Dokumentation Cooked-Web-Applikation

a) Spezifikation und Design

Die Web-Applikation Cooked ist eine digitale Rezeptplattform, die Nutzern eine übersichtliche und benutzerfreundliche Möglichkeit bietet, Rezepte zu entdecken, zu speichern und eigene Gerichte zu erstellen. Ziel ist es, eine werbefreie Umgebung zu schaffen, die das Kochen angenehm, einfach und inspirierend gestaltet.

Wichtige Use Cases

- Rezeptübersicht anzeigen
- Rezeptdetails anzeigen
- Rezept erstellen
- Rezept bearbeiten
- Rezept löschen
- Favorit hinzufügen / entfernen
- Bewertung abgeben
- Wochenplan-Eintrag erstellen
- Einkaufsliste verwalten
- Profil bearbeiten
- Admin: Nutzer verwalten

Anpassungen zur ursprünglichen Spezifikation

Während der Umsetzung wurden einige Funktionen erweitert. Ergänzt wurden ein Wochenplan zur Rezeptplanung, eine automatisch generierte Einkaufsliste, ein Favoriten-System mit Herz-Button sowie ein Admin-Bereich zur Nutzerverwaltung. Außerdem wurden einzelne Punkte angepasst: Die geplante Mitarbeiter-Rolle wurde durch eine Admin-Rolle ersetzt, Rezepte werden direkt veröffentlicht, Kategorien werden über Dropdown-Menüs ausgewählt und Bewertungen bestehen nun aus Sternen in Kombination mit Text. Das grundlegende Design sowie der Aufbau der Anwendung sind weitgehend wie in der ursprünglichen Planung geblieben.

b) Implementierung der UseCases:

Rezeptanzeige: Rezeptübersicht und -details anzeigen

Die Rezeptübersicht wird über den Endpoint GET /api/recipes geladen und im Frontend als Kartenliste dargestellt. Klickt ein Nutzer auf ein Rezept, wird er auf die Detailseite weitergeleitet, wo die vollständigen Informationen (Zutaten, Schritte, Dauer, Bewertungen) angezeigt werden. Die Detaildaten werden über GET /api/recipes/{id} aus dem Backend geladen. So können Nutzer bequem zwischen Übersicht und Einzelansicht wechseln.

Rezeptverwaltung: Rezept erstellen, bearbeiten, löschen
Im Frontend gibt es zwei Seiten: einmal „Neues Rezept erstellen“ und einmal „Rezept bearbeiten“. Dort füllt man ein Formular aus (Name, Kategorien, Minuten, Portionen, Zutaten, Schritte, optional Bild). Wenn man auf Speichern klickt, prüft das Frontend erst, ob alles passt (z.B. Titel nicht leer, mindestens eine Zutat usw.) und schickt dann die Daten ans Backend: POST zum Erstellen und PUT zum Bearbeiten. Beim Bearbeiten/Löschen schaut das Backend zusätzlich, ob man das überhaupt darf (entweder Admin oder man hat das Rezept selbst erstellt) – sonst kommt 403 (keine Berechtigung). Löschen passiert über den Löschen-Button und sendet einen DELETE ans Backend, dann wird das Rezept aus der DB entfernt.

Wochenplanung & Einkaufsliste:
Der Nutzer kann Rezepte einzelnen Wochentagen. Dafür gibt es im Backend den Endpoint /api/mealplan, über den neue Einträge gespeichert werden. Wenn der Nutzer ein Rezept hinzufügt, wird es zusammen mit dem gewählten Wochentag in der Datenbank gespeichert. Der Wochenplan kann jederzeit angepasst werden: Einträge können bearbeitet (z. B. anderer Tag oder andere Portionszahl) oder komplett gelöscht werden. Diese Änderungen laufen über die entsprechenden PUT- und DELETE-Requests. Aus allen geplanten Rezepten wird automatisch eine Einkaufsliste erstellt. Dafür werden die Zutaten aller Rezepte gesammelt und im Frontend übersichtlich angezeigt. So sieht der Nutzer auf einen Blick, welche Zutaten er für die gesamte Woche benötigt.

Nutzerinteraktionen (Favoriten & Bewertungen):
Angemeldete Nutzer können Rezepte als Favoriten markieren, indem sie auf das Herz-Symbol klicken. Dabei wird im Backend der Endpoint PUT /api/favorites/{productId} aufgerufen, um ein Rezept hinzuzufügen, und DELETE /api/favorites/{productId}, um es wieder zu entfernen. Die Favoriten werden direkt gespeichert und beim nächsten Laden wieder angezeigt.
Zusätzlich können Nutzer Bewertungen abgeben. Dafür wählen sie eine Sternebewertung (1–5 Sterne) und schreiben optional einen kurzen Text. Beim Absenden wird die Bewertung über den Endpoint POST /api/review an das Backend geschickt und gespeichert. Danach werden die Bewertungen automatisch neu geladen und unter dem Rezept angezeigt. So können Nutzer aktiv mit den Rezepten interagieren, ihre Lieblingsrezepte speichern und anderen ihre Meinung mitteilen.

c) Bereitstellung
URL der Anwendung:
Die Webanwendung ist öffentlich erreichbar unter:
https://htwg-in-schneider.github.io/frontend-cooked/

Testzugänge:

- **Admin**
  E-Mail: admin@gmail.com
  Passwort: Cooked12
- **User**
  E-Mail: customer@gmail.com
  Passwort: Cooked12

d) Optimierung
Bei der Optimierung haben wir uns vor allem auf Performance und Nutzererlebnis konzentriert, nicht auf klassisches SEO.

Unsere Bilder liegen überwiegend im .webp-Format (z. B. Bild_homepage.webp, essen1.webp), wodurch die Dateigröße kleiner ist und die Seiten schneller laden.

Im Banner (SpecialBanner.vue) wird das Hauptbild gezielt priorisiert geladen (loading="eager", decoding="async", fetchpriority="high"), damit der wichtigste Inhalt direkt sichtbar ist (Above-the-Fold-Optimierung).

Das Layout wurde responsive angepasst, sodass Banner und Navigation auch auf mobilen Geräten übersichtlich bleiben und kein störendes Overflow entsteht.

Im Backend/Frontend haben wir unnötige Requests vermieden, z. B. keine zusätzlichen Review-API-Aufrufe beim Wochenplan-Drag & Drop.

Außerdem speichern wir den Status der Einkaufsliste persistent in der Datenbank (Shopping-Checks), statt ihn nur im Frontend zu halten – dadurch gibt es weniger UI-Flackern und weniger unnötige API-Calls.