost:63342
? Origin: http://localhost:63342
? Referer: http://localhost:63342/E-Mensa%20Werbeseite/beispiele/newsletteranmeldung.html?_ijt=rffr18m0ot57jvo72s5inrvbjp&_ij_reload=RELOAD_ON_SAVE
? Sec-Fetch-Dest: document
? Sec-Fetch-Mode: navigate
? Sec-Fetch-Site: same-origin
? Sec-Fetch-User: ?1
? Upgrade-Insecure-Requests: 1
? User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0
```

Accept: Welche Formate mit welcher Priorität (q) werden vom Server gestattet

Accept-Encoding: Kompression der Dateien und die benutzten Algorithmen dazu(gzip, deflate, br)

Accept-Language: Welche Sprachen und mit welcher Priorität vom Server akzeptiert werden

Connection: keep-alive: Die Verbindung zwischen Server und Client wird aufgehoben, bis einer der Teilnehmer sie unterbricht

Content-Length: bezeichnet die Größe der Dateien in Bytes

Cookie: Session-ID der verwendeten Cookie, die mit dem Server angebunden ist

Host: ID des Hosts

Origin: URL des Hostservers

Referer: Absolute Adresse der Webseite, die die Anfrage macht

Sec-Fetch-Dest: Ziel des Headers

Sec-Fetch-Mode: Methode der Anfrage(unterscheidet zwischen einfachem Browsen von HTML Seiten und Laden von Ressourcen). In dem Fall steht "navigate" für Top-Level-Anfragen.

Sec-Fetch-Site: gibt die Beziehung zwischen dem Ursprung einer Anforderung und dem Ursprung der angeforderten Ressource an

Sec-Fetch-User: wird nur für Anfragen gesendet, die vom Benutzer initiiert wurden, und der Wert ist immer ?1

Upgrade-Insecure-Requests: sendet ein Signal an den Server, das die Präferenz des Clients für eine verschlüsselte und authentifizierte Antwort ausdrückt

User-Agent: ermöglicht dem Server den Browser Client, Benutzersystem und andere nützliche Informationen des Clients erkennen

Request Body

```
Form data
geschlecht: "Frau"
vname: "vname"
nname: "nname"
email: "em@g.com"
benachrichtungsintervall: "Tag"
```

Im Body stehen die übertragene Daten, in dem Fall Daten vom Client wie Geschlecht, Vorname, Nachname, E-Mail und Benachrichtungsintervall.

Aufgabe 9)

font-family: Die Schrift von Textelementen wird gesetzt

color: Die Farbe eines Elements setzen

text-decoration: Ändert das Aussehen des Textes durch Zierlinien

font-weight: setzt wie dünn oder dick chars in einem Text dargestellt werden

border: erlaubt einem die Ränder eines Textes zu gestalten

margin-left: Wird verwendet, um außerhalb definierter Grenzen einen Raum um Elemente herum zu erstellen und zu gestalten.

background-color: Die Farbe des Hintergrunds eines Elements ändern

width: setzt wie breit eine Element auf der Webseite ist

line-height: gibt an und setzt wie hoch eine Zeile sein soll

text-align: dadurch wird die horizontale Position eines Elements auf der Webseite

padding-bottom: die Distanz zwischen einem Element und seiner Grenze

display: gibt an wie ein Element(Text, Bilder, etc.) dargestellt wird

border-right: die rechte Grenze von verschiedenen Elementen einstellen

padding-right: Abstand zwischen Element und seiner rechten Grenze schaffen

padding-left: Abstand zwischen Element und seiner linken Grenze schaffen

position: gibt an wie ein Element auf der Webseite positioniert ist

content: Die Content-Eigenschaft wird verwendet, um generierte Inhalte einzufügen(z.B. Links).

```
* {  
    font-family: Arial;  
}
```

* ist ein Universal Selektor. Alle Texte werden auf Ariel gesetzt.

```
[href = "#etabelle"]{  
    font-weight: bold;  
    color: orange;  
    text-decoration: none;  
}
```

Alle Elemente, die folgenden Attribut haben, haben die beschriebenen Eigenschaft.

```
tr:nth-child(2){  
    background-color: moccasin;  
}
```

nth-child ist eine Pseudoklasse, mit der man dem entsprechenden "Child" (je nachdem, was für eine Nummer in Klammern steht) speziell zugreifen kann.

Meilenstein 2

Aufgabe	Geschätzte Zeit	Tatsächliche Zeit	Kommentar
1	5	5	-
2	15	32	<p>Abweichungsgrund: (Hristomir) Quiz war schwieriger als erwartet, aber eine gute Wiederholung.</p> <p>(Zulfahmi) beim ersten Versuch habe ich den Test nicht bestanden</p> <p>beim zweiten schon</p>
3	45	150	<p>Abweichungsgrund: (Hristomir) Bei vielen Teilaufgaben musste man vieles selbst recherchieren. Vom Aufwand her aber passend.</p> <p>(Zulfahmi) Ich verstehe die Aufgabe 4 e) nicht. wie kann ich auf die Get Parameter "show_description" zugreifen?</p> <p>(Aufgabenstellung nicht richtig gelesen)</p> <p>bei der Aufgabe 4 g) habe ich viel Zeit verschwendet, da ich versuche immer die Übersetzungen auf externe Datei zu lesen.</p>
4	2	2	-
5	60	90	<p>Abweichungsgrund: Am meisten Zeit bei d) wegen Schwierigkeiten mit Array Operationen gebraucht</p>
6	30	30	-
7	20	45	<p>Abweichungsgrund: die Funktionsweise von \$_SERVER verstehen</p>

			die Erkennung von nicht erhaltenen Worte überlegen
8	120	90	Abweichungsgrund: Die in den vorherigen Aufgaben investierte Zeit hat sich bei der letzten Aufgabe gelohnt.
Gesamte Zeit Meilenstein 2	7,4 Stunden(444 Minuten)		

Aufgabe 3

1)

Falsch	Richtig
<p>- Komma nach 4. Element vergessen</p> <pre>\$allergens = [11 => 'Gluten', 12 => 'Krebstiere', 13 => 'Eier', 14 => 'Fisch' 17 => 'Milch'];</pre>	<pre>\$allergens = [11 => 'Gluten', 12 => 'Krebstiere', 13 => 'Eier', 14 => 'Fisch', 17 => 'Milch'];</pre>

- schließende Klammer bei Unterarray fehlt

```
$meal = [
    'name' => 'Süßkartoffeltascher
    'description' => 'Die Süßkarto
    'price_intern' => 2.90,
    'price_extern' => 3.90,
    'allergens' => [11, 13,
    'amount' => 42
];
```

```
$meal = [
    'name' => 'Süßkartoffeltas
    'description' => 'Die Süßk
    'price_intern' => 2.90,
    'price_extern' => 3.90,
    'allergens' => [11, 13],
    'amount' => 42
];
```

- fehlendes Stichwort "function"

```
calcMeanStars(array $ratings) : float {
    $sum = 1;
    foreach ($ratings as $rating) {
        $sum += $rating['stars'] / count($ratings);
    }
    return $sum;
}
```

```
function calcMeanStars(array $ratings) : float {
    $sum = 1;
    foreach ($ratings as $rating) {
        $sum += $rating['stars'] / count($ratings);
    }
    return $sum;
}
```

- fehlende schließende Klammer für if-Bedingung

```
$showRatings = [];
if (!empty($_GET[GET_PARAM_SEARCH_TEXT])){
    $searchTerm = $_GET[GET_PARAM_SEARCH_TEXT];
    foreach ($ratings as $rating) {
        if (strpos($rating['text'], $searchTerm) !== false) {
            $showRatings[] = $rating;
        }
    }
}
```

```
$showRatings = [];
if (!empty($_GET[GET_PARAM_SEARCH_TEXT])) {
    $searchTerm = $_GET[GET_PARAM_SEARCH_TEXT];
    foreach ($ratings as $rating) {
        if (strpos($rating['text'], $searchTerm) !== false) {
            $showRatings[] = $rating;
        }
    }
}
```

2)

Als erstes werden Texte von Bewertungen nach einer gesuchten Zeichenkette gefiltert. Diese Funktionalität wird im Browser durch die Textbox und die Submit-Taste angesteuert. Zudem wird die durchschnittliche Wertung eines Gerichts automatisch beim Öffnen der Webseite berechnet und angezeigt.

3)

empty() - Überprüft, ob eine Variable leer ist (das Gegenteil von isset())

strpos(String, subString) - findet die Position des ersten Vorkommens von subString in String (berücksichtigt Groß- und Kleinbuchstaben)

```
function calcMeanStars(array $ratings) : float {
```

: “: float ” als Deklaration des Rückgabetypen. Diese Funktionalität gibt den Werttyp an, den eine Funktion zurückgeben soll.

count(\$arr) - Gibt die Anzahl der Elemente in dem Array \$arr zurück

Meilenstein 3

Aufgabe	Geschätzte Zeit	Tatsächliche Zeit	Kommentar
1	5	5	-
2	10	15	Abweichungsgrund: Syntax von SQL bisher unbekannt.
3	20	30	Abweichungsgrund: mehr Zeit beim Suchen benötigt.
4	5	5	-
5	20	40	Abweichungsgrund: Um alle Allergene darzustellen und zu konkatenieren, braucht man mehr Zeit.
6	30	50	Abweichungsgrund: Mengenoperationen in SQL sind schwer zu übersetzen.
7	5	5	-
9	40	80	Abweichungsgrund: SQL Statements waren für eine erfolgreiche Implementierung unnötig. Man soll die Anzahl der Anmeldungen NACH dem Hochzählen speichern.
10	5	5	-
11	30	90	Abweichungsgrund: Ich musste VL wiederholen (; + extra Zeit für Überarbeitung...
Gesamte Zeit Meilenstein 3	5.42 Stunden(325 Minuten)		

Aufgabe 2

Abfrage

```
SELECT * FROM gericht;  
SELECT * FROM allergen;  
SELECT * FROM kategorie;  
SELECT * FROM gericht_hat_allergen;  
SELECT * FROM gericht_hat_kategorie;
```

Aufgabe 3

Abfrage

```
SELECT * FROM gericht;
```

Ergebnis

id	name	beschreibung	erfasst_am	vegetarisch	vegan	preis_intern	preis_extern
1	1 Bratkartoffeln mit Speck un...	Kartoffeln mit Zwiebeln und...	2020-08-25	0	0	2.3	4
2	2 Bratkartoffeln mit Zwiebeln	Kartoffeln mit Zwiebeln und...	2020-08-25	1	1	2.3	4
3	3 Grilltofu	Fein gewürzt und mariniert	2020-08-25	1	1	2.5	4.5
4	4 Lasagne	Klassisch mit Bolognesesoße...	2020-08-24	0	0	2.5	4.5
5	5 Lasagne vegetarisch	Klassisch mit Sojagranulats...	2020-08-24	1	0	2.5	4.5
6	6 Hackbraten	Nicht nur für Hacker	2020-08-25	0	0	2.5	4
7	7 Gemüsepfanne	Gesundes aus der Region, de...	2020-08-25	1	1	2.3	4
8	8 Hühnersuppe	Suppenhuhn trifft Petersilie	2020-08-25	0	0	2	3.5
9	9 Forellenfilet	mit Kartoffeln und Dill dip	2020-08-22	0	0	3.8	5
10	10 Kartoffel-Lauch-Suppe	der klassische Bauchwärmer ...	2020-08-22	1	0	2	3
11	11 Kassler mit Rosmarinkartoff...	dazu Salat und Senf	2020-08-23	0	0	3.8	5.2
12	12 Drei Reibekuchen mit Apfel...	grob geriebene Kartoffeln a...	2020-08-23	1	0	2.5	4.5
13	13 Pilzpfanne	die legendäre Pfanne aus Pi...	2020-08-23	1	0	3	5
14	14 Pilzpfanne vegan	die legendäre Pfanne aus Pi...	2020-08-24	1	1	3	5
15	15 Käsebrötchen	schmeckt vor und nach dem E...	2020-08-24	1	0	1	1.5
16	16 Schinkenbrötchen	schmeckt auch ohne Hunger	2020-08-25	0	0	1.25	1.75
17	17 Tomatenbrötchen	mit Schnittlauch und Zwiebe...	2020-08-25	1	1	1	1.5
18	18 Mousse au Chocolat	sahnige schweizer Schokolad...	2020-08-26	1	0	1.25	1.75
19	19 Suppenkreation à la Chef	was verschafft werden muss, ...	2020-08-26	0	0	0.5	0.9
20	20 Currywurst mit Pommes	KA	2020-06-26	0	0	1	400

Abfrage

```
SELECT erfasst_am FROM gericht;
```

Ergebnis

	erfasst_am
1	2020-08-25
2	2020-08-25
3	2020-08-25
4	2020-08-24
5	2020-08-24
6	2020-08-25
7	2020-08-25
8	2020-08-25
9	2020-08-22
10	2020-08-22
11	2020-08-23
12	2020-08-23
13	2020-08-23
14	2020-08-24
15	2020-08-24
16	2020-08-25
17	2020-08-25
18	2020-08-26
19	2020-08-26
20	2020-06-26

Abfrage

```
SELECT erfasst_am, name FROM gericht
ORDER BY name ASC;
```

Ergebnis

	erfasst_am	name
1	2020-08-25	Bratkartoffeln mit Speck und Zwiebeln
2	2020-08-25	Bratkartoffeln mit Zwiebeln
3	2020-06-26	Currywurst mit Pommes
4	2020-08-23	Drei Reibekuchen mit Apfelmus
5	2020-08-22	Forellenfilet
6	2020-08-25	Gemüsepfanne
7	2020-08-25	Grilltofu
8	2020-08-25	Hackbraten
9	2020-08-25	Hühnersuppe
10	2020-08-22	Kartoffel-Lauch-Suppe
11	2020-08-24	Käsebrötchen
12	2020-08-23	Kassler mit Rosmarinkartoffeln
13	2020-08-24	Lasagne
14	2020-08-24	Lasagne vegetarisch
15	2020-08-26	Mousse au Chocolat
16	2020-08-23	Pilzpfanne
17	2020-08-24	Pilzpfanne vegan
18	2020-08-25	Schinkenbrötchen
19	2020-08-26	Suppenkreation á la Chef
20	2020-08-25	Tomatenbrötchen

Abfrage

```
SELECT beschreibung, name FROM gericht
ORDER BY name ASC LIMIT 5;
```

Ergebnis

	beschreibung	name
1	Kartoffeln mit Zwiebeln und g...	Bratkartoffeln mit Speck und Zwi...
2	Kartoffeln mit Zwiebeln und o...	Bratkartoffeln mit Zwiebeln
3	grob geriebene Kartoffeln aus...	Drei Reibekuchen mit Apfelmus
4	mit Kartoffeln und Dilldip	Forellenfilet
5	Gesundes aus der Region, deft...	Gemüsepfanne

Abfrage

```
SELECT beschreibung, name FROM gericht
ORDER BY name ASC LIMIT 5 OFFSET 5;
```

Ergebnis

	beschreibung	name
1	Fein gewürzt und mariniert	Grilltofu
2	Nicht nur für Hacker	Hackbraten
3	Suppenhuhn trifft Petersilie	Hühnersuppe
4	der klassische Bauchwärmer mit frischen Kräutern	Kartoffel-Lauch-Suppe
5	schmeckt vor und nach dem Essen	Käsebrötchen

Abfrage

```
SELECT DISTINCT typ FROM allergen;
```

Ergebnis

	typ
1	Getreide (Gluten)
2	Allergen

Abfrage

```
SELECT name FROM gericht WHERE name LIKE 'K%';
```

Ergebnis

	name
1	Kartoffel-Lauch-Suppe
2	Käsebrötchen
3	Kassler mit Rosmarinkartoffeln

Abfrage

```
SELECT id, name FROM gericht WHERE name LIKE '%suppe%';
```

Ergebnis

	id	name
1	9	Hühnersuppe
2	11	Kartoffel-Lauch-Suppe
3	20	Suppenkreation á la Chef

Abfrage

```
SELECT * FROM kategorie WHERE eltern_id IS NULL;
```

Ergebnis

	id	name	eltern_id	bildname
1	1	Aktionen	<null>	kat_aktionen.png
2	2	Menüs	<null>	kat_menu.gif

Abfrage

```
UPDATE allergen
SET name = 'Kamut'
WHERE code = 'a6';
```

Ergebnis

	name
4	Gerste
5	Dinkel
6	Hafer
7	Kamut
8	Fisch
9	Krebstiere
10	Schwefeldioxid/Sulfit
11	Sellerie
12	Milch und Laktose
13	Butter
14	Käse
15	Margarine
16	Sesam
17	Nüsse
18	Mandeln
19	Haselnüsse
20	Walnüsse
21	Erdnüsse

Abfrage

```
INSERT INTO gericht (id, name, beschreibung, erfasst_am, vegetarisch, vegan, preis_intern, preis_extern)
VALUES (21, 'Currywurst mit Pommes', 'keine Ahnung', '2020-08-26', 0, 0, 1, 400);
```

Ergebnis

20	21 Currywurst mit Pommes	keine Ahnung
----	--------------------------	--------------

Abfrage

```
INSERT INTO gericht_hat_kategorie (gericht_id, kategorie_id)
VALUES (21, 3);
```


Ergebnis

	gericht_id	kategorie_id
11	16	5
12	17	5
13	18	5
14	21	3

Aufgabe 6

Abfrage

```
SELECT g.name, GROUP_CONCAT(a.name) FROM gericht_hat_allergen gha
      JOIN gericht g on g.id = gha.gericht_id
      JOIN allergen a on a.code = gha.code
GROUP BY g.name;
```

Ergebnis

	name	GROUP_CONCAT(a.name)
1	Bratkartoffeln mit Speck und Zwiebeln	Nüsse, Gerste, Dinkel, Krebstiere, Butter
2	Bratkartoffeln mit Zwiebeln	Butter, Kamut, Erdnüsse
3	Forellenfilet	Walnüsse, Schwefeldioxid/Sulfit, Milch und Laktose
4	Gemüsepfanne	Gerste
5	Grilltofu	Gerste, Butter, Dinkel, Walnüsse
6	Hackbraten	Mandeln, Roggen, Walnüsse, Krebstiere
7	Hühnersuppe	Roggen
8	Kassler mit Rosmarinkartoffeln	Käse, Mandeln, Hafer
9	Lasagne vegetarisch	Schwefeldioxid/Sulfit
10	Pilzpfanne	Erdnüsse
11	Pilzpfanne vegan	Weizen, Dinkel, Erdnüsse, Margarine, Walnüsse

Abfrage

```
SELECT g.name, GROUP_CONCAT(a.name) FROM gericht_hat_allergen gha
      RIGHT JOIN gericht g on g.id = gha.gericht_id
      LEFT JOIN allergen a on a.code = gha.code
GROUP BY g.name;
```

Ergebnis

name	GROUP_CONCAT(a.name)
1 Bratkartoffeln mit Speck und Zwiebeln	Nüsse, Gerste, Dinkel, Krebstiere, Butter
2 Bratkartoffeln mit Zwiebeln	Butter, Kamut, Erdnüsse
3 Currywurst mit Pommes	<null>
4 Drei Reibekuchen mit Apfelmus	<null>
5 Forellenfilet	Walnüsse, Schwefeldioxid/Sulfit, Milch und Laktose
6 Gemüsepfanne	Gerste
7 Grilltofu	Gerste, Butter, Dinkel, Walnüsse
8 Hackbraten	Mandeln, Roggen, Walnüsse, Krebstiere
9 Hühnersuppe	Roggen
10 Kartoffel-Lauch-Suppe	<null>
11 Käsebrötchen	<null>
12 Kassler mit Rosmarinkartoffeln	Käse, Mandeln, Hafer
13 Lasagne	<null>
14 Lasagne vegetarisch	Schwefeldioxid/Sulfit

Abfrage

```
SELECT a.name, GROUP_CONCAT(g.name) FROM gericht_hat_allergen gha
  JOIN gericht g on g.id = gha.gericht_id
RIGHT JOIN allergen a on a.code = gha.code
GROUP BY a.name;
```

Ergebnis

name	GROUP_CONCAT(g.name)
1 Butter	Bratkartoffeln mit Zwiebeln, Grilltofu, Bratkartoffeln mit Speck und Zwiebeln
2 Dinkel	Bratkartoffeln mit Speck und Zwiebeln, Grilltofu, Pilzpfanne vegan
3 Erdnüsse	Bratkartoffeln mit Zwiebeln, Pilzpfanne, Pilzpfanne vegan
4 Fisch	<null>
5 Gerste	Bratkartoffeln mit Speck und Zwiebeln, Grilltofu, Gemüsepfanne
6 Getreideprodukte	<null>
7 Hafer	Kassler mit Rosmarinkartoffeln
8 Haselnüsse	<null>
9 Kamut	Bratkartoffeln mit Zwiebeln
10 Käse	Kassler mit Rosmarinkartoffeln
11 Krebstiere	Hackbraten, Bratkartoffeln mit Speck und Zwiebeln
12 Mandeln	Hackbraten, Kassler mit Rosmarinkartoffeln
13 Margarine	Pilzpfanne vegan
14 Milch und Laktose	Forellenfilet
15 Nüsse	Bratkartoffeln mit Speck und Zwiebeln

Abfrage

```
SELECT k.name, COUNT(g.name) as total FROM gericht_hat_kategorie gha
  JOIN gericht g on gha.gericht_id = g.id
JOIN kategorie k on gha.kategorie_id = k.id
GROUP BY k.name
ORDER BY total ASC ;
```

Ergebnis

	name	total
1	Desserts	3
2	Vorspeisen	3
3	Hauptspeisen	8

Abfrage

```
SELECT temp.name, temp.total FROM (SELECT k.name, COUNT(g.name) as total FROM gericht_hat
  JOIN gericht g on gha.gericht_id = g.id
  JOIN kategorie k on gha.kategorie_id = k.id
 GROUP BY k.name
ORDER BY total ASC) as temp
WHERE total > 2;
```

Ergebnis

Analog wie oben

Aufgabe 7

```
CREATE TABLE IF NOT EXISTS gericht(
  id int(8) PRIMARY KEY,
  name varchar(80) NOT NULL UNIQUE,
  beschreibung varchar(800) NOT NULL ,
  erfasst_am date NOT NULL ,
  vegetarisch boolean NOT NULL DEFAULT false,
  vegan boolean NOT NULL DEFAULT false,
  preis_intern double NOT NULL CHECK ( preis_intern > 0 ),
  preis_extern double NOT NULL ,
  CHECK ( preis_intern <= preis_extern )
);
```

```

CREATE TABLE IF NOT EXISTS kategorie(
  id int(8) PRIMARY KEY,
  name varchar(80) NOT NULL ,
  eltern_id int(8) REFERENCES kategorie(id)
  ON UPDATE CASCADE
  ON DELETE NO ACTION,
  bildname varchar(200)
);

CREATE TABLE IF NOT EXISTS gericht_hat_allergen(
  code char(4) REFERENCES allergen(code)
  ON UPDATE CASCADE
  ON DELETE NO ACTION ,
  gericht_id int(8) NOT NULL REFERENCES gericht(id)
  ON UPDATE CASCADE
  ON DELETE NO ACTION
);

CREATE TABLE IF NOT EXISTS gericht_hat_kategorie(
  gericht_id int(8) NOT NULL REFERENCES gericht(id)
  ON UPDATE CASCADE
  ON DELETE NO ACTION,
  kategorie_id int(8) NOT NULL REFERENCES kategorie(id)
  ON UPDATE CASCADE
  ON DELETE NO ACTION
);

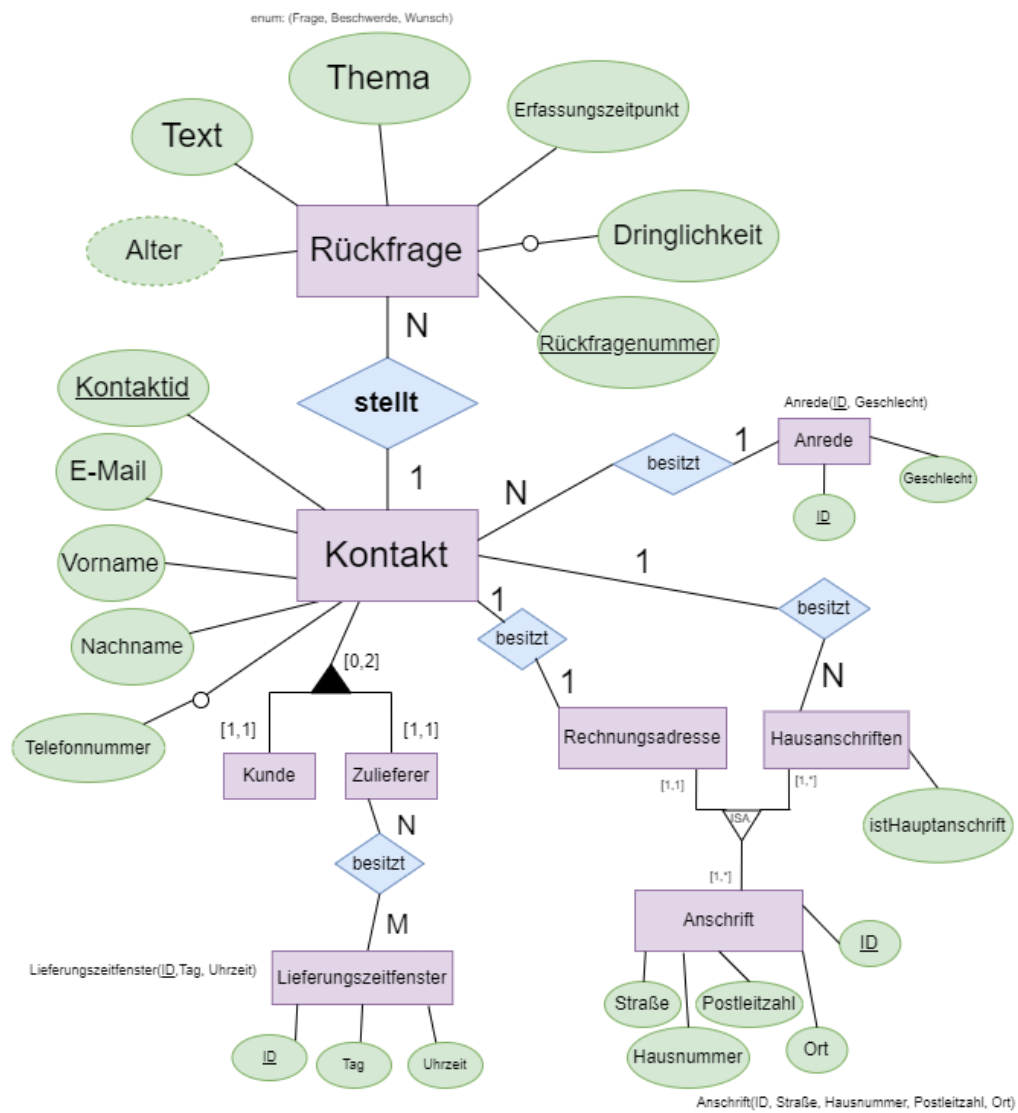
```

Aufgabe 11

Die Verwendung von Datenbanken gewährt die folgenden Vorteile:

- Dateien sind in Verbindung zueinander, heißt Sortierung oder Einfügen bzw. Löschen in der Datenbank ist viel effizienter.
- Nutzung von Standards: die Namen von Variablen, Tabellen, usw. sind eindeutig kategorisiert, was Duplikate mit Stichwörtern(alter, table, char) vermeidet.
- Wiederverwendung durchaus möglich und erwünscht wegen kürzere Entwicklungszeiten
- hohe Flexibilität und Verfügbarkeit der Daten

Rückfrage(Rückfragennummer, Text, Thema, Erfassungszeitpunkt, Dringlichkeit)

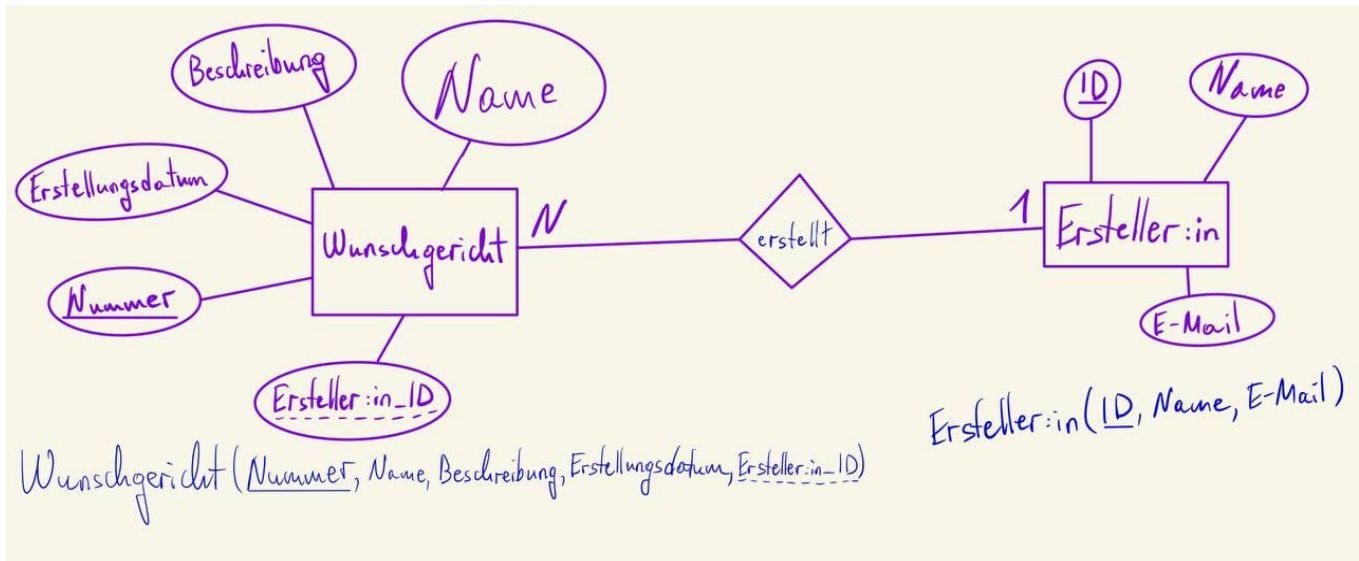


Kontakt(Kontaktid, E-Mail, Vorname, Nachname, Telefonnummer, Rückfrage)

Meilenstein 4

Aufgabe	Geschätzte Zeit	Tatsächliche Zeit	Kommentar
1	45	90	Abweichungsgrund: (Hristomir) Ich musste Abendessen kochen, sonst nichts Schwieriges.
2	40	40	-
3	40	60	Abweichungsgrund: (Hristomir) Ich musste das Projekt in einem neuen Ordner kopieren und eine neue Datenbank einrichten, damit das Ding läuft.
4	40	45	Abweichungsgrund: (Hristomir) Ich musste extra nach INDEX in SQL suchen. (hat sich aber gelohnt)
5	20	10	Abweichungsgrund: Quiz war leichter als erwartet.
6	10	10	-
7	40	90	Abweichungsgrund: (Zulfahmi) Erste Umgang mit Template Sprache
8	50	30	Abweichungsgrund: (Zulfahmi) eigentlich nur Copy & Paste und verteilen
Gesamte Zeit Meilenstein 3	6.25 Stunden(375 Minuten)		

Aufgabe 1



SQL CONSOLE:

```
CREATE TABLE IF NOT EXISTS Wunschgericht(  
    Nummer int NOT NULL AUTO_INCREMENT,  
    Name VARCHAR(22),  
    Beschreibung VARCHAR(100),  
    ID_Ersteller int,  
    Erstellungsdatum date,  
    PRIMARY KEY (Nummer)  
);
```

```
CREATE TABLE IF NOT EXISTS Ersteller(  
    ID int NOT NULL AUTO_INCREMENT,  
    Name VARCHAR(22) DEFAULT 'anonym',  
    EMail VARCHAR(22),  
    PRIMARY KEY (ID)  
);
```

Abfrage:

```
SELECT * FROM Wunschgericht, Ersteller WHERE Wunschgericht.ID_Ersteller =  
Ersteller.ID  
ORDER BY Erstellungsdatum LIMIT 5;
```


Ergebnis:

	Nummer	wunschgericht.Name	Beschreibung	ID_Ersteller	Erstellungsdatum	ID	ersteller.Name	E-Mail
1	1	'testGericht'	'Beschr'	2	2022-11-22	2	'testName'	'fd@l.com'
2	2	'testGericht'	'g'	3	2022-11-22	3	'testName'	'fd@l.com'
3	3	'Nasi Lemak'	'Based Malay meal'	4	2022-11-22	4	'Zulfahmi'	'zz@m.com'
4	4	'Moussaka'	'based Bulgarian meal'	5	2022-11-22	5	'Hristomin'	'hd@m.com'
5	5	'Döner'	'sheeeeeeeesh'	6	2022-11-22	6	'Andrew Tate'	'dd@ll.com'

Kommentar: ID_Ersteller beginnt ab 2, weil ich aus Versehen die Seite neu gestartet habe.
Kein fataler Fehler.

Aufgabe 2

Änderungen am Quelltext in **/werbeseite** Ordner sind wie folgt:

gerichte.php

array_walk_recursive() Funktion hinzugefügt, um htmlspecialchars relativ zu jedem Element im Array \$gerichte anzuwenden

Newsletteranmeldung.php

Funktionen checkName und checkMail liefern String-Wert zurück, der mit htmlspecialchars gefiltert wird

werbeseite.php

Whitelisting von Sprachen eingesetzt

wunschgericht.php

Konkatenation von jeder abgefragten Dateneinheit mit '/' vorne und hinten
htmlspecialchars für jede Variable mit Daten aus DB

```
mysqli_real_escape_string($database_connect, $gerichtName);
```

gegen SQLI

Aufgabe 3

- Sind Dateien, Variablen, Funktionen, usw. zweckdienlich benannt? - Ja
- Existieren Fehler im Quelltext?
 - keine, Code hat funktioniert
- Was würden Sie anders strukturieren?
 - Anmeldeformular würde ich so strukturieren:

Ihr Name:

Vorname

Ihre Email:

Newsletter bitte in:

DE ✓

☐ Den...

Zum Newsletter anm.

Links zu Sections würde ich in der Mitte haben und die 3 Punkte unter "Das ist uns wichtig" würde ich im Grid links haben

- Was fällt Ihnen sonst am Quelltext auf?
 - strpos() wurde benutzt, würde stripos() benutzen, damit Suche nicht case-sensitive ist
 - Grids ganz effektiv genutzt, sodass Seite gut strukturiert ist
- Was haben Sie selbst durch das Lesen des anderen Quelltexts gelernt?
 - CONCAT_WS() und SEPARATOR als Zusatzfunktionen zu GROUP_CONCAT()
 - alert() Funktion
- Was würden Sie jetzt bei Ihrem Quelltext anpassen?
 - Allergene in einer kleineren Tabelle unten darstellen
 - Footer angepasst

Fazit: Code ist in der Regel gut lesbar und verständlich. Man kann gut nachvollziehen, was für Ergebnisse die SQL-Statements liefern und wie sie in der Implementierung verwendet werden. Es ist auch ganz nice, dass das andere Team die Allergene in einer kleineren Tabelle angezeigt hat und ganz geschickt CSS-Spezifitäten gewichtet hat.:

Aufgabe 4

1)

```
ALTER TABLE gericht_hat_kategorie
ADD UNIQUE INDEX unique_gericht_kategorie (gericht_id,
kategorie_id);
```

2)

```
ALTER TABLE gericht
ADD UNIQUE INDEX idx_name(name);
```

siehe:

<https://www.tutorialspoint.com/sql/sql-indexes.htm#:~:text=An%20index%20in%20a%20database.or%20more%20specific%20page%20numbers>

3), 4), 5)

```
ALTER TABLE gericht_hat_kategorie
ADD CONSTRAINT gericht_hat_kategorie_fk
FOREIGN KEY (gericht_id)
REFERENCES gericht(id)
ON UPDATE CASCADE
ON DELETE CASCADE;
```

```
ALTER TABLE gericht_hat_allergen
ADD CONSTRAINT gericht_hat_allergen_fk
FOREIGN KEY (gericht_id)
REFERENCES gericht(id)
ON UPDATE CASCADE
ON DELETE CASCADE
```

6)

```
ALTER TABLE gericht_hat_kategorie ADD PRIMARY KEY (gericht_id,
kategorie_id);
```

Meilenstein 5

Aufgabe	Geschätzte Zeit	Tatsächliche Zeit	Kommentar
1	180	135	Abweichungsgrund: Optional Aufgaben sind nicht erledigt
2	60	45	Abweichungsgrund: nur Bilder eintragen
3	30	60	Abweichungsgrund: muss Vorlesung wiederholen(Zulfahmi) musste länger als erwartet googeln (Hristomir)
4	30	30	-
5	45	30	Abweichungsgrund: nur eine Prozedur erstellen
Gesamte Zeit Meilenstein 5	5 Stunden(300 Minuten)		

Aufgabe 1

1)

CREATE TABLE IF NOT EXISTS benutzer

(

id INT(8) PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(200) NOT NULL,
email VARCHAR(100) NOT NULL UNIQUE,
passwort VARCHAR(200) NOT NULL,
admin BOOLEAN DEFAULT false,
anzahlFehler int NOT NULL DEFAULT 0,
anzahlFnmeldungen int NOT NULL NOT NULL,
letzteAnmeldung DATETIME NOT NULL,
letzterFehler DATETIME

);

2)

session_start(); in **index.php**(Frontcontroller) hinzugefügt

3)

Adminkonto:

- Email: admin@emensa.example.com
- Passwort: example
- Salt: prefix
- Hash: 615a9a551452e04155fa346a85ec85010078ac99

Aufgabe 4

view_suppengerichte

	! name
1	Hühnersuppe
2	Kartoffel-Lauch-Suppe
3	Suppenkreation á la Chef

view_anmeldungen(kann variabel sein)

	! anzahlAnmeldungen
1	2

view_kategoriegerichte_vegetarisch

gerichtname	kategorienname
1 <null>	Aktionen
2 Käsebrötchen,Tomatenbrötchen	Desserts
3 <null>	Erstiewoche
4 Bratkartoffeln mit Zwiebeln,Grilltofu,Lasagne vegetarisch	Hauptspeisen
5 <null>	Mensastars
6 <null>	Menus
7 Käsebrötchen,Tomatenbrötchen	Vorspeisen

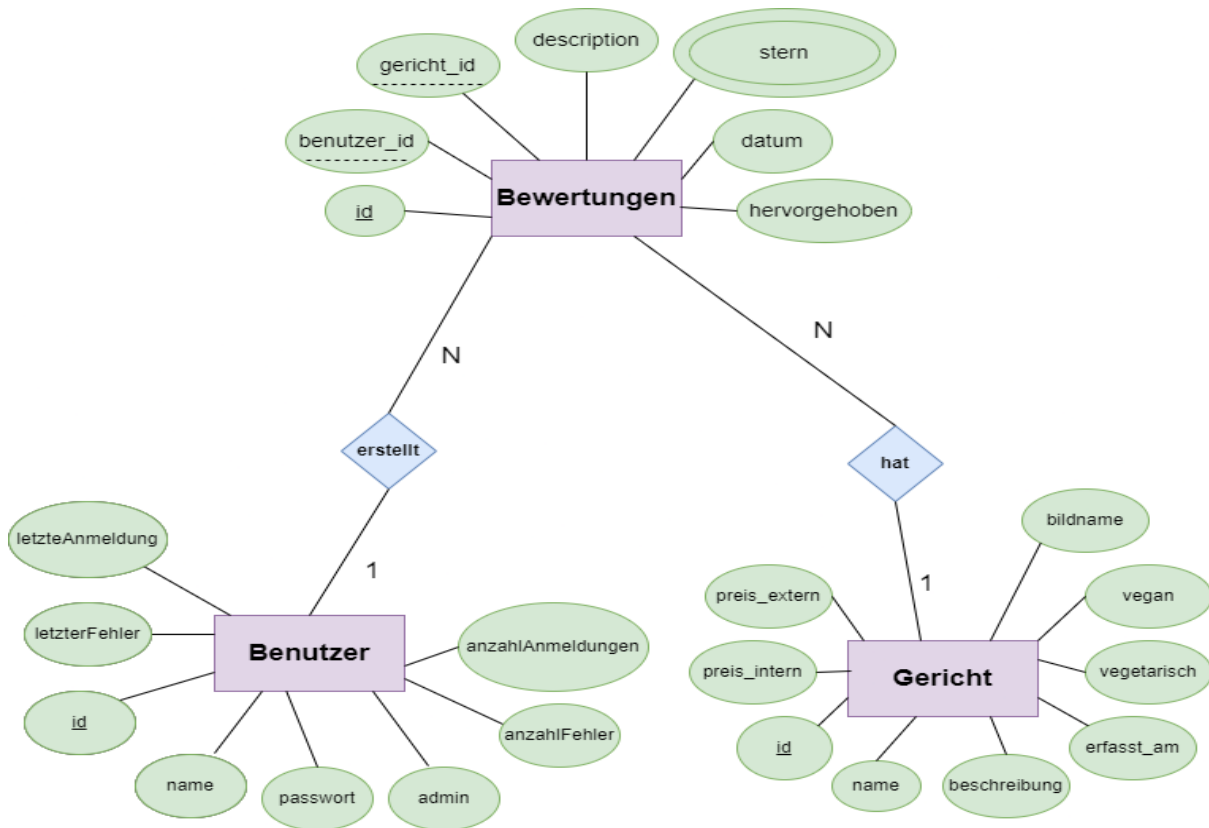
Aufgabe 5

- 1) Anzahl Fehler - Es wird immer neu in jedem Session berechnet**
- 2) letztes Anmeldedatum - Es wird immer neu in jedem Session berechnet**

Meilenstein 6

Aufgabe	Geschätzte Zeit	Tatsächliche Zeit	Kommentar
1	60	60	Kenntnisse von Meilenstein 4 waren sehr hilfreich.
2	40	60	Abweichungsgrung: Aufgabenstellung falsch verstanden, erstmal Checkbox gemacht anstatt von Link(Zulfahmi)
Gesamte Zeit Meilenstein 6	120 Minuten		

Aufgabe 1



- Bewertungen(id, description, stern, datum, hervorgehoben, benutzer_id, gericht_id)
- Benutzer(id, name, password, admin, anzahlFehler, anzahlAnmeldungen, letzterFehler, letzteAnmeldung)
- Gericht(id, name, beschreibung, erfasst_am, vegetarisch, vegan, bildname, preis_intern, preis extern)