

## Aufgabe

Es soll eine Web-Applikation entstehen, die eine Rezepte-Seite (ähnlich Chefkoch, etc.) darstellt. Die genauen Anforderungen an das Projekt werden im folgenden Dokument beschrieben. Das Projekt muss aus einem Angular Frontend und Node + Express Backend bestehen.

### Abgabe

Bitte ladet euer Projekt als Zip-Datei auf Moodle hoch. In dieser Zip-Datei müssen alle notwendigen Dateien vorhanden sein. Den Namen eurer Zip-Datei könnt ihr beliebig wählen.

Achtet darauf, der Zip-Datei weder `node_modules` noch `dist` Ordner hinzuzufügen. Diese min. 3 Ordner müssen vor dem Zippen entfernt werden!

Fügt eurer Abgabe außerdem eine `README.md` Datei hinzu, in der die Namen aller Kollaboratoren vermerkt sind. Solltet ihr mir irgendetwas zu eurem Projekt mitteilen wollen, macht das bitte **nicht** via E-Mail oder Teams, sondern ausschließlich in dieser `README.md` Datei. In dieser Datei beschreibt ihr bitte auch alle nicht offensichtlichen Features.

Abgabeschluss ist der **22.12.2024** um **23:59**. Verspätete Abgaben werden sofort mit 0 Punkten bewertet.

### Gruppen

Das Projekt kann entweder eigenständig oder in Zweiergruppen angefertigt werden. Achtet darauf, dass der Arbeitsanteil aller Gruppenmitglieder ungefähr gleich hoch sein muss.

### Technologien

Folgende Technologien **müssen** für die Umsetzung verwendet werden:

- Frontend: TypeScript, Angular 18
- Backend: Node.js, Express
- Datenbank: NeDB, SQLite o.Ä.

Weitere Komponenten dürfen und sollen über npm hinzugefügt werden! Achtet auch darauf, dass euch unter keinen Umständen eine externe Datenbank bereitgestellt wird.

## Applikation (80 Punkte)

Die folgende Anforderungsliste beschreibt Nutzerverhalten, das von eurem System im Back- und Frontend unterstützt werden muss. Wichtig ist, wie gut und umfangreich die Anforderungen umgesetzt wurden, nicht nur ob sie überhaupt umgesetzt wurden. Weitere Funktionalitäten können nach Belieben hinzugefügt werden und wirken sich positiv auf die „Umfang/ Komplexität“ Bewertung aus.

### Anforderungen

- Nutzer können sich registrieren und einloggen (Nutzer = Eingeloggt, Gast = Nicht Eingeloggt)
- Nutzer können neue Rezepte anlegen
- Nutzer können Rezepte kommentieren
- Nutzer können Rezepte bewerten
- Gäste und Nutzer können Rezepte, Kommentare und Bewertungen ansehen
- Gäste und Nutzer können Nutzerprofile einsehen
- Nutzerprofile enthalten mindestens die Rezept- und Kommentarhistorie

- Rezepte enthalten mindestens einen Titel, die Zutatenliste und eine Beschreibung in Form eines Fließtextes

### **Authentifizierung**

Es reicht vollkommen aus, die relevanten APIs und Frontend Interfaces mittels Basic Auth zu sichern. Komplexere Authentifizierungsverfahren können als „extra Feature“ implementiert werden, werden in der Bewertung aber weniger als alle regulären Anforderungen gewichtet.

### **Bilder**

Rezept- und Profilbilder sind nicht nötig.

### **Ausführbarkeit**

Euer Projekt muss mit wenigen Anweisungen ausführbar sein. Euren Projekten werden ausnahmslos keine Datenbanken zur Verfügung gestellt, ihr seid also auf NeDB, SQLite o.Ä. angewiesen.

- Im Frontend: `npm install`, dann `npm run-script build`
- Im Backend: `npm install`, dann `npm run-script start`
- Sollten weitere Schritte unbedingt nötig sein, dokumentiert sie in der `README.md`

## **API (20 Punkte)**

Die Backend-API muss nach REST-standards entworfen werden und sinnvolle Antworten für Erfolge und Misserfolge anbieten. Richardson Maturity Model Level 2 reicht hier aus.

Dokumentiert dafür alle verfügbaren Operationen, deren Parameter und mögliche Antworten in beliebigem textuellen Format. Benennt die Datei „API-DOC“ und platziert sie im Root-Ordner eurer Zip-Datei.

Achtet darauf, dass die API anhand der API-DOC bewertet wird. Fehlt die API-DOC, wird die API mit 0 Punkten bewertet.