

# Indra Van Wynendaele (202289380)

---

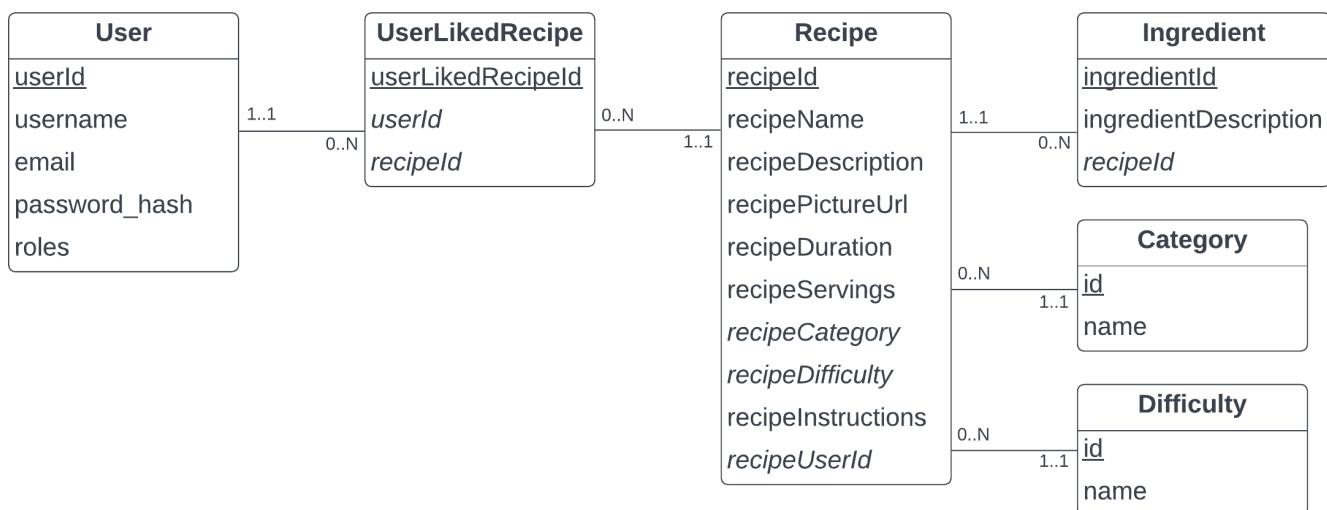
- Front-end Web Development
  - <https://github.com/Web-IV/2324-frontendweb-IndraVanWynendaele>
  - <https://two324-frontendweb-indravanwynendaele.onrender.com>
- Web Services:
  - <https://github.com/Web-IV/2324-webservices-IndraVanWynendaele>
  - <https://two324-webservices-indravanwynendaele.onrender.com>

## Logingegevens Admin

- e-mailadres: [indra.vanwynendaele@gmail.com](mailto:indra.vanwynendaele@gmail.com)
- Wachtwoord: 12345678

## Projectbeschrijving

Mijn project is een verzamelplaats voor recepten. Gebruikers die niet ingelogd zijn, kunnen alle recepten en hun details bekijken. Ingelogde gebruikers kunnen nieuwe recepten uploaden en kunnen ook bestaande recepten van andere gebruikers liken. Hun eigen recepten kunnen ze natuurlijk ook bewerken en verwijderen.



## Screenshots

De home pagina van de website:

Welcome to RecipeVault, your culinary sanctuary for a world of favorite recipes! Dive into a diverse collection of dishes tailored for every taste and expertise. Not only can you explore and save others' cherished recipes, but you can also showcase your culinary prowess by sharing your own. Join us in celebrating the joy of home-cooked goodness - together, let's whip up something truly delicious!

De pagina met verzameling van alle recepten van alle gebruikers:

**Avocado Toast with Eggs and Tomatoes**  
A delicious and nutritious avocado toast topped with perfectly cooked eggs and juicy tomatoes.

**Chicken Teriyaki with Rice and Green Beans**  
Enjoy a savory and sweet dish with tender chicken teriyaki, fluffy rice, and crisp green beans.

Wanneer je een recept wilt liken en niet ingelogd bent:

You need to be logged in!

You can login [here](#).  
Don't have an account yet? You can register [here](#).

**Avocado Toast with Eggs and Tomatoes**  
A delicious and nutritious avocado toast topped with perfectly cooked eggs and juicy tomatoes.

**Chicken Teriyaki with Rice and Green Beans**  
Enjoy a savory and sweet dish with tender chicken teriyaki, fluffy rice, and crisp green beans.

De volledige beschrijving van een recept:



## Stuffed Bell Peppers with Minced Beef and Veggies

Savor the delicious combination of bell peppers filled with seasoned minced beef and colorful veggies.

CATEGORY	DURATION	SERVINGS	DIFFICULTY
Breakfast	30	2	Easy

### Ingredients

- › 5 grams of ingredient\_one

### Instructions

recipe one instructions

Login form:

All Recipes
Add recipe

[Twitter](#)
[My account](#)



### Log in

email:

Password:

Show

Cancel
Log in

Register form:

All Recipes
Add recipe

[Twitter](#)
[My account](#)



### Register

Username:

Email:

Password:

Show

Confirm password:

Show

Cancel
Register

Je eigen gelikete en gemaakte recepten:

All Recipes   Add recipe

**RECIPE Vault**

Liked & created recipes

Search  X 🔍



### Chicken Teriyaki with Rice and Green Beans

Enjoy a savory and sweet dish with tender chicken teriyaki, fluffy rice, and crisp green beans.

[Full recipe](#) X



### Panna Cotta with Crumble

Indulge in the silky smoothness of vanilla panna cotta paired with a delightful crumble topping.

[Full recipe](#) X

Je eigen gelikete en gemaakte recepten:



Enjoy a savory and sweet dish with tender chicken teriyaki, fluffy rice, and crisp green beans.

[Full recipe](#) X



Indulge in the silky smoothness of vanilla panna cotta paired with a delightful crumble topping.

[Full recipe](#) X



### Stuffed Bell Peppers with Minced Beef and Veggies

Savor the delicious combination of bell peppers filled with seasoned minced beef and colorful veggies.

[Full recipe](#) X

Recept bewerken:

All Recipes   Add recipe

**RECIPE Vault**

Edit recipe

Name:

Description:

PictureUrl :

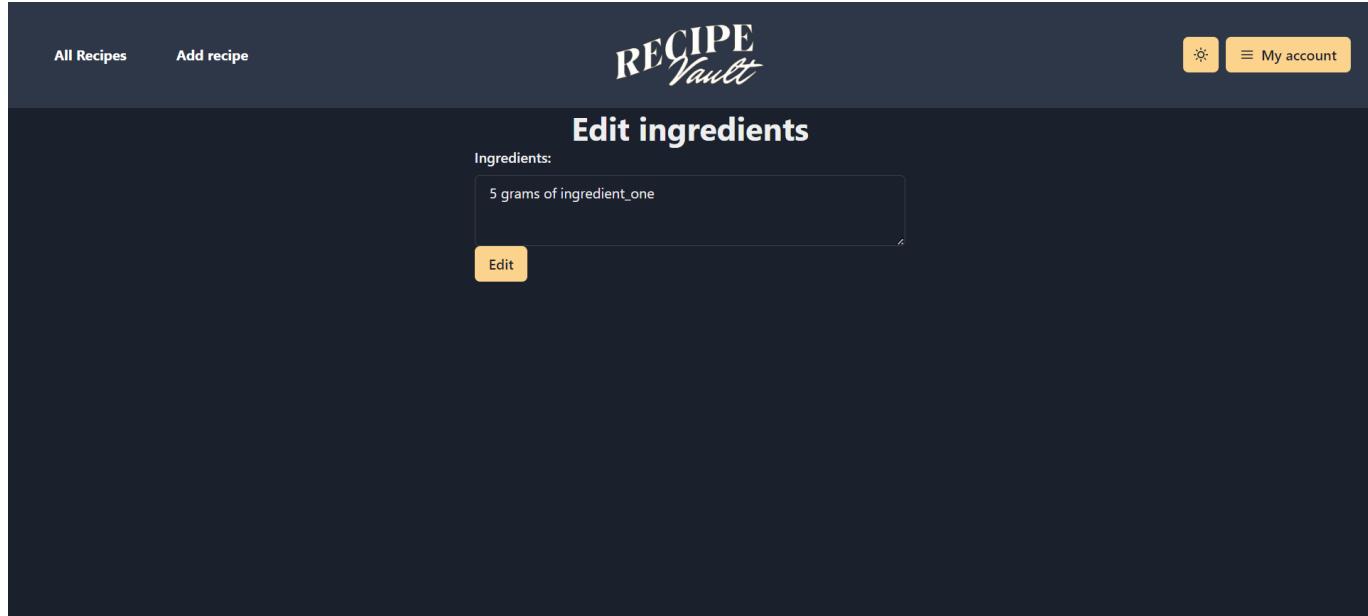
Duration in minutes:

Servings:

Category:

Difficulty:

Ingrediënten bij een recept bewerken:



## API calls

<http://localhost:9000/swagger>

## Behaalde minimumvereisten

Front-end Web Development

- **componenten**

- heeft meerdere componenten - dom & slim (naast login/register)
- applicatie is voldoende complex
- definieert constanten (variabelen, functies en componenten) buiten de component
- minstens één form met meerdere velden met validatie (naast login/register)
- login systeem

- **routing**

- heeft minstens 2 pagina's (naast login/register)
- routes worden afgeschermd met authenticatie en autorisatie

- **state-management**

- meerdere API calls (naast login/register)
- degelijke foutmeldingen indien API-call faalt
- gebruikt useState enkel voor lokale state
- gebruikt gepast state management voor globale state - indien van toepassing

- **hooks**

- gebruikt de hooks op de juiste manier

- **varia**

- een aantal niet-triviale e2e testen
- minstens één extra technologie
- maakt gebruik van de laatste ES-features (async/await, object destructuring, spread operator...)
- duidelijke en volledige README.md
- volledig en tijdig ingediend dossier en voldoende commits

## Web Services

- **datalaag**

- voldoende complex (meer dan één tabel, 2 een-op-veel of veel-op-veel relaties)
- één module beheert de connectie + connectie wordt gesloten bij sluiten server
- heeft migraties - indien van toepassing
- heeft seeds

- **repositorylaag**

- definieert één repository per entiteit (niet voor tussentabellen) - indien van toepassing
- maapt OO-rijke data naar relationele tabellen en vice versa - indien van toepassing

- **servicelaag met een zekere complexiteit**

- bevat alle domeinlogica
- bevat geen SQL-queries of databank-gerelateerde code

- **REST-laag**

- meerdere routes met invoervalidatie
- degelijke foutberichten
- volgt de conventies van een RESTful API
- bevat geen domeinlogica
- geen API calls voor entiteiten die geen zin hebben zonder hun ouder (bvb tussentabellen)
- degelijke autorisatie/authenticatie op alle routes

- **algemeen**

- er is een minimum aan logging voorzien
- een aantal niet-triviale integratietesten (min. 1 controller >=80% coverage)
- minstens één extra technologie
- maakt gebruik van de laatste ES-features (async/await, object destructuring, spread operator...)
- duidelijke en volledige README.md
- volledig en tijdig ingediend dossier en voldoende commits

## Projectstructuur

### Front-end Web Development

De code die nodig is om te linken met mijn API staat in de daarvoor bestemde api map. Mijn authenticatie context en dark- en lightmode context staan beide in de contexts map. De algemene componenten van mijn

website staan in de components map. Elke component heeft zo zijn eigen map met zijn eigen onderdelen. De algemene componenten staan gewoon in de components map. Mijn css-bestanden staan ook per component in de map css. Als laatste de pages map. Hier staan alle paginas van mijn website, ook weer per component in een submap.

## Web Services

De core map bevat de kernfunctionaliteiten van de API zoals bijvoorbeeld de middlewares, authenticatie, de roles, logging, ... . In de data map zorg ik voor een connectie met de databank, dat deze wordt opgevuld en ook mockdata wordt hier voorzien. De repository map voert queries uit op de databank zodat de API calls uitgevoerd kunnen worden. In de service map roep je de functies uit de repository op en doe je een check of alles goed gaat, of je gooit een error. De rest map krijgt de API calls binnen en valideert of de meegegeven informatie wel klopt. Als de validatie oké is, wordt de API call doorgestuurd naar de service laag.

## Extra technologie

### Front-end Web Development

#### *ChakraUI*

ChakraUI biedt UI componenten aan die al styling hebben. Als je zelf nog iets van styling wil toepassen kan dit ook heel erg makkelijk. Bijvoorbeeld de `colorScheme`-prop. Deze gebruik ik heel vaak voor mijn Buttons. Ook heeft ChakraUI een `useColorMode` hook waarmee ik makkelijk de light- en darkmode kan toepassen.

<https://www.npmjs.com/package/@chakra-ui/react>

### Web Services

#### *Swagger*

Swagger Documentatie zorgt voor een duidelijke API. Bij elke functie staat er uitleg over wat deze functie doet, en wat ze gaat returnen. Ik gebruik ook de UI van Swagger waardoor je alle documentatie op de website zelf kan zien. Je kan op de UI ook alle API calls uitvoeren en testen of de verwachte waarde ook gelijk is aan de waarde die de server terugstuurt. Swagger houdt ook rekening met authenticatie en autorisatie bij de API.

<https://www.npmjs.com/package/swagger-jsdoc>

<https://www.npmjs.com/package/koa2-swagger-ui>

## Testresultaten

### Front-end Web Development

In mijn `addRecipes` test controleer ik of het recept correct wordt aangemaakt met de RecipeForm en Ingredientform. Ook test ik of er een error getoond wordt wanneer er een verkeerde input is.

In de `recipes` test controleer ik of de lijst van alle recepten correct getoond wordt. Ook mijn Search field test ik (resultaat, geen resultaat, search field leeg gemaakt). Daarnaast wordt ook getest of er een loader getoond wordt wanneer er een traag antwoord is en een error wanneer de API call faalt. Als laatste test ik of het volledige recept getoond wordt als je op de knop daarvoor klikt.

### Web Services

Voor elke component test is alle API calls. Ik houd rekening met een succesvolle API call, een API call die een error teruggeeft en een API call die een 404 Not Found terug geeft wanneer bijvoorbeeld de categorie van

een recept niet bestaat. Bij de post en put test ik er ook op of alle velden correct ingevuld zijn.

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	95.09	81.52	96.93	95.09	
repository	88.65	75	90.62	88.65	
category.js	82.75	100	66.66	82.75	19-23
difficulty.js	82.75	100	66.66	82.75	19-23
ingredient.js	86.95	66.66	100	86.95	35-37,51-53,64-66
recipe.js	94.03	76.92	100	94.03	109-111,132-134,146-148
user.js	81.81	71.42	83.33	81.81	11-14,37-39,50-52
userLikedRecipe.js	89.09	71.42	100	89.09	36-38,50-52
rest	100	100	100	100	
category.js	100	100	100	100	
difficulty.js	100	100	100	100	
health.js	100	100	100	100	
index.js	100	100	100	100	
ingredient.js	100	100	100	100	
recipe.js	100	100	100	100	
user.js	100	100	100	100	
userLikedRecipe.js	100	100	100	100	
service	83.84	73.62	100	83.84	
_handleDBError.js	34.69	33.33	100	34.69	9-25,29-43
category.js	100	100	100	100	
difficulty.js	100	100	100	100	
health.js	100	100	100	100	
ingredient.js	87.65	73.68	100	87.65	11-12,34-35,44-45,52-53,64-65
recipe.js	87.78	70.37	100	87.78	43-44,48-49,53-54,71-72,79-80,84-85,89-90,107-108
user.js	83.96	70	100	83.96	45-47,54-57,64-68,73-77,95-96,115-116
userLikedRecipe.js	90.76	75	100	90.76	32-33,37-38,48-49

```

Test Suites: 7 passed, 7 total
Tests:     114 passed, 114 total
Snapshots: 0 total
Time:      8.75 s
Ran all test suites.
Done in 10.12s.

```

## Gekende bugs

### Front-end Web Development

- Wanneer je niet ingelogd bent en je wilt een nieuw recept toevoegen moet je 2x inloggen ookal staat na de eerste keer inloggen je userId en token al in localStorage.
- Wanneer de jwt expired is, word je niet automatisch uitgelogd.

### Web Services

/