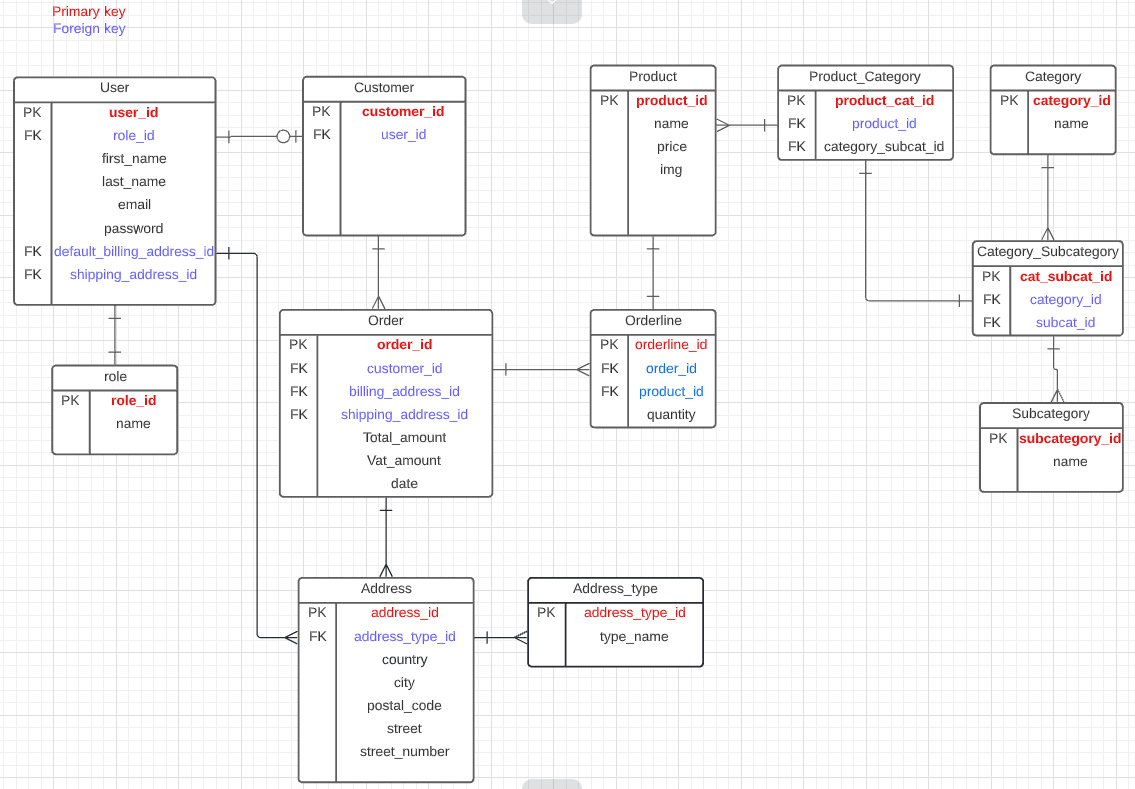
Technische analyse

**Database**

Voor de webshop wordt database aangemaakt met behulp van MariaDB. De databasestructuur ziet er als volgt uit:



* Elke gebruiker kan maar één rol hebben (ofwel ‘admin’ ofwel ‘customer’ ofwel ‘visitor’). ‘Admin’ kan webshop beheren, kan geen bestellingen plaatsen. Een ‘visitor’ is een gebruiker die nog geen bestelling heeft geplaatst, ‘visitor’ kan producten bekijken op de website.
* Een ‘user’ wordt ‘customer’ wanneer hij/zij een bestelling plaatst.
* Een klant kan meerdere orders maken, orders kunnen veel producten bevatten, 1 order kan verschillende orderlijnen bevatten.
* Eén klant heeft 2 adressen(levering & betaling adres), één order kan meerdere adressen hebben, één adres kan van verschillende type zijn.
* Eén product behoort tot één subcategorie.
* Eén categorie kan meerdere producten bevatten.
* Eén categorie kan meerdere subcategorieën bevatten.

Mogelijke routes:

Voor admin gedeelte

login

<http://localhost:8000/admin/login>

<http://localhost:8000/admin> (als je ingelogd bent, zie je het startscherm)

users

<http://localhost:8000/admin/users>

<http://localhost:8000/admin/user/1>

<http://localhost:8000/admin/user/1/edit>

roles

<http://localhost:8000/admin/roles>

<http://localhost:8000/admin/roles/1/edit>

<http://localhost:8000/admin/roles/create>

producten

<http://localhost:8000/admin/products> (je krijgt een lijst met alle producten te zien)

<http://localhost:8000/admin/products/1> (om een product met id 1 te zien)

<http://localhost:8000/admin/product/1/edit>

<http://localhost:8000/admin/product/create>

Noot: db tabellen worden niet via admin aangepast

database tabellen

<http://localhost:8000/admin/database> (om alle tabellen te zien die je database zijn)

<http://localhost:8000/admin/database/create> (om een nieuwe tabel toe te voegen aan database)

<http://localhost:8000/admin/database/1/edit/> (om een bepaalde tabel aan te passen)

visitor omgeving

<http://localhost:8000/home>

<http://localhost:8000/category/1> (wanneer je op één van de categorieën klikt op de menubalk, in het voorbeeld is dat categorie met id 1)

<http://localhost:8000/proudct/1> (wanneer je op een productcard klikt, in dit voorbeeld is het een detailpage van een product met id 1)

<http://localhost:8000/category/1/subcategory/2> (om producten van subcategory 2 uit categorie 1

te zien)

<http://localhost:8000/login>

<http://localhost:8000/register>

<http://localhost:8000/shoppingcart> (winkelmandje)

<http://localhost:8000/checkout>

**Back end**

Backend wordt in laravel uitgevoerd

**Model**

De relaties tussen de tabellen worden in Models bepaald.

Bv:

*class* Customer *extends* Model  
{  
 *public function* orders(){  
 *return* $this->hasMany(Order::*class*);  
 }

(<https://laravel.com/docs/9.x/eloquent-relationships#retrieving-intermediate-table-columns>)

**Controllers**

Data wordt in Json format gestuurd

ProductController

In ProductController worden functies aangemaakt om een lijst van producten te tonen, een product op id te tonen, om een product aan te maken, editeren en verwijderen. Ook filters worden in productController aangemaakt.

$allProducts= Product::all();

$product = Product::where(‘id’, this - > id)-> first();

$productsWithCat = Product::with(‘categories)->get();

CategoryController:

In CategoryController worden functies aangemaakt om alle categorieën te tonen, om categorie op id te tonen, om categorie toe te voegen, aan te passen en te verwijderen. LET OP: categorie moet een subcategorie hebben.

$allCategories = Category::all();

$category = Category::where (‘id’, this->id) -> first();

Of $category = Category::find(this->category\_id);

SubcategoryController

In SubcategoryController worden functies aangemaakt om alle subcategorieën te tonen, om subcategorie op id te tonen, om subcategorie toe te voegen, aan te passen en te verwijderen.

LET OP: subcategorie moet een categorie hebben.

$allSubcategories = Category::all();

OrderController

In orderController worden functies voorzien om alle orders en order op een bepaald id op te halen. Ook functies om een order aan te maken en te verwijderen.

$orders = Order::all();

*public function* getOrder($id)  
{  
 $order = Order::all()->where('order\_id', '=', $id);  
  
 *return* $order;  
}

UserController

In userController worden de functies voorzien om alle gebruikers te tonen en om een gebruiker op id te tonen, een functie om een gebruiker aan te maken, aan te passen en te verwijderen.

$allUsers = User::all();

$user = User::find(this->user\_id);

*public function* getUser($id)  
{  
 $user = User::all()->where('user\_id', '=', $id);  
  
 *return* $user;  
}

authController

In authController worden de functies voorzien voor login en register.

De paswoorden uit input worden geëncrypteerd door Bcrypt hashing voor ze naar database worden gestuurd

Hash::make('password')

**Homepage**

Op de homepage worden de productCards getoond van de 6 nieuwste producten op de website om de laatst toegevoegde items in de kijker te zetten.

$products = Product::orderBy(‘created\_at’, ‘DESC’) -> take(6)->get ();

**Front-end**

Voor front-end wordt Vue.js, vuex, VueRouter en axios geïnstalleerd.

(low priority) Poppervue zal ik gebruiken om aan de gebruiker meer informatie te geven over een button of een inputveld, bv wat in een veld ingevuld moet worden.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**Winkelmandje**

De gebruiker krijgt melding (als pop-up) wanneer hij/zij een product aan zijn winkelmandje toevoegt, ook wanneer hij/zij een product uit winkelmandje verwijdert dat een product verwijderd of toegevoegd is.

Met axios worden de producten uit api gefetched met een GET request. Een voorbeeld van request om een product op te halen:

Axios.get(‘localhost:8000/products/1’).then (response)

Een product wordt toegevoegd door op de ‘Add’-button te klikken. Hiervoor wordt methode ‘addToCart’ gebruikt. Als een product al aanwezig is in het winkelmandje, wordt het aantal producten met 1 verhoogd en gebruiker krijgt een melding dat nog een product werd toegevoegd.

Door op ‘Delete’-knop te drukken wordt het product uit het winkelmandje verwijderd en wordt de totaalprijs en aantal producten in het mandje ook aangepast.

Voor pop-up’s kan [SweetAlert](https://sweetalert.js.org/docs/#configuration) gebruikt worden

**Login-Register**

Wanneer een gebruiker zich registreert, kan hij/zij zich niet als ‘admin’ registreren. De rol van de gebruiker is by default ‘visitor.

In backend wordt authenticationController voorzien met login en register routes

In vuex worden de getters/ actions/ mutations aangemaakt voor login en registratie:

Voor registratie wordt er een formulier voorzien in vue die via een ‘POST’ request data van de gebruiker naar database stuurt. (first\_name, last\_name, email, password)

Axios.post(‘http://localhost:8000/register’, data)

De paswoorden worden geëncrypteerd in backend . Passwoord mag niet als gewone tekst in database komen.

Bij registratie wordt er nagekeken of de useremail al bestaat of niet. De inputvelden van registratieformulier moeten gevalideerd worden. Het kan met VeeValidator (is een form validation library)

# install with npm

npm install vee-validate --save

<https://vee-validate.logaretm.com/v4/>

Na de registratie kom je op de login page waar de gebruiker zijn email en paswoord moet invoeren om in te loggen. Hier wordt een POST request gestuurd met email en paswoord van de gebruiker (Axios.post)

De gebruiker krijgt een token die wordt in headers meegegeven bij login.

Nadat de gebruiker ingelogd is, wordt de naam van die gebruiker getoond (‘Welkom, naam gebruiker) ipv login/register.

**Betaling**

Als extra wil ik Laravel cashier (stripe) integreren om betaling mogelijk te maken

<https://laravel.com/docs/9.x/billing#introduction>

**Admin- customer omgeving**

De webshop bevat een admin, visitor en customer omgeving. Admin gedeelte wordt met behulp van ‘Laravel-admin’ gemaakt. Dit is een admin interface voor Laravel waarmee je data op de webshop kan beheren vanuit admin omgeving.

Met admin-panel kan je gebruikers toevoegen/aanpassen/ verwijderen, rollen toekennen aan de gebruikers en database beheren ( je kan niewe tabellen toevoegen, relaties naar andere tabellen leggen, producten toevoegen)

<https://laravel-admin.org/docs/en> <https://github.com/z-song/laravel-admin>

