ZLI Abschlussprojekt

A picture containing graphics, font, logo, graphic design

Description automatically generated****

Contents

[Projektantrag 3](#_Toc138941928)

[Konzeption 4](#_Toc138941929)

[Idee 4](#_Toc138941930)

[Use Cases 5](#_Toc138941931)

[Risiken 5](#_Toc138941932)

[Wireframe 5](#_Toc138941933)

[Home Page 5](#_Toc138941934)

[Recipe Single View 6](#_Toc138941935)

[Nutrition Plan 6](#_Toc138941936)

[Klassendiagramm Struktur 8](#_Toc138941937)

[Flow Chart User Access 9](#_Toc138941938)

[Technologien 10](#_Toc138941939)

[Neu 10](#_Toc138941940)

[Arbeitspakete 10](#_Toc138941941)

[Planänderungen 11](#_Toc138941942)

[Testprotokelle 12](#_Toc138941943)

[Backend Endpoint 15](#_Toc138941944)

[Datenbank 17](#_Toc138941945)

[A screenshot of a computer

Description automatically generated with low confidence 17](#_Toc138941946)

[Deployment 18](#_Toc138941947)

[18](#_Toc138941948)

[18](#_Toc138941949)

[Page Beschreibung 18](#_Toc138941950)

[18](#_Toc138941951)

[18](#_Toc138941952)

[Home Page 19](#_Toc138941953)

[My Recipes Page 19](#_Toc138941954)

[Nutrition Plan 20](#_Toc138941955)

[Bookmarks 20](#_Toc138941956)

[Contact 20](#_Toc138941957)

[Login 20](#_Toc138941958)

[Inbetriebnahme 21](#_Toc138941959)

Projektantrag

|  |  |  |
| --- | --- | --- |
| Projekttitel | | |
| RootCook | | |
| **Projektleiter (Name, E-Mail)** | **Coach (Name, E-Mail)** | |
| Linus Daniels, linustimo.daniels@gmail.com | Fabian Dennler, fabian.dennler@zli.ch | |
| **Projektstart** | **Projektende** | |
| 07.06.2023 | 05.07.2023 | |
| **Idee / Kurzbeschreibung** (Um was geht es bei diesem Projekt?) | | |
| Eine Webapp, die auf einem Raspberrypi über Kubernetes gehostet wird. Auf dieser kann man Rezepte erstellen, suchen, löschen usw. | | |
| **Ausgangslage** (Was ist los? Warum wird der Projektauftrag gestellt?) | | |
| Abschlussprojekt ZLI | | |
| **Projektziele** (Erwartete Resultate. Forderungen und Wünsche) | | **Prior.1** |
| Im Internet zugängliche Webapp | | B |
| Rezepte anzeigen, suchen, filtern, erstellen, löschen usw. | | A |
| Rezepte haben genaue Details zu den Nährwerten und haben ein Rating | | A |
| Accounts, mit einem Login System über einen Token | | A |
| Rezepte auf mehrere sprachen | | C |
| Ernährungsplan wobei man Rezepte einfügen kann | | A |
| **Vorgehensziele** (Forderungen und Rahmenbedingungen) | | **Prior.1** |
| Frontendentwicklung mit React.js | | B |
| Container basierte Umgebung (Datenbank, Frontend, Backend) | | A |
| Verwendung von APIs | | A |
| **Lernziele** (Welche Kompetenzen werden angestrebt?) | | **Prior.1** |
| Vertiefung des Verstehens mit Kubernetes. | | B |
| Frontend, Backend und Datenbanken Kompetenzen verbessern. | | A |
| **Ressourcen** (Personal, Sachmittel, Material?) | | **Menge** |
| Raspberry Pi | | 1 |
| **Projektentscheidung** (Freigabeunterschrift) | | **Datum** |
| Freigabe durch: | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1) A = Must have | B = Starkes Wunschziel, wenn irgendwie möglich | C = “Nice to have“ |  |

Konzeption

Idee

Meine Idee war es, eine Webanwendung zu entwickeln, welche es ermöglicht Rezepte anzuzeigen, zu filtern, zu suchen, zu erstellen, zu bearbeiten, zu löschen etc. Jedes dieser Rezepte soll genaue angaben, zu Kalorien, Fette usw. haben. Anhand dieser Angaben, wird ein Rating erstellt, welches aussagt, wie gesund oder ungesund ein Rezept ist. Die Rezepte werden mit Labels, wie z. B. vegan, vegetarisch oder glutenfrei gekennzeichnet. Anhand dieser Rezepte, kann man sich dann einen Ernährungsplan erstellen. Es gäbe eine Ansicht von selbst erstellten Rezepten und eine andere für die anderen Rezepte. Die eigenen Rezepte kann man, falls man will veröffentlichen und sie somit für alle zugänglich machen, ansonsten sind diese nur mit dem Login erreichbar.

Use Cases

Logged in User

* Rezepte erstellen, löschen und bearbeiten
* Rezepte anzeigen, suchen und filtern
* Einen Ernährungsplan erstellen

Not logged in User

* Rezepte anzeigen, suchen und filtern

Risiken

* Zeitdruck
* Umsetzung mit React.js

Wireframe

Home Page

A screenshot of a computer

Description automatically generated with medium confidence

Recipe Single View

A screenshot of a computer

Description automatically generated with medium confidence

Nutrition Plan

A screenshot of a computer screen

Description automatically generated with low confidence

Klassendiagramm Struktur

A picture containing text, diagram, screenshot, plan

Description automatically generated

Flow Chart User Access

A diagram of a website

Description automatically generated with medium confidence

Technologien

* Raspbian 64bit
* Docker
* Kubernetes
* JavaScript / React.js
* HTML
* CSS
* SQL
* APIs?
* Visual Studio Code
* Postman
* MySql Workbench
* Notion
* Word
* Powerpoint
* Canva

Neu

* React.js
* Raspbian 64bit

Arbeitspakete

**Konzeption:**

* Projekt-Idee finden (2h)
* Projektantrag verfassen (2h)
* Eigenständigkeitserklärung ausfüllen (2h)
* Konzeptionierung des Projekts (2h)

**Entwicklung Produkt:**

* Raspberry Pi aufsetzen (2h)
* Datenbank aufsetzten und Tabellen erstellen (4h)
* Frontend-Entwicklung (20h)
  + React.js lernen (4h)
  + Home Page (4h)
  + Recipe view Page (2h)
  + Nutrition Plan Page (4h)
  + My Recipes Page (2h)
  + Bookmarks Page (2h)
  + Login Page (2h)
* Backend-Entwicklung (16h)
  + Token erstellen für Login (2h)
  + Daten aus der Datenbank lesen, schreiben und löschen (4h)
  + Anderes (10h)
* Implementierung von Sicherheitsmaßnahmen (4h)
* Dockerfiles schreiben (4h)
* Deployment auf dem Raspberry Pi (2h)

**Finalisierung:**

* Testprotokoll erstellen & testen (4h)
* Demovideo aufnehmen (2h)
* Demovideo schneiden & erläutern (2h)
* Projektdokumentation abschliessen (4h)

**Präsentations- und Factsheetvorbereitung:**

* Factsheet erstellen (4h)
* Powerpoint Präsentation erstellen (4h

Planänderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Was** | **Wie** | **Begründung** |
| 09.06.2023 | Planänderung | Kubernetes Kluster mit Minikube auf lokalem Gerät und nicht auf dem Raspberry PI | Da ich e nur ein Node, haben werde, kann ich genau so gut auf meinem Gerät arbeiten. |
| 08.06.2023 | Planäderung | Docker / Kubernetes deployment, wird umgesetz, falls gnügend Zeit ist. | React.js zu lernen und umzusetzen geht länger als gedacht. |
| 21.06.2023 | Planäderung | Der Ernährungsplan wird nicht umgesetzt | Rezepte zu editieren ist wichtiger als der Ernährungsplan und das Editieren dauert länger als erwartet. |
| 22.06.2023 | Planäderung | Rezeptratings werden gestrichen | Der Plan war es eine API dafür zu verwenden, ich habe jedoch keine gefunden und selber eine zuschreiben, wäre zu Zeit aufwendig |

Testprotokelle

|  |  |
| --- | --- |
| **Testfall Nr.** | **1** |
| Beschreibung | Registrierung |
| Voraussetzungen | * Der eingegebene Username und E-Mail, dürfen noch nicht in der Datenbank existieren. |
| Testschritte | 1. Gehe zur Registrierung auf der Login Page. 2. Gib einen Usernamen und eine E-Mail an. ( E-Mail muss ein @ enthalten ) 3. Gib bei den zwei Passwort Feldern zweimal dasselbe Passwort ein. 4. Drücke auf Register |
| Erwartetes Ergebnis | Ein alert( «You have successfully registered» ) erscheint und man wird anschliessend zum Login weitergeleitet. |
| Testergebnis | erfüllt |
| Testdatum | 29.06.2023 |
| Bemerkungen | Bei falschen angaben, kommen Fehlermeldungen, wie «Passwords dont match» oder «User already exists». |

|  |  |
| --- | --- |
| **Testfall Nr.** | **2** |
| Beschreibung | Login |
| Voraussetzungen | * Du hast dich erfolgreich registriert |
| Testschritte | 1. Melde dich mit deinem Benutzer, deiner E-Mail und deinem Passwort an. 2. Drücke auf Login |
| Erwartetes Ergebnis | Ein alert( «You have successfully logged in» ) erscheint und man wird anschliessend zur Home Page weitergeleitet. |
| Testergebnis | erfüllt |
| Testdatum | 29.06.2023 |
| Bemerkungen | Bei falschen angeben, erscheint die Fehlermeldung «Error logging in: User doesnt exist». |

|  |  |
| --- | --- |
| **Testfall Nr.** | **3** |
| Beschreibung | Logout |
| Voraussetzungen | * Du hast einen dich erfolgreich registriert * Du hast dich erfolgreich angemeldet |
| Testschritte | 1. Navigiere zur Bookmarks Page, oder eine andre, abgesehen von der Home Page. 2. In der NavBar drücke auf Logout |
| Erwartetes Ergebnis | Ein alert( «You have successfully logged out» ) erscheint und man wird anschliessend zur Home Page weitergeleitet. |
| Testergebnis | erfüllt |
| Testdatum | 30.06.2023 |
| Bemerkungen | Bemerkung, falls du versuchen solltest dich abzumelden, du aber noch gar nicht angemeldet bist, wirdst du diese Meldung trotzdem erhalten. |

|  |  |
| --- | --- |
| **Testfall Nr.** | **4** |
| Beschreibung | Rezept erstellen |
| Voraussetzungen | * Du hast einen dich erfolgreich registriert * Du hast dich erfolgreich angemeldet |
| Testschritte | 1. Navigiere zur MyRecipes Page 2. Drücke auf «Add Recipe» 3. Gib dem Rezept einen Titel, eine Beschreibung, eine bereits gehostete Bildquelle und Anweisungen. 4. Setzte die Sliders, auf belibige Werte 5. Setze « Do you want to publish this Recipe? » auf No. 6. Wähle eine Kategorie und ein Rezepttyp 7. Füge Zutaten deiner wahl hinzu, indem du im Input Feld etwas eintippst, und anschliessend auf das «+» bei einer Zutat drückst, um es hinzuzufügen. gib bei jeder Zutat eine Menge an, indem du auf der Zutat, im Input Feld etwas einträgst. 8. Drücke auf Submit |
| Erwartetes Ergebnis | Ein alert( «You have successfully added a recipe» ) erscheint und man wird anschliessend zur MyRecipe Page weitergeleitet.  Du solltest dort nun dein erstelltes Rezept vorfinden. Auf der Home Page sollte, das Rezept nicht ersichtlich sein, da published auf No gesetzt wurde. |
| Testergebnis | erfüllt |
| Testdatum | 30.06.2023 |
| Bemerkungen | Hinzugefügte rezepte, sind auf der rechten Seite ersichtlich, und können mit «-» entfernt werden.  «Do you want to publish this Recipe? » wird auf No gesetzt sein, wenn du auf die Page kommst, man muss es jedoch manuell nochmals setzten, damit der Wert genommen wird. |

|  |  |
| --- | --- |
| **Testfall Nr.** | **1** |
| Beschreibung |  |
| Voraussetzungen |  |
| Testschritte |  |
| Erwartetes Ergebnis |  |
| Testergebnis |  |
| Testdatum |  |
| Bemerkungen |  |

Backend Endpoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Useage** | **Access** | **Parameters / query etc.** | **Method** |
| **/register** | Generiert einen Benutzer in der Datenbank. | Alle | * Username * Email * Passwort * Passwort Bestätigung | POST |
| **/Login** | Generiert einen Access und Refresh Token in der Datenbank + Local Storage oder Cookie. | Alle | * Username * Email * Passwort | POST |
| **/refresh** | Generiert anhand des **refreshTokens** einen neuen AccessToken. | Alle | * refreshToken | GET |
| **/logout** | Löscht den **refreshToken** | Alle | * refreshToken | GET |
| **/recipe** | Gibt alle Rezepte und deren meal\_type zurück. (Falls **UserId**  übergeben wird, werden nur die Rezepte dieses Users zurück gegen, sonst nur die publizierten) | Alle | * UserId | GET |
| **/recipe** | Erstellt Rezepte und wenn nötig Zutaten. | Nur eingeloggte Benutzer | * filterIngredients * recipeData | POST |
| **/recipe** | Updated Rezepte und falls nötig erstellt Zutaten. | Nur eingeloggte Benutzer | * filterIngredients * recipeData | PUT |
| **/recipe** | Löscht ein Rezept | Nur eingeloggte Benutzer | * recipeId | DELETE |
| **/user** | Gibt den User mit dem entsprechenden **refreshToken** zurück | Nur eingeloggte Benutzer | * refreshToken | GET |
| **/bookmarks** | Gibt alle mit einem Bookamark gekennzeichneten Rezepten, des Benutzers mit **UserId** zurück. | Nur eingeloggte Benutzer | * UserId | GET |
| **/bookmarks** | Erstellt ein Bookmark für das Rezept mit **RecipeId** und den User mit **UserId**. | Nur eingeloggte Benutzer | * UserId * RecipeId | POST |
| **/bookmarks** | Löscht ein Bookmark für das Rezept mit **RecipeId** und den User mit **UserId**. | Nur eingeloggte Benutzer | * UserId * RecipeId | DELETE |
| **/ingredients** | Gibt die Zutaten, zurück, welche im Rezept mit der **RecipeId** vorhanden sind. | Alle | * RecipeId | GET |
| **/ingredients/all** | Gibt alle Zutaten zurück | Nur eingeloggte Benutzer | keine | GET |
| **/recipehasingredient** | Gibt zurück, wie viel von welcher Zutat im Rezept mit der **RecipeId** ist. | Nur eingeloggte Benutzer | * RecipeId | GET |

A screenshot of a computer

Description automatically generated with low confidenceDatenbank

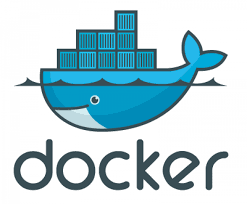
In dieser Datenbank steht das Rezept im Mittelpunkt. Jedes Rezept enthält einen Titel, eine Beschreibung, ein Bild, Anleitungen, verschiedene Nährwertangaben, einen Veröffentlichungsstatus, einen Benutzer, eine Kategorie und einen Rezepttyp. Die Zutaten für jedes Rezept werden in einer separaten Tabelle gespeichert, in der nur der Name der Zutat erfasst wird. Da eine Zutat in mehreren Rezepten vorkommen kann und jedes Rezept mehrere Zutaten haben kann, wird eine Zwischentabelle verwendet.

Jedes Rezept gehört zu einer bestimmten Kategorie wie Vegan, Vegetarisch, Fleisch oder Fisch. Es gibt natürlich noch weitere Kategorien, aber in diesem Fall werden nur diese verwendet. Zusätzlich hat jedes Rezept einen Typ, der das Mahlzeitenkonzept wie Frühstück, Mittagessen, Abendessen oder Dessert beschreibt.

Ein Rezept wird immer von einem bestimmten Benutzer erstellt. Um Lesezeichen zu speichern, wurde eine Zwischentabelle zwischen Benutzer und Rezept erstellt, um festzuhalten, welcher Benutzer bei welchem Rezept ein Lesezeichen gesetzt hat.

Jeder Benutzer in dieser Datenbank hat einen Benutzernamen, eine E-Mail-Adresse und ein Passwort. Die ACCESS\_TOKEN und REFRESH\_TOKEN werden vom Backend zur Authentifizierung verwendet und sind daher ebenfalls in der Datenbank gespeichert, um den Wert in einem Cookie mit dem in der Datenbank zu vergleichen.

Deployment



Ich habe Dockerfiles für mein Frontend und Backend erstellt, die beide auf einem Node-Image basieren. Mein Frontend läuft jedoch auf Nginx. Anfangs habe ich die Images auf meiner GitHub Container Registry (ghcr) hochgeladen. Aufgrund von Authentifizierungsproblemen bei GitHub habe ich jedoch später Docker Hub als Container-Registry verwendet, um meine Images dort zu speichern.



Da ich nur ein Gerät zur Verfügung habe, habe ich mein Kubernetes-Cluster mit Minikube erstellt. Um mein Frontend und Backend zu betreiben, habe ich Dockerfiles erstellt, die beide auf einem Node-Image basieren. Während das Backend direkt auf Node läuft, habe ich für das Frontend den Nginx-Webserver ausgewählt. Ursprünglich habe ich die Docker-Images auf meine GitHub Container Registry (ghcr) hochgeladen. Allerdings stieß ich auf Schwierigkeiten bei der Authentifizierung bei GitHub, weshalb ich später Docker Hub als Container Registry verwendete. Für die Bereitstellung meiner Anwendungen habe ich YAML-Dateien erstellt, die Deployments und Services für das Backend, Frontend, die Datenbank und phpMyAdmin enthalten. Für die Datenbank entschied ich mich für das MySQL-Image. Um das Datenbankpasswort sicher zu speichern, erstellte ich ein Secret. Um einen sicheren externen Zugriff auf meine Anwendungen zu ermöglichen, entschied ich mich für die Verwendung von Ingress anstelle von NodePorts. Hierbei setzte ich den Nginx-Ingress-Controller ein und konfigurierte Routing-Regeln, um die Anfragen automatisch an die richtigen Ziele weiterzuleiten. Anfangs hatte ich einige Schwierigkeiten, von außen auf die Pods zugreifen zu können. Nach einigem Aufwand konnte ich jedoch phpMyAdmin über den Port 81 erreichbar machen, während der Ingress-Controller auf Port 80 lief. Dadurch wurden alle Anfragen an das Frontend weitergeleitet, es sei denn, der Pfad begann mit "/api/". In diesem Fall erfolgte eine Weiterleitung an das Backend.

Obwohl dieser Teil meiner Infrastruktur reibungslos funktionierte, traten noch einige Probleme mit der Datenbank auf. Aufgrund von Zeitmangel hatte ich jedoch keine Möglichkeit, diese Versions- und andere Schwierigkeiten zu beheben.

Page Beschreibung

Home Page

Auf meiner Home Page können sowohl angemeldete Benutzer als auch Gäste alle veröffentlichten Rezepte einsehen. Ich habe die Möglichkeit, die Rezepte nach meinen eigenen Vorlieben zu filtern, zu sortieren und gezielt nach bestimmten Ergebnissen zu suchen.

Um eine optimale Performance und eine reibungslose Benutzererfahrung zu gewährleisten, verarbeite ich die Daten direkt im Frontend. Dank der Verwendung von React, einer leistungsstarken JavaScript-Bibliothek, werden die Komponenten automatisch mit den passenden Informationen aktualisiert und angezeigt.

Als angemeldeter Benutzer wird mein persönlicher Benutzername auf der Home Page prominent präsentiert, um eine individuelle Note zu verleihen und mich willkommen zu heißen.

**Sortierungsmöglichkeiten:**

* Titel ASC / Titel DESC
* Kalorien ASC / Kalorien DESC
* Proteine ASC / Proteine DESC
* Nahrungsfasern ASC / Nahrungsfasern DESC
* Zucker ASC / Zucker DESC
* Fett ASC / Fett DESC

**Filterungsmöglichkeiten:**

* Vegan
* Vegetarisch
* Fleisch
* Fisch
* Kalorien Min / Max
* Proteine Min / Max
* Nahrungsfasern Min / Max
* Zucker Min / Max
* Fett Min / Max

My Recipes Page

A grey and yellow rectangle with white text

Description automatically generated with low confidenceUm alle Funktionen dieser Webseite nutzen zu können, ist eine Registrierung oder Anmeldung erforderlich. Sobald ich angemeldet bin, habe ich Zugriff auf meine persönliche Rezepte-Seite, auf der ich alle von mir erstellten Rezepte einsehen kann. Zusätzlich habe ich die Möglichkeit, diese Rezepte nach meinen Wünschen zu bearbeiten, zu löschen oder sogar neue Rezepte zu erstellen.

Beim Erstellen neuer Rezepte bietet mir die Webseite eine praktische Funktion, um Zutaten auszuwählen, die ich hinzufügen möchte. Sobald ich meine Auswahl treffe, werden diese Zutaten in die Datenbank eingetragen. Sollte es sich um neue, bisher nicht vorhandene Zutaten handeln, werden diese automatisch erstellt und anschließend in der Zwischentabelle korrekt zugeordnet.

Auf diese Weise ermöglicht mir die Webseite, meine Rezeptsammlung bequem zu verwalten und bei Bedarf neue kulinarische Kreationen hinzuzufügen.

Nutrition Plan

Nicht umgesetzt.

Bookmarks

Auf dieser Seite hast du die Möglichkeit, alle Rezepte einzusehen, die du mit einem Lesezeichen versehen hast. Dieses praktische Feature ermöglicht es, interessante Rezepte zu markieren und später schnell wiederzufinden.

Die Funktionalität der Lesezeichen ist eng mit den zugrunde liegenden Rezepten verknüpft. Es gibt jedoch keine separate Speicherung der markierten Rezepte. Stattdessen verweisen die Lesezeichen auf die entsprechenden Rezepte in der Datenbank.

Es ist wichtig zu beachten, dass, falls der Eigentümer eines bestimmten Rezepts dieses löscht, auch alle damit verbundenen Lesezeichen automatisch entfernt werden. Dies geschieht aufgrund der Verwendung des Foreign Keys in der Datenbank, bei dem die ON DELETE CASCADE-Einstellung aktiviert ist.

Diese Vorgehensweise gewährleistet die Integrität der Datenbank, indem verwaiste oder ungültige Verweise vermieden werden.

Contact

Nicht umgesetzt.

Login

Die Login-Seite bietet sowohl die Option, sich einzuloggen als auch sich zu registrieren. Bei der Registrierung werden die Nutzerdaten in die Datenbank eingetragen und sicher gespeichert.

Nach dem erfolgreichen Login erhält man Tokens, die als Authentifizierungsmittel dienen. Diese Tokens müssen bei jeder Anfrage mitgesendet werden, um auf geschützte Ressourcen oder Funktionen zugreifen zu können. Die Tokens sind zeitlich begrenzt und können erneuert werden, um die Sicherheit zu gewährleisten und unbefugten Zugriff zu verhindern.

Technologien

A picture containing text, screenshot, logo, font

Description automatically generated

Ich habe mich für ein React Frontend entschieden, da ich dies schon länger mal machen wollte und es eine der verbreitetsten Java Script Libarys überhaupt ist.

Inbetriebnahme

**Clone den Quell Code auf dein lokales Gerät:**

git clone <https://github.com/LinusTimoDaniels/Abschlussprojekt_ZLI.git>

Führe anschliessend die Befehle unter Command, im vorgesehenen Pfad aus:

|  |  |
| --- | --- |
| **Pfad** | **Command** |
| /RootCook/rootcook-backend | npm install / npm run dev |
| /RootCook/rootcook-frontend | npm install / npm run start |

Anschliessend kann man unter localhost:3000 das Frontend und unter localhost:8080 das Backend erreichen. Es muss allerdings noch eine mysql DB-Server mit folgender Datenbank installiert werden: /Datenbank/db.sql

MariaDB:  
<https://mariadb.org/download/?t=mariadb&p=mariadb&r=11.2.0&os=windows&cpu=x86_64&pkg=msi&m=mva>