

📍 Avenue V. Maistriau 8a
B-7000 Mons

☎ +32 (0)65 33 81 54

📧 scitech-mons@heh.be

WWW.HEH.BE

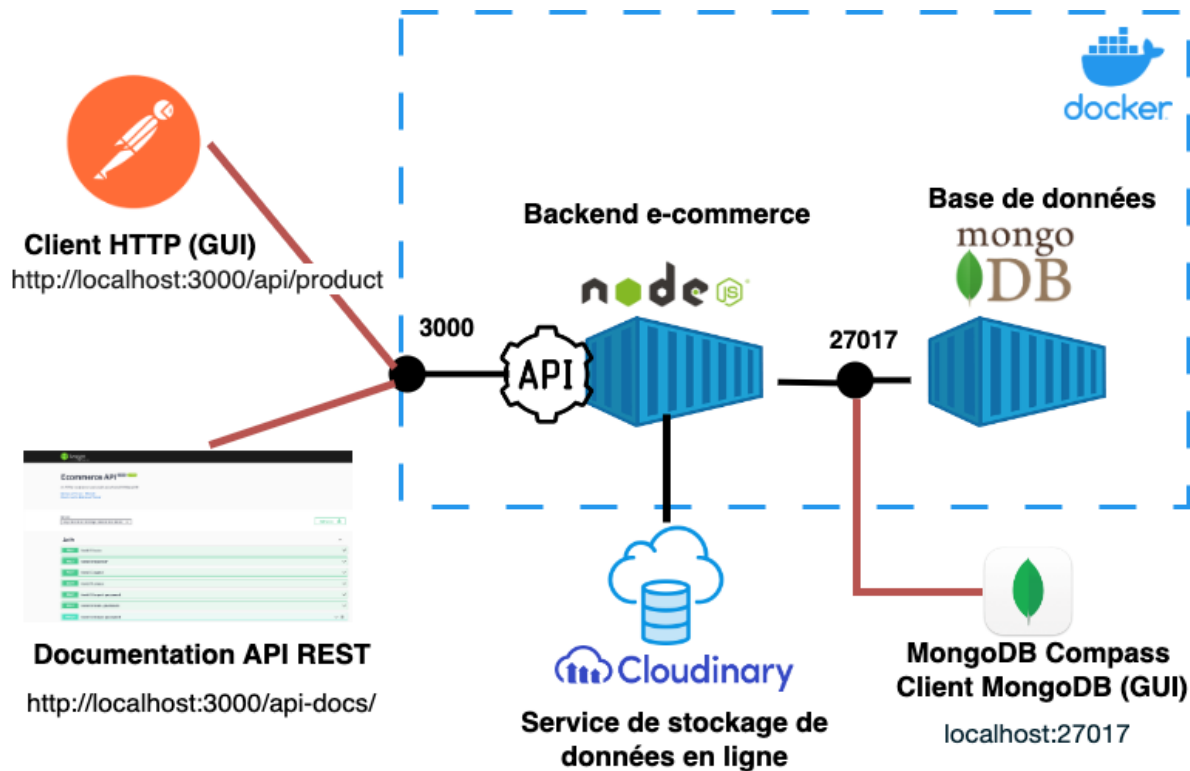
UE : Développement de web services

- **AA** : Développement de web services - Travaux pratiques
Antoine Malaise & Gianni Tricarico

Bachelier en Informatique et Systèmes
Orientation Réseaux et Télécommunications

3. Exercices sur l'utilisation de l'API REST du backend Ecommerce

3.1 Mise en place de l'environnement de travail



Docker est une plateforme permettant de lancer des applications dans des conteneurs logiciels (conteneurisation).

Téléchargez et installez l'application *Docker Desktop* disponible sur le site <https://www.docker.com/get-started/>.

Docker va permettre d'exécuter deux conteneurs :

- Backend e-commerce (développé en NodeJS).
- Base de données (MongoDB).

Conteneur : Backend e-commerce

Le **backend** expose une **API REST** permettant de manipuler les fonctionnalités standards d'un site e-commerce. Le code source a été développé, en Node JS, par monsieur Mahmoud Yasser dont les sources sont disponibles sur GitHub à l'adresse suivante : <https://github.com/Braineaneer/EcommerceAPI>.

Conteneur : Base de données

MongoDB est une base de données NoSQL orientée documents (JSON).

Service tiers : Cloudinary

Le backend utilise un service de stockage en ligne (**Cloudinary**) pour entreposer les images du site (<https://cloudinary.com/>).

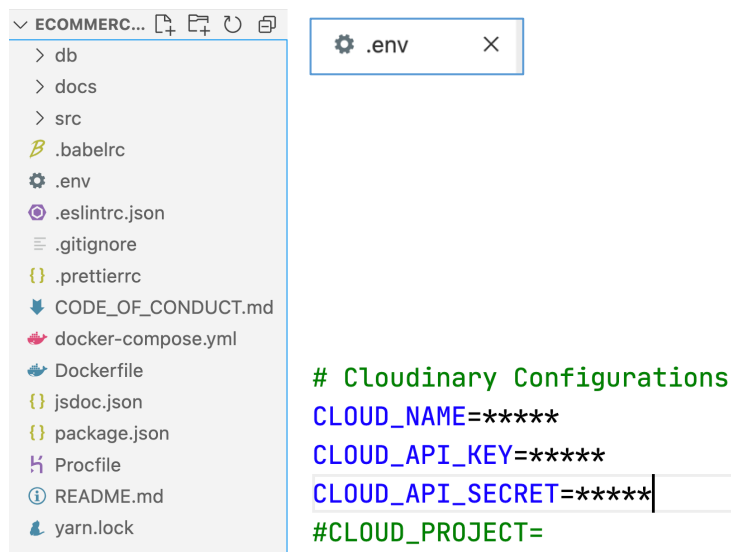
Outil : MongoDB Compass

Cet outil va permettre d'interagir avec la base de données **MonoDB**.

Téléchargez et décompressez le fichier *EcommerceAPI.zip* accessible sur **eCampus**.

Téléchargez et décompressez le fichier *EcommerceAPI.zip* accessible sur **eCampus**.

Voici la structure du code source du backend.



















Créez un compte sur site de **Cloudinary**.

Ensuite, veuillez configurer la section « Mail configuration » et la section « Cloudinary configuration ».

Créez et exécutez les deux conteneurs à l'aide de la commande suivante :

```
SORTIE  TERMINAL  CONSOLE DE DÉBOGAGE  PROBLÈMES
npm does not support Node.js v14.2.0
You should probably upgrade to a newer version of node as we
can't make any promises that npm will work with this version.
You can find the latest version at https://nodejs.org/
(base) giannitricarico@MBP-de-Gianni: ~/Library/CloudStorage/OneDrive-HauteEcoleHainaut/TRI/AC2022-2023/BA2/TP_Web_
Service/EcommerceAPI → docker-compose up
```

À partir de *Docker Desktop*, vous pouvez observer que les deux conteneurs ont bien démarré.

<input type="checkbox"/>	<div><div>ecommerceapi</div></div>	-	Running (2/2)				
<input type="checkbox"/>	<div><div>backend-1 c60e6668093b </div></div>	ecommerceapi:latest	Running	3000:3000 	32 seconds ago 		
<input type="checkbox"/>	<div><div>mongodb-1 36c76ea4359d </div></div>	mongo	Running	27017:27017 	32 seconds ago 		

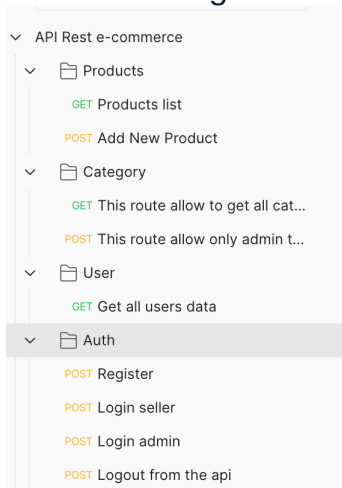
Dès lors, vous pouvez consulter la documentation de l'API REST en tapant dans la barre d'adresse de votre navigateur l'url suivante : <http://localhost:3000/api-docs/>

Vous aurez besoin d'une application permettant de tester l'API REST. Pour ce faire, téléchargez l'outil **Postman** à l'adresse suivante : <https://www.postman.com/>

MongoDB Compass est une interface graphique pour interroger et analyser vos données MongoDB. Téléchargeable sur le site suivant : <https://www.mongodb.com/products/compass>

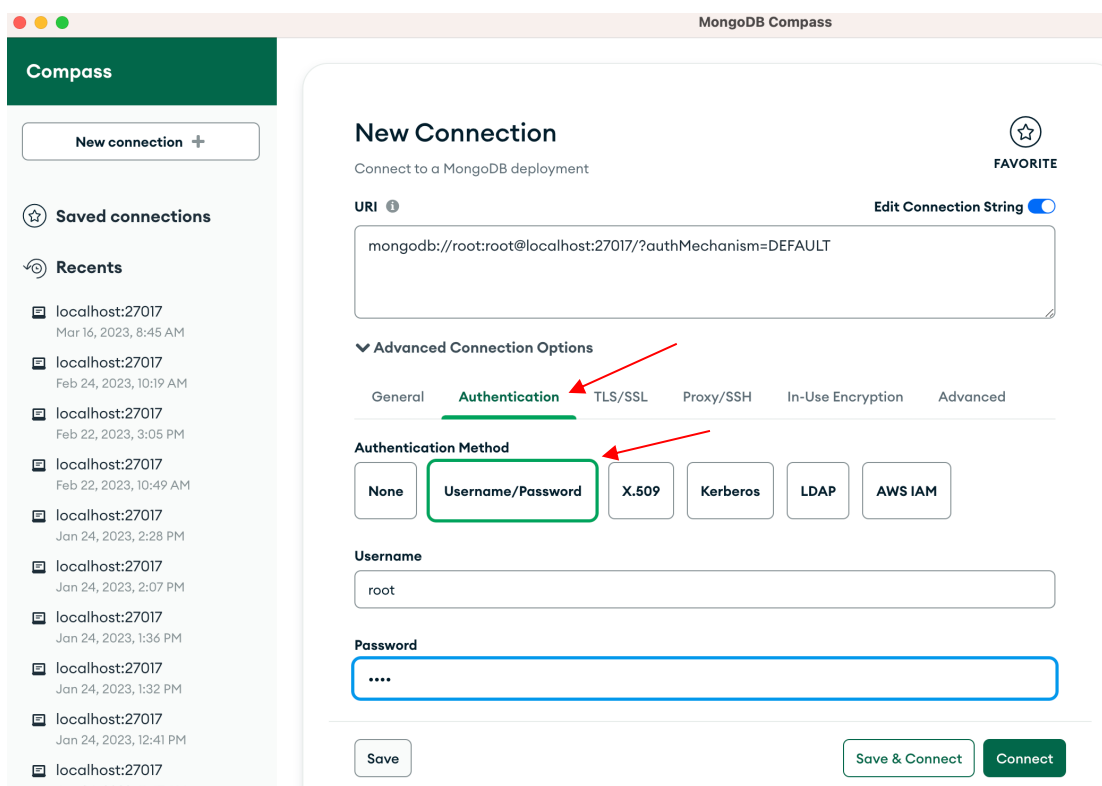
3.2 Consignes

Vous devez organiser vos requêtes à l'aide de dossiers comme illustrés ci-dessous :



Pour l'ensemble des requêtes demandées, vous devez impérativement vous aider de la documentation de l'API REST fournie.

Utilisez **MongoDB Compass** pour vous connecter à la base de données. Pour s'authentifier auprès de la BD, veuillez mentionner le username (root) et le password (root).

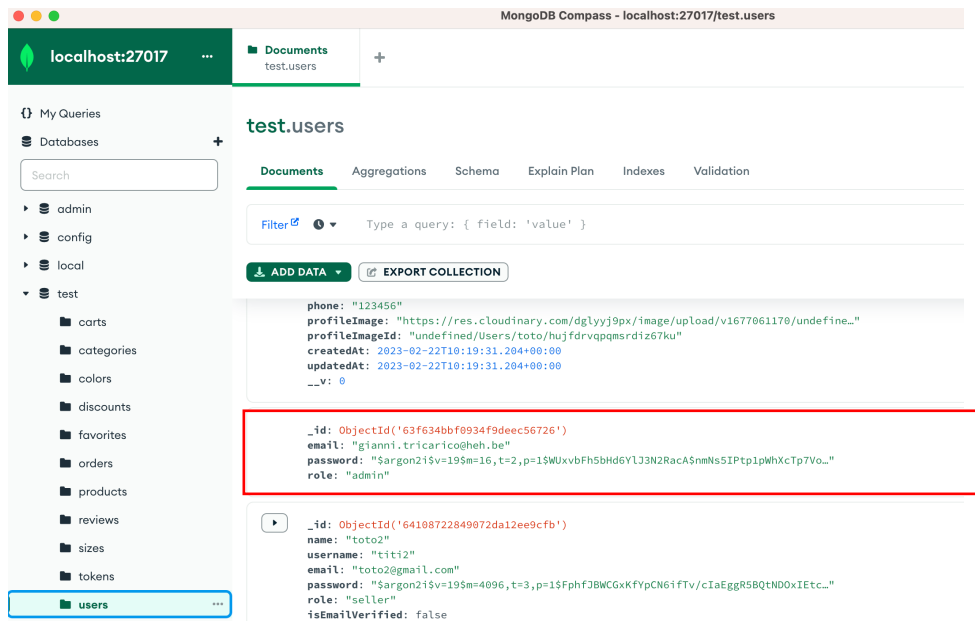


3.3 Créer un compte de type « vendeur »

Créez un compte de type « vendeur » à l'aide de l'API. Dans le body de la requête, vous devez affecter la valeur « seller » à l'attribut « role ».

Pour vérifier que l'utilisateur a bien été créé, réalisez les deux tâches suivantes :

- À l'aide de mongoDB Compass, contrôler la présence de cet utilisateur dans la base de données.
-
- Au moyen de l'API, récupérer la liste des utilisateurs inscrits et examiner qu'il en fait partie.



3.4 Créer un compte de type « admin »

La création du compte « admin » ne peut pas se faire à partir de l'API REST (/auth/register). Vous devez directement insérer le compte « admin » dans la base de données MongoDB.

Pour rappel, dans une base de données, les mots de passe ne doivent pas être enregistrés en clair, mais doivent être stockés dans sa version hachée pour des raisons de sécurité. Pour ce faire, vous allez sur le site suivant : <https://argon2.online/>

Ce site va vous permettre d'hacher votre mot de passe admin.

Argon2 Hash Generator

Plain Text Input

admin#1

Salt: bljRQNTZigMcawMD

Parallelism Factor: 1

Memory Cost: 16

Iterations: 2

Hash Length: 16

Argon2i Argon2d Argon2id

[How to Choose the Right Parameters for Argon2 »](#)

Output in HEX Form [COPY](#)

c84eb1d8035983658de2af55dcfbd58b

Output in Encoded Form [COPY](#)

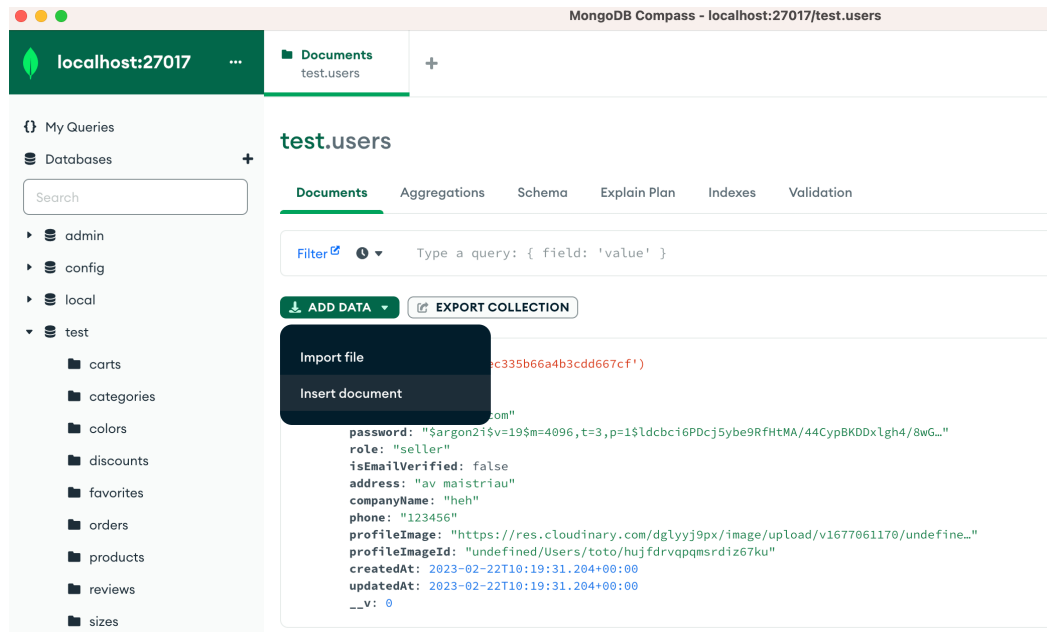
\$argon2i\$v=19\$m=16,t=2,p=1\$YmxqUjF0dFppZ01jYXdNRA\$yE6x2ANZg2WN4q9V3PvVi

W

Mot de passe

Ensuite, vous devez insérer un document JSON dans la collection « users ». Ce document aura la structure suivante :

```
{
  "email": "gianni.tricarico@heh.be",
  "password": "$argon2i$v=19$m=16,t=2,p=1$WUxvbFh5bHd6YlJ3N2RacA$nmNs5IPtp1pWhXcTp7Vo3Q",
  "role": "admin"
}
```



3.5 Se connecter en tant qu'admin

Utilisez Postman pour envoyer une requête au backend afin d'obtenir votre jeton d'accès. Ce dernier sera disponible dans les en-têtes de la réponse.

3.6 Ajouter une catégorie de produit

En utilisant le jeton reçu lors de la connexion précédente, ajoutez une catégorie de produit. Ensuite, listez l'ensemble des catégories afin de vérifier la présence de la catégorie créée.

Pour rappel, l'envoi du token s'effectue par le biais de l'en-tête « Authorization » avec pour valeur « Bearer votreToken »

3.7 Ajouter un produit

A l'aide de Postman, ajoutez un produit dans la catégorie précédemment créée en tant que vendeur (seller).

Par la suite, listez l'ensemble des produits afin de vérifier sa présence.

3.8 Modifier le prix d'un produit

En vous aidant de la documentation, envoyez une requête au backend pour modifier le prix d'un produit.

Ensuite, vérifiez si ce changement a bien été effectué.

3.9 Ajouter un nouveau produit

Ajoutez un nouveau produit dans votre catégorie.

3.10 Récupérer le détail d'un produit

Récupérez le détail d'un produit précis (utiliser son ID).

3.11 Lister l'ensemble des produits avec un filtre sur les champs

Réalisez une requête HTTP pour lister l'ensemble des produits en récupérant uniquement leurs champs « name » et « price ».

3.12 Supprimer un produit

Effectuez une requête pour supprimer un produit.

4. Modalités du projet Ecommerce

Le but de ce projet consiste à réaliser le front-end (l'interface graphique) d'un site web Ecommerce à l'aide des concepts de React vus au cours théorique. Pour cette boutique en ligne, vous avez libre choix sur les produits.

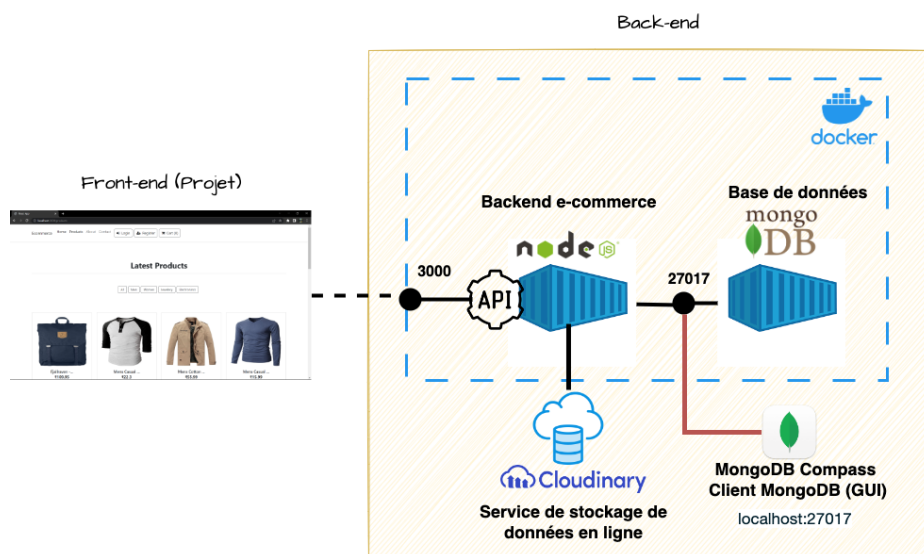
Ce front-end va interagir avec le back-end et la base de données tous deux utilisés lors de l'exercice « **3.4 Tester une API Rest** ».

Afin de faciliter la mise en forme de vos pages et de les rendre « responsives », vous devez intégrer le framework « **React Bootstrap** ». Le site est accessible à l'url suivante : <https://react-bootstrap.github.io/>

L'évaluation orale portera sur les deux points suivants :

- Présentation du projet (20%)
- Défense du projet – Questions/Réponses (80%)

4.1 Architecture de l'application



1. Tâches à réalisées

Tâche 1 : Découpage et implémentation de la page « Products » en composants React

Compétences ciblées: Créer des composants React, utilisation de la fonction *map()* et utilisation des props



Le générateur d'images factices est disponible à l'adresse suivante :

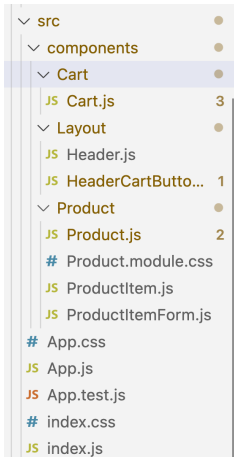
<https://dummyimage.com/>

Dans un premier temps, les produits seront stockés en dur dans un tableau.

Les composants « React Bootstrap » utilisés :

Composant React	Utilise le composant React Bootstrap
Header	NavBar
HeaderCarButton	Badge
Products	Grid cards
ProcdutItem	Cards

Structure de l'arborescence du projet :

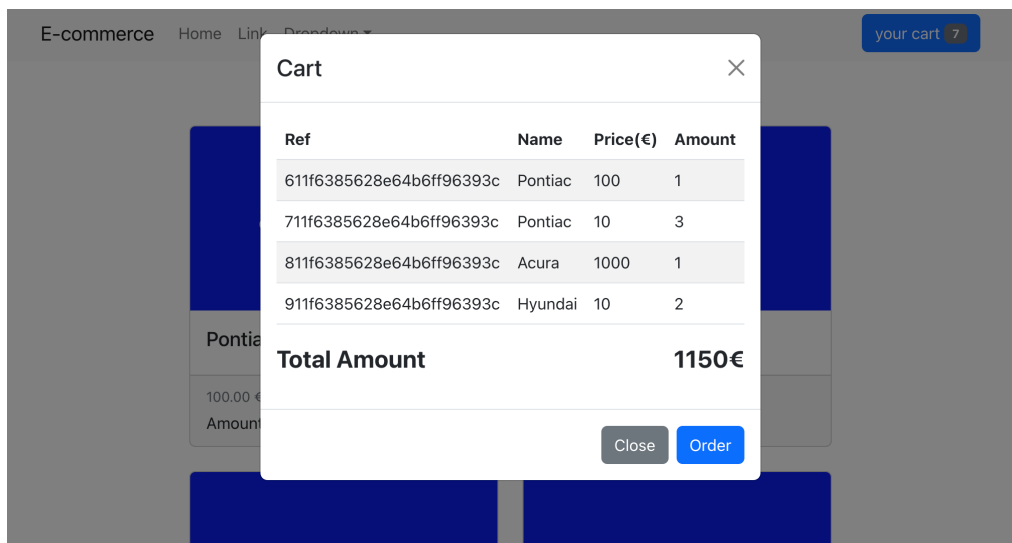


Tâche 2 : Gestion de l'affichage du panier

Compétences ciblées : Affichage conditionnel, adapter le composant « Modal » de React-Bootstrap de type « Static backdrop », gestion d'événement et utilisation du hook *useState()*.

Un clic sur le bouton « your cart » va permettre d'afficher le contenu du panier dans une fenêtre modale.

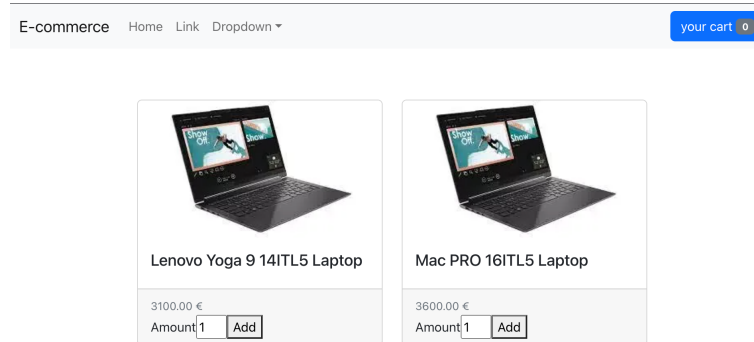
Pour cette tâche, le panier va lister les produits qui sont stockés dans un tableau. La fermeture de cette fenêtre se fera par le biais du bouton « Close ».



Tâche 3 : Afficher les produits provenant du back-end

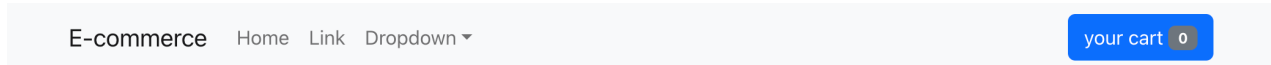
Compétences ciblées : Utilisation de la bibliothèque Axios, interaction avec une API Rest et utilisation du hook `useEffect()`.

La bibliothèque Axios (<https://axios-http.com/docs/intro>) va vous permettre d'envoyer des requêtes HTTP et de recevoir des réponses HTTP. Axios vous permettra de communiquer avec l'API Rest du back-end.



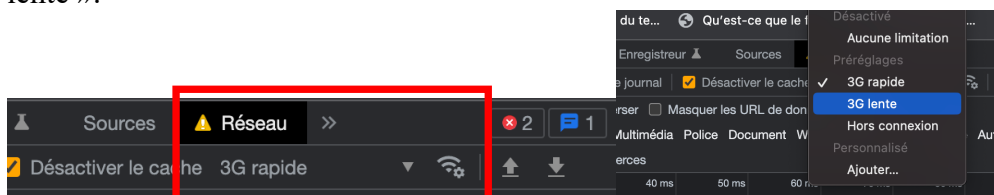
Il faudra prévoir deux situations lors de la récupération des données.

3. **Afficher l'état de chargement des données** : Utiliser une icône de chargement animée (spinner)
4. **Serveur du back-end ne répond plus** : Afficher une alerte avec un message d'erreur.

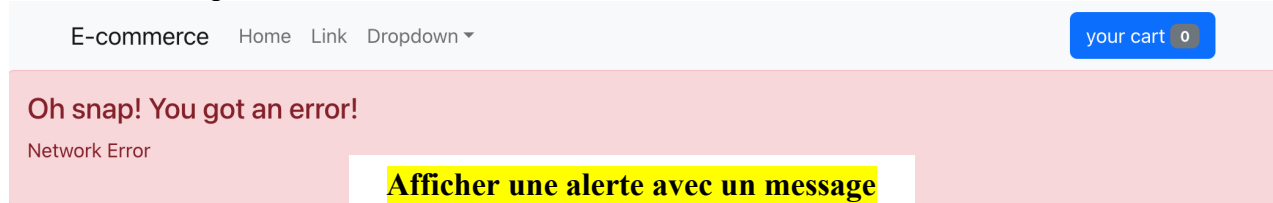


Afficher l'état de chargement des

Pour simuler une connexion lente, vous pouvez vous aider des outils pour les développeurs intégrés à Chrome. Vous devez sélectionner l'onglet « Réseau » et choisir dans le menu « 3G lente ».



Pour simuler la panne du serveur back-end, il vous suffit d'arrêter son conteneur.



Tâche 4 : Gestion d'ajout de produit dans le panier

Compétences ciblées : définition et utilisation d'un contexte, utilisation du hook `useContext()`, définition d'une fonction réductrice et utilisation du hook `useReducer()`.

A form with a text input labeled 'Amount' containing the value '1', and an 'Add' button. Above the input, the text '100.00 €' is displayed.

L'utilisateur définit la quantité d'un produit à l'aide du champ « **Amount** » et l'ajoute au panier en cliquant sur le bouton « **Add** ». Dans un premier temps, le contenu du panier est stocké dans un tableau.

L'affichage du panier se réalise en cliquant sur le bouton « **your cart** ». Le panier doit se rapprocher de la capture d'écran ci-dessous.

The screenshot shows a 'Cart' modal window. It contains a table with the following data:

Ref	Name	Price(€)	Amount
611f6385628e64b6ff96393c	Pontiac	100	2
711f6385628e64b6ff96393c	Pontiac	10	1
811f6385628e64b6ff96393c	Acura	1000	3
Total Amount			3210€

At the bottom of the modal are 'Close' and 'Order' buttons.

Amélioration du panier

Lorsqu'un utilisateur ajoute un article existant, le panier doit uniquement mettre à jour sa quantité (indice : utiliser la méthode `findIndex()`).

Chaque article du panier peut être supprimé via un bouton (indice : utiliser la méthode `filter()`). Ce dernier sera garni d'une icône SVG « Trash/ poubelle ».

Vous pouvez trouver l'icône sur le site : <https://heroicons.com/> ou <https://www.flaticon.com/authors/freepik>

Tâche 5 : Gestion du bouton « your cart »

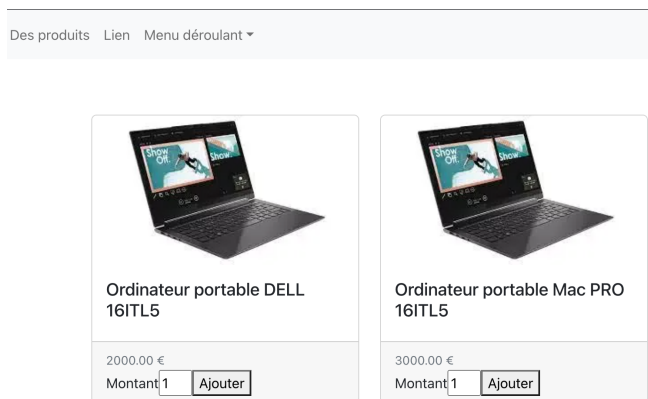
Compétences ciblées : utilisation de la fonction *reduce()*.

Le bouton « your cart » doit afficher le nombre d'articles dans le panier.

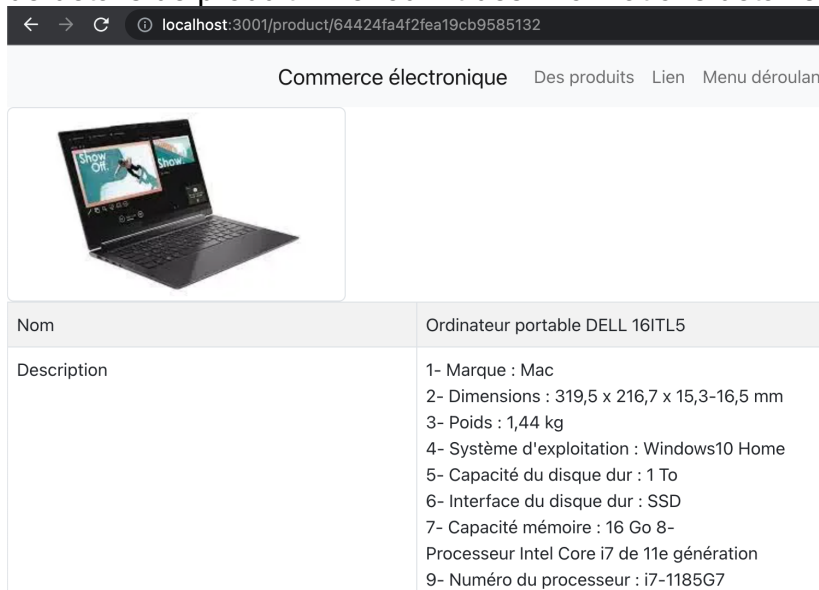
your cart 6

Tâche 6 : Gérer l'affichage de la page détails de produit

Compétences ciblées : utilisation du routage (BrowserRouter, Routes et Route), manipulation de routes dynamiques, création de liens dynamique et navigation dans les pages.

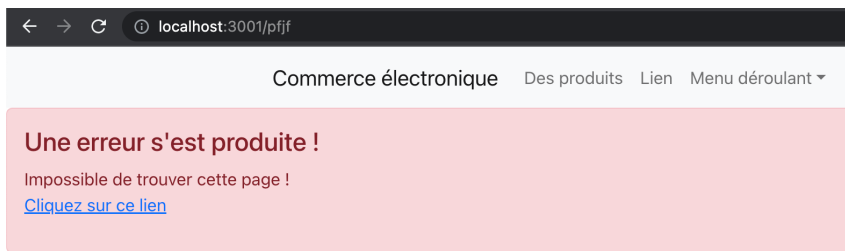


Lorsqu'un internaute clique sur un produit spécifique, ce dernier est dirigé vers la page de détails de produit. Elle fournit des informations détaillées sur le produit sélectionné.



Un clic sur le menu produits (ou accueil) permet de revenir sur la page d'accueil qui présente l'ensemble des produits.

L'internaute peut taper une URL incorrecte dans la barre d'adresse du navigateur. Vous devez gérer ce cas de figure en affichant la page ci-dessous.



Un clic sur le lien «Cliquez sur ce lien » renvoie l'internaute sur la page accueil.