

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Zeiterfassung Luke Braun	3
Zeiterfassung Kai Fedin	4
Meilenstein 1	5
Aufgabe 1	5
Aufgabe 6	5
Aufgabe 9	6

Task	Status	Responsibility	Hinweise	Difficulty
A1.1	Nicht gestar... ▾			easy
A1.2	Nicht gestar... ▾			easy
	Nicht gestar... ▾			

Zeiterfassung Luke Braun

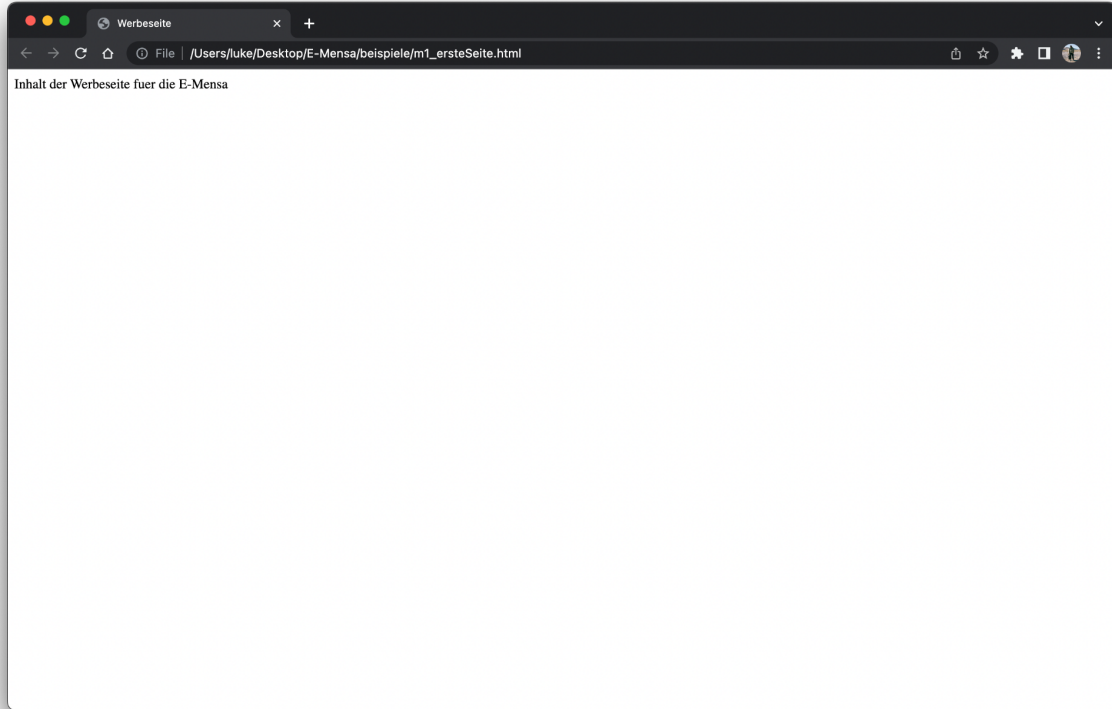
Meilenstein	Aufgabe	Geschätzte Zeit	Tatsächliche Zeit
M4	A1	60 min	90 min
M4	A2	30 min	45 min
M4	A3	60 min	60 min
M4	A4	60 min	105 min
M4	A5	20 min	20 min
M4	A6	0 min	0 min
M4	A7	60 min	120 min
M4	A8	240 min	240 min

Zeiterfassung Kai Fedin

Meilenstein	Aufgabe	Geschätzte Zeit	Tatsächliche Zeit
M5	A1	6h	5h
M5	A2	1h	1h
M5	A3	1h	0,5h
M5	A4	1h	1,5
M5	A5	1h	0,75h

Meilenstein 1

Aufgabe 1



Aufgabe 6

Request Method: GET

→ Fordert Ressourcen vom Server an

Status Code 200 OK

→ Website wurde gefunden

Accept-Encoding: gzip, deflate, br

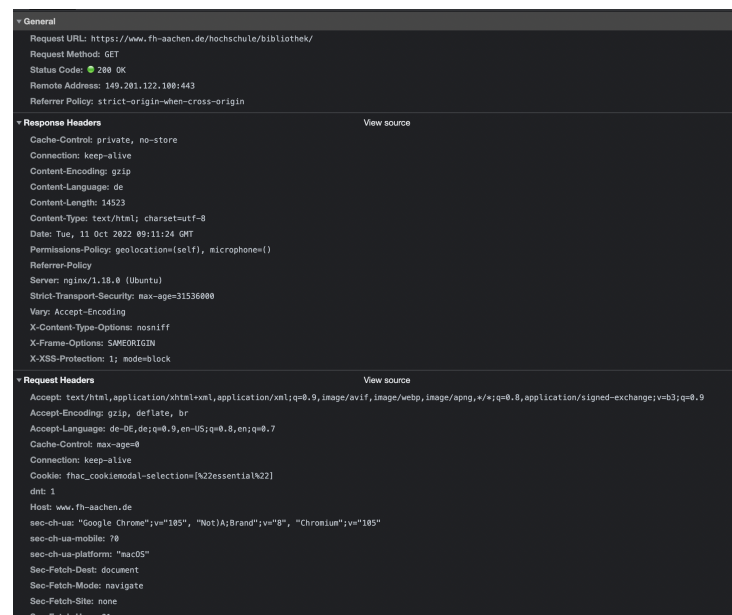
→ Gibt erlaubte und präferiertes coding an

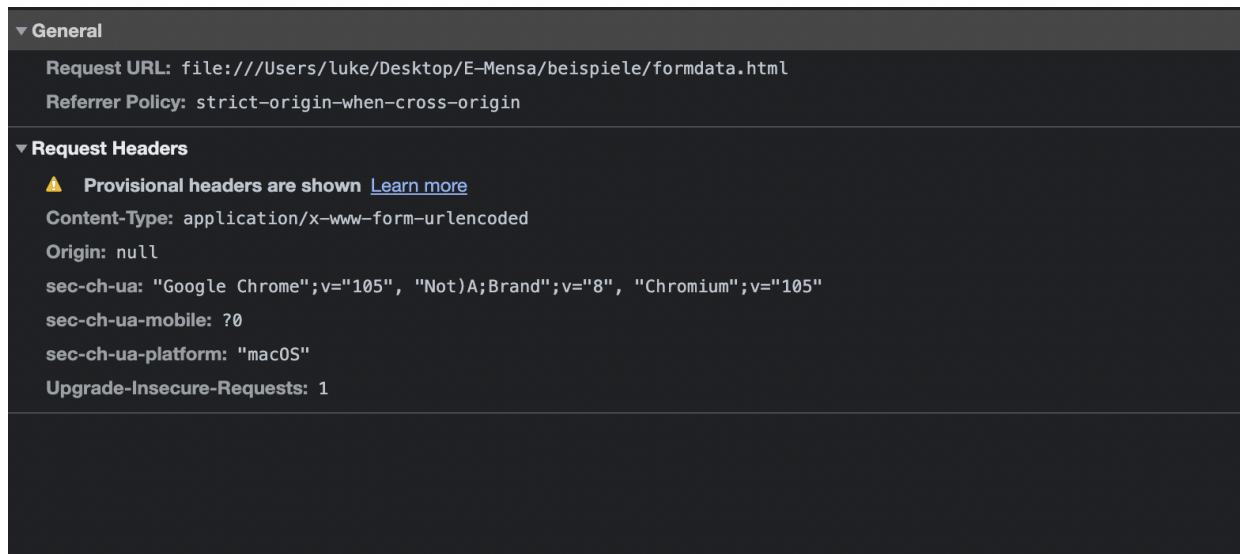
Host: www.fh-aachen.de

→ Liefert infos über Host und Ports

Server: nginx/1.18.0 (Ubuntu)

→ Liefert infos über den Typ des Webserver





Aufgabe 9

Verwendete CSS Eigenschaften

- font-family: Arial
- font-weight: bold
- text-decoration: none
- text-align: center
- margin-left: auto
- margin-right: auto
- border: 2px solid black (a, footer, etc.)
- padding: 0px
- color: black
- background-color: rgb(...)
- display: flex
- border-top: 1px solid orange
- justify-content: end

Verwendete CSS Selektoren

- *
- Typ Selektor (a)
- Klassen Selektor
- ID Selektor
- Vererbung

Meilenstein 2

Aufgabe 3

1)

```

8      $allergens = [
9          11 => 'Gluten',
10         12 => 'Krebstiere',
11         13 => 'Eier',
12         14 => 'Fisch',
13         17 => 'Milch'
14     ];

```

```

8      $allergens = [
9          11 => 'Gluten',
10         12 => 'Krebstiere',
11         13 => 'Eier',
12         14 => 'Fisch',
13         17 => 'Milch'
14     ];

```

```

20         'price_extern' => 3.90,
21         'allergens' => [11, 13,
22         'amount' => 42
23     ];

```

```

20         'price_extern' => 3.90,
21         'allergens' => [11, 13],
22         'amount' => 42
23     ];

```

```

41     if (!empty($_GET[GET_PARAM_SEARCH_TEXT])) {
42         $searchTerm = $_GET[GET_PARAM_SEARCH_TEXT];
43         foreach ($ratings as $rating) {

```

```

41     if (!empty($_GET[GET_PARAM_SEARCH_TEXT])) {
42         $searchTerm = $_GET[GET_PARAM_SEARCH_TEXT];
43         foreach ($ratings as $rating) {

```

```

59     calcMeanStars(array $ratings): float {
60         $sum = 1;
61         foreach ($ratings as $rating) {

```

```

59     function calcMeanStars(array $ratings): float{
60         $sum = 1;
61         foreach ($ratings as $rating) {

```

2)

Anzeige der Reviews

Suche der Reviews

Berechnung der Durchschnitts-Bewertungen

3)

Unbekannte Sprachelemente

strpos(\$val1, \$val2) - sucht erstes vorkommen von val2 in val1 : returned false wenn nicht gefunden

stripos() - gleiche methode wie strpos nur case-insensitiv

```
array_push(array &$amp;array, mixed ...$values): int
```

entspricht

```
$array[] = $var;
```

Funktionalität Suche:

Durch den Aufruf der Suche wird ein Get-Request getriggert welches eine neue URL aufruft:

http://localhost:9050/?search_text={Sucheingabe}

Beim Aufruf der Seite wird nun auf den Parameter Search Text

“GET_PARAM_SEARCH_TEXT” zugegriffen um alle Reviews in ein showRatings Array zu speichern welche im Titel diesen Search Text enthalten

Meilenstein 4

Aufgabe 2

Im Falle der Emensa Webseite haben wir es nur mit einer Bedrohung durch SQL Injection zu tun. Diese kann innerhalb des Formulars für Wunschgerichte auftreten, da in diesem Fall die eingegebenen Parameter des Users in das SQL Statement übertragen werden. Zur Absicherung wurde hier mit der `mysqli_real_escape_string()` Funktion gearbeitet. Sie escaped alle SQL Statements und sorgt dafür das die Datenbank sie als reinen Text ansieht was ein Injection verhindert

```
/**
 * Security Measures
 * mysqli_real_escape_string() escapes all strings that are passed
 */

$name = mysqli_real_escape_string($link, $_POST[POST_GERICHT_NAME]);
$beschreibung = mysqli_real_escape_string($link, $_POST[POST_GERICHT_BESCHREIBUNG]);
$creator = mysqli_real_escape_string($link, $_POST[POST_GERICHT_ERSTELLER]);
$email = mysqli_real_escape_string($link, $_POST[POST_GERICHT_EMAIL]);
```

Meilenstein 5

Aufgabe 5

Man könnte im allgemeinen die Benutzer Updates als Prozeduren aufbauen

```
$user->anzahlfehler_update();
$user->letzterfehler_update();
```