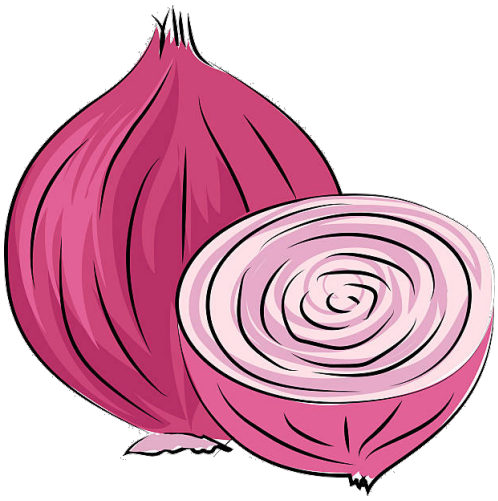

Rezepti

Christoph Sherr , Tobias Lotz, Dominik Hummel

June 2023

1 Kurzbeschreibung:

Eine Seite Für Kochrezepte von verschiedenen Ländern all around the globe.

2 Zielgruppe Nutzer und Beeinflusser:

Unsere Zielgruppe sind alle Personen die Kochen wollen. Nutzer könnten alle Personen sein die ein bestimmten Rezept suchen oder einfach Rezepte Endecken wollen. Beeinflusser können Firmen sein die unseren konntet für ihre Zwecke nutzen wollen zb. Influenzer die Rezepte probieren oder Firmen die Produkte durch Rezepte vermarkten wollen.

3 Funktionen

- Rezepte Filtern
- Rezepte suchen
- Anmeldung/Registrierung
- Rezept erstellen
- Rezept bewerten
- Rezept unter Favoriten speichern
- Rezepte Endecken

4 Referenzdokument

Ziel war es, eine Plattform zu schaffen, auf der Rezepte weltweit ausgetauscht werden können. Jeder Nutzer soll mit Hilfe eines Accounts nach Rezepten suchen und eigene Rezepte veröffentlichen können.

Die Nutzer sollen Accounts anlegen, Rezepte einsehen und neue Rezepte erstellen können. Es soll auch möglich sein, Rezepte ohne Anmeldung anzusehen. Es soll möglich sein, Rezepte nach Zutaten, Kategorien und anderen Merkmalen zu suchen. Außerdem soll es einen Admin-Bereich geben, in dem die Seite verwaltet werden kann. Die Seite sollte intuitiv bedienbar sein, unabhängig vom Endgerät des Nutzers.

Der Projektrahmen bestand hauptsächlich aus der Planung, der Implementierung und der Dokumentation des Ergebnisses. Die Implementierung nahm die meiste Zeit in Anspruch, ging aber teilweise mit der Planung einher.

Für die Implementierung war bereits bei der Planung klar, dass wir ein Versionskontrollsystem benötigen, wofür wir Git zusammen mit GitHub verwendeten, was uns auch bei der Planung half, da wir einen Backlog in Github Projects einrichten konnten.

Wir entschieden uns, Docker Compose für unser Projekt zu verwenden, damit jeder eine lauffähige Entwicklungsumgebung hatte. So konnte die Entwicklungsumgebung einfach auf jeder Maschine identisch zum Laufen gebracht werden. Dies war auch für die Synchronisation der Datenbank nützlich. So konnten wir beim Start eine gedumpte SQL Datenbank laden. Für das Backend erstellten wir eine allgemeine Klasse, über die man in jeder PHP Datei auf die Datenbank zugreifen konnte. Für das Frontend haben wir zunächst Vanilla CSS verwendet.

Die Verantwortlichkeiten wurden bereits in der Planung festgelegt: Domme sollte sich hauptsächlich um das Frontend und das Design der Website kümmern, Tobias sollte sich um das Backend und die Datenbank kümmern. Christoph hatte die Aufgabe, den Überblick zu behalten und das Frontend mit dem Backend zu verbinden.

5 Komplettes Product-Backlog als Tabelle

Im Verlauf des Projektes haben wir die "Projects" Funktion von GitHub verwendet, welche u.a. ein Kanban Board zu organisation anbietet. Dieses kann unter folgendem Link gefunden werden:
`https://github.com/users/PlexSheep/projects/2`

Aufgabe	Status
Dokumentation	done
Use local dependencies instead of CDN	done
AJAX	done
fix xss in search	done
Benutzerverwaltung	done
synchronized header and footer	done
logout	done
Docker environment for windows	done
Kurzbeschreibung	done
user profile page	done
Rezepte erstellen	done
Startseite	done
detail view	done
Rezept search	done
jquery ui integration	done
jquery integration	done
load shared html from central files as templates	done
search for ingridients	done
search for tags	done
fix potential sql injections	done
redisign detail view to be fancy	done
implement basic search	done
upload images to on creation	done
automate user creation for db	done
Referenzdokument	done
fix website for firefox	done
switch from vanilla css to bootstrap	done
store images in the upload directory	done
automatically load recipies from database	done
add a favicon	done
use font available on any system	done
data sanitisation	done
password management	done
admin section	done
session management	done
Database structure	done

6 Technologien/Requirements

- Clienttechnologie:
  - CSS
  - JS
  - jQuery
  - jQueryUI
  - Bootstrap o.ä als Addon-Frameworks
  - HTML5
- Servertechnologie:
  - PHP (8.0)
  - MariaDB (MySQL) kein PHP-Framework wie Synfony o.ä
  - nginx
- Funktionelle Vorgaben:
  - Sessionverwaltung
    - Wir in der common.php datei (l.7) gestartet
    - Session expire in (l. 9 - 17)
    - Session abfrage z.B: in admin.php (l.14-19)
  - Login-Logout; automatisches Logout nach bestimmter Zeit
    - Session expire in (l. 9 - 17)
  - Admin-Bereich mit Useraccount-Verwaltung
    - In admin.php
    - Nutzerverwaltung in den JavaScript Funktionen
  - Passwort-Verwaltung
    - In admin.php (l.105-129)
  - Rechte-Verwaltung (Adminrechte-Userrechte)
    - Die Rechteverwaltung wird hier mit in Sessionvariablen gespeicherten IDs geregelt. z.B: in admin.php (l.14-19)

- Dateneingaben client- und serverseitig prüfen (RegExp)
  - Implementiert in common.php test_for_bad_chars, test_for_bad_chars_array. Die Funktionen selbst sind in der datei common.php (l.341-345) und (l.347-353) deklariert.
- Datensatzmanipulation in SQL-Server (speichern; auslesen/ausgeben; bearbeiten; löschen)
  - Implementiert an vielen stellen im Projekt. Die Rezepte, welche auf der Startseite augegeben werden, werden aus der Datenbank geladen
    z.B: featured_recipies (l.5-8 ...), gespeichert in z.: admin.php (l.137-149), bearbeitet in z.B: admin.php (l.105-132), gelöscht in z.B: admin.php (l.72-99).
- Konfigurationsdaten via Konfigurationsdatei einlesen
  - Eine Konfigurationsdatei wird mit (PHP INI) realisiert. Diese wird in common.php (l.20-55) geladen und deren Daten verwendet.
- Einbindung von jQuery und jQuery UI
  - JQery und JQuery UI werden in der admin.php Datei verwendet, um z.B: einen Confirmation-Dialog zu realisieren, der den Nutzer fragt, ob eine aktion ausgefhrt werden soll, sowie auch um AJAX einzubinden.
- Dynamische laden/nachladen mit Hilfe von AJAX
  - Realisiert in admin.php (siehe oben)
- Meldungsfenster und Userbestätigungen mit jQuery und jQuery UI
  - Realisiert in admin.php (siehe oben)
- Datenexport via JSON
  - Realisiert in get_user_data.php
- Dynamische Menüstruktur mit responsive Webdesign
  - Auf der gesammten Seite vorhanden

7 Verzeichnisstruktur

Das Projekt hat mehrere Verzeichnisse, die nur für das Setup der Dockercontainer relevant sind. Diese sind die nginx, php, site.conf und dev Verzeichnisse. Der tatsächliche code der Webapp befindet sich im Verzeichnis webdir. In diesem Verzeichnis werden die haupt customer-facing Seiten gespeichert. In den in webdir enthaltenen Unterordnern admin, auth, und templates werden jeweils die Dateien für die Administarorenseiten, Authentifizierung und Templates gespeichert.

Figure 1: Directories

8 Beschreibung der Datenstrukturen und Tabellen

Für die Datenbank wurde nach Vorgabe MariDB verwendet. Diese Datenbank wurde in einem Dockercontainer gehostet und dann zur Bereitstellung der Daten angebunden.

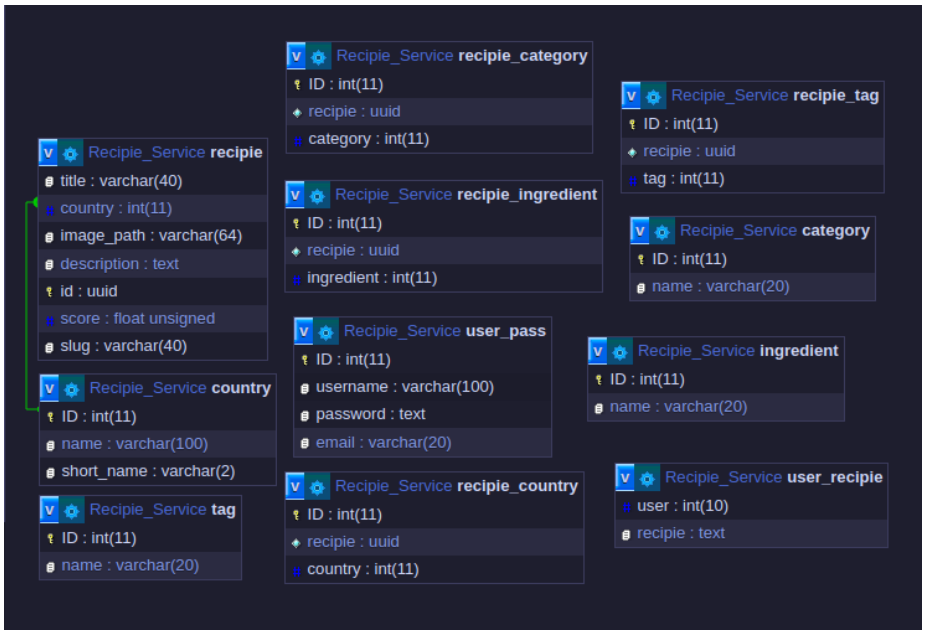

Figure 2: Datenbank Tabellen

8.1 recipie

Haupttabelle der Rezepte
title: Titel des Rezepts
country: Land das dem Rezept zugewiesen wurde
image_path: ID des Bildes, welchen in der webdir/img gespeichert werden.
description: Beschreibung des Rezeptes
id: uuid des Rezeptes
score: Bewertung des Rezeptes (Nicht genutzt)
slug: Referenzierung in der URL

8.2 country

Tabelle für die Länder ID: ID des Landes
name: Name des Landes

short_name: Landeskürzel

8.3 tag

Tabelle für Tags ID: ID des Tags
name: Name des Tags

8.4 recipie_category

Verbindngstabelle für Rezepte und Länder ID: Primary key
recipie: ID des Rezeptes
category: ID der Kategorie

8.5 recipie_ingridient

Verbindngstabelle für Rezepte und Zutaten ID: Primary key
recipie: ID des Rezeptes
ingidient: ID der Zutat

8.6 user_pass

Tabelle für die Core Userdaten ID: ID der User
username: Nutzername der User
password: Passwort der Nutzer
email: Email der Nutzer

8.7 recipie_country

Verbindngstabelle für Rezepte und Länder ID: Primary key
recipie: ID des Rezeptes
country: ID des Landes

8.8 recipie_tag

Verbindngstabelle für Rezepte und Tags ID: Primary key
recipie: ID des Rezeptes
tag: ID des Tags

8.9 category

Tabelle für die Kategorieen ID: Primary key
recipie: ID des Rezeptes
name: Name der Kategorie

8.10 ingidient

Tabelle für die Zutaten ID: Primary key
recipie: ID des Rezeptes
name: Name der Zutat

8.11 user_recipie

Verbindngstabelle für User und Rezepte user: ID des Users
recipie: ID des Rezeptes

9 Installationsanleitung