Ein Projekt von: Nilusche Liyanaarachchi und Luis Diniz Do Carmo

**Link für die Webversion des Dossiers:**   
  
[dbwtdossier-docarmoliyanaarachchi.rf.gd](http://dbwtdossier-docarmoliyanaarachchi.rf.gd/)

# Meilenstein 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Aufgabe** | **Geschätzte Zeit** | **Tatsächliche Zeit** | **Sonstiges** |
| 1 | 10 Minuten | 10 Minuten |  |
| 2 | 2 Minuten | 2 Minuten |  |
| 3 | 10 Minuten | 8 Minuten |  |
| 4 | 25 Minuten | 35 Minuten |  |
| 5 | 30 Minuten | 40 Minuten |  |
| 6 | 20 Minuten | 30 Minuten |  |
| 7 | 15 Minuten | 8 Minuten |  |
| 8 | 30 Minuten | 25 Minuten |  |
| 9 | 30 Minuten | 60 Minuten | Genutzt wurden   * Font-Family * Font-Size * Margin-Left * Font-Weight * Line-Height * Color * Text-Decoration * Border * Border-Color * Border-Top * Display: inline * Border-Left * Padding-Left * Border-Left-Color |
| 10 | 150 Minuten | Ca. 480 Minuten | Vorgehensweise: Wir haben das Website-Mock-Up unterteilt. Dann hat jeder seinen Teil in HTML geschrieben. Kleine Verbesserungen und Anpassungen wurden in Absprache getätigt.  Schließlich wurde zuletzt in Zusammenarbeit das Stylesheet geschrieben.  Der Arbeitsaufwand war deutlich höher als erwartet. Das lag vor allem, an der Stilbearbeitung. Wir haben viele Möglichkeiten ausprobiert, was sehr Zeitintensiv war. |

# Meilenstein 2

|  |  |  |  |
| --- | --- | --- | --- |
| Aufgabe | Geschätzte Zeit | Tatsächliche Zeit | Sonstiges |
| 1 | 10 Minuten | 8 Minuten |  |
| 2 | 45 Minuten |  |  |
| 3 | 60 Minuten | 160 Minuten | Aufgabenteil a:   * In Zeile 12 hat ein Komma gefehlt * In Zeile 22 hat ein ‚]‘ gefehlt * In Zeile 43 hat eine Klammer gefehlt * In Zeile 62 hat das Schlüsselwort ‚Funktion‘ gefehlt   Aufgabenteil b:   * Enthält drei Listen: Ratings, Allergenen und Details zum Gericht * Listet alle Bewertungen auf und bietet eine Filtermöglichkeit an. * Berechnet den Durchschnitt aller Bewertungen * PHP-Großteil wird vor dem HTML-Teil programmiert. Anschließend folgt der HTML-Teil mit Einbindung von PHP |
| 4 | 5 Minuten | 3 Minuten |  |
| 5 | 30 Minuten | 80 Minuten |  |
| 6 | 30 Minuten | 20 Minuten |  |
| 7 | 30 Minuten | 40 Minuten |  |

# Meilenstein 3

|  |  |  |  |
| --- | --- | --- | --- |
| Aufgabe | Geschätzte Zeit | Tatsächliche Zeit | Sonstiges |
| 1 | 30 Minuten | 25 Minuten |  |
| 2 | 50 Minuten | 50 Minuten |  |
| 3 | 30 Minuten | 20 Minuten |  |
| 4 | 30 Minuten | 35 Minuten |  |
| 5 |  |  |  |

## Zu Aufgabe 3.2

create DATABASE emensawerbseite if not exists;  
  
use emensawerbseite;  
  
create table gericht(  
 id bigint PRIMARY KEY,  
 name varchar(80) not null unique,  
 beschreibung varchar(800) not null,  
 erfasst\_am date not null,  
 vegetarisch boolean not null,  
 vegan boolean not null,  
 preis\_intern double not null,  
 preis\_extern double not null CHECK (preis\_extern > preis\_intern)  
);  
  
create table allergen(  
 code char(4) PRIMARY KEY,  
 name varchar(300) not null,  
 typ varchar(20) not null  
);  
  
create table kategorie(  
 id bigint primary key auto\_increment,  
 name varchar(80) not null,  
 eltern\_id bigint,  
 bildname varchar(200)  
);  
  
create table gericht\_hat\_allergen(  
 code char(4),  
 gericht\_id bigint not null,  
 foreign key (code) references allergen(code),  
 foreign key (gericht\_id) references gericht(id)  
  
);  
create table gericht\_hat\_kategorie  
(  
 gericht\_id bigint not null,  
 kategorie\_id bigint not null,  
 foreign key (gericht\_id) references gericht (id),  
 foreign key (kategorie\_id) references kategorie (id)  
);

und

alter database emensawerbseite Collate = 'utf8mb4\_unicode\_ci';

alter table allergen Collate = 'utf8mb4\_unicode\_ci';  
alter table gericht Collate = 'utf8mb4\_unicode\_ci';  
alter table gericht\_hat\_allergen Collate = 'utf8mb4\_unicode\_ci';  
alter table gericht\_hat\_kategorie Collate = 'utf8mb4\_unicode\_ci';  
alter table kategorie Collate = 'utf8mb4\_unicode\_ci';

## Zu Aufgabe 3.3

use emensawerbeseite;  
  
select Count (\*) as 'Anzahl Zeilen'  
from allergen;  
  
select Count(\*) as 'Anzahl Gerichte'  
from gericht;  
  
select Count(\*) as 'Gericht hat Allergen'  
from gericht\_hat\_allergen;  
  
select Count(\*) as 'Gericht hat Kategorie'  
from gericht\_hat\_kategorie;  
  
select Count(\*) as 'Kategorie'  
from kategorie;

## Zu Aufgabe 5

1. 
2. 
3. 
4. 
5. ?
6. Ein Bild, das Text enthält.

   Automatisch generierte Beschreibung
7. Ein Bild, das Text enthält.

   Automatisch generierte Beschreibung
8. Ein Bild, das Text enthält.

   Automatisch generierte Beschreibung
9. Ein Bild, das Text enthält.

   Automatisch generierte Beschreibung
10. Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
11. Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
12. Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
13. Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
14. Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
15. Ein Bild, das Text, Gerät, Anzeige enthält.

    Automatisch generierte Beschreibung
16. Ein Bild, das Text enthält.

    Automatisch generierte BeschreibungEin Bild, das Text enthält.

    Automatisch generierte Beschreibung
17. Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung

# Meilenstein 4

|  |  |  |  |
| --- | --- | --- | --- |
| Aufgabe | Geschätzte Zeit | Tatsächliche Zeit | Sonstiges |
| 1 | 30 Minuten | 35 Minuten | Wunschgericht (Erstellungsdatum, ID, Gerichtname, (gestrichelt: ErstellerID))  Ersteller(Email, Name)  Das ERD-Diagramm befindet sich unter /beispiele |
| 2 | 25 Minuten | 25 Minuten | * Newsletter Anmeldung war anfällig für CSRF * Wunschgericht war anfällig für CSRF und SQL Injection   Behebung:  CSRF durch Einführung eines Tokens, der bei jedem Aktualisieren  der Seite neu generiert wird. SQL Injections durch prepared Statements. |
| 3 | / | / |  |
| 4 | 20 Minuten | 15 Minuten |  |
| 5 | 40 Minuten | 50 Minuten | Wir haben das bestehende Projekt an dieser Stelle sofort in ein  Laravel Projekt umgewandelt. |
| 6 | 35 Minuten | 30 Minuten |  |
| 7 | 20 Minuten | 25 Minuten | Größtenteils haben wir diese Aufgabe schon im Aufgabenteil 5 erledigt.  Daher der geringe Zeitaufwand |
| 8 | 35 Minuten | 50 | 1)    Alternativ auch gericht\_id und kategorie\_id zu einem gemeinsamen  Primärschlüssel benennen    2)    3)  Alle Foreign Keys  SELECT TABLE\_NAME,      COLUMN\_NAME,      CONSTRAINT\_NAME,      REFERENCED\_TABLE\_NAME,      REFERENCED\_COLUMN\_NAME  FROM INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE  where REFERENCED\_TABLE\_NAME is not Null;  Alter table gericht\_hat\_kategorie  drop foreign key  gericht\_hat\_kategorie\_ibfk\_1;  Alter Table gericht\_hat\_kategorie  ADD FOREIGN KEY (gericht\_id) REFERENCES gericht(id) on DELETE CASCADE;  Zuerst bestehenden Key löschen, dann neuen hinzufügen      4)  Eigenschaft 1)    Eigenschaft 2)    5)  Löschen des Fremdschlüssels    Neuerstellen des Fremdschlüssels    6) |
|  |  |  |  |