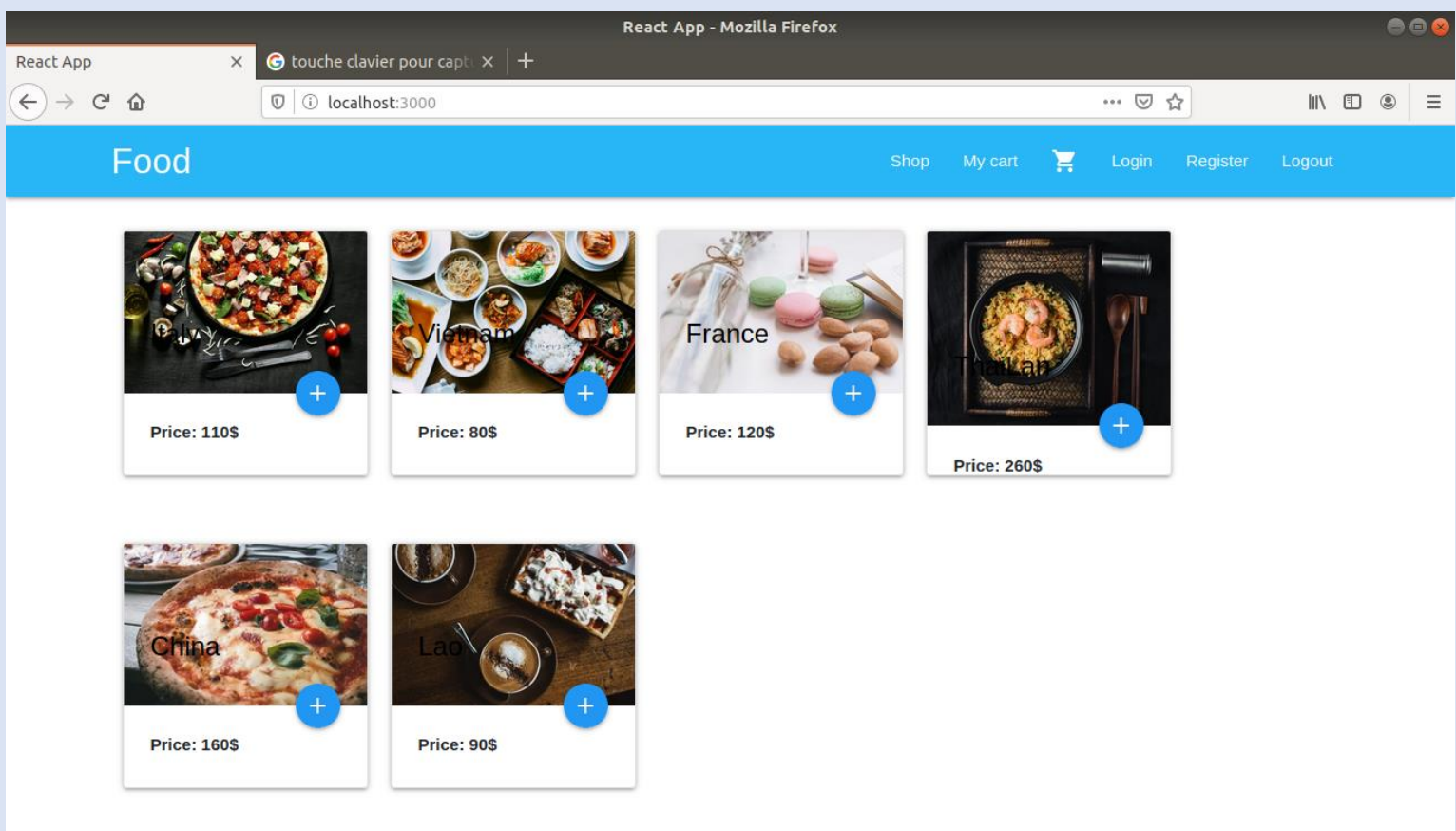




Projet ExpressJS/ReactJS, Gestion de stock

Réaliser par : Samouka DORE
Minh Tuan Vo
Quentin Couturier



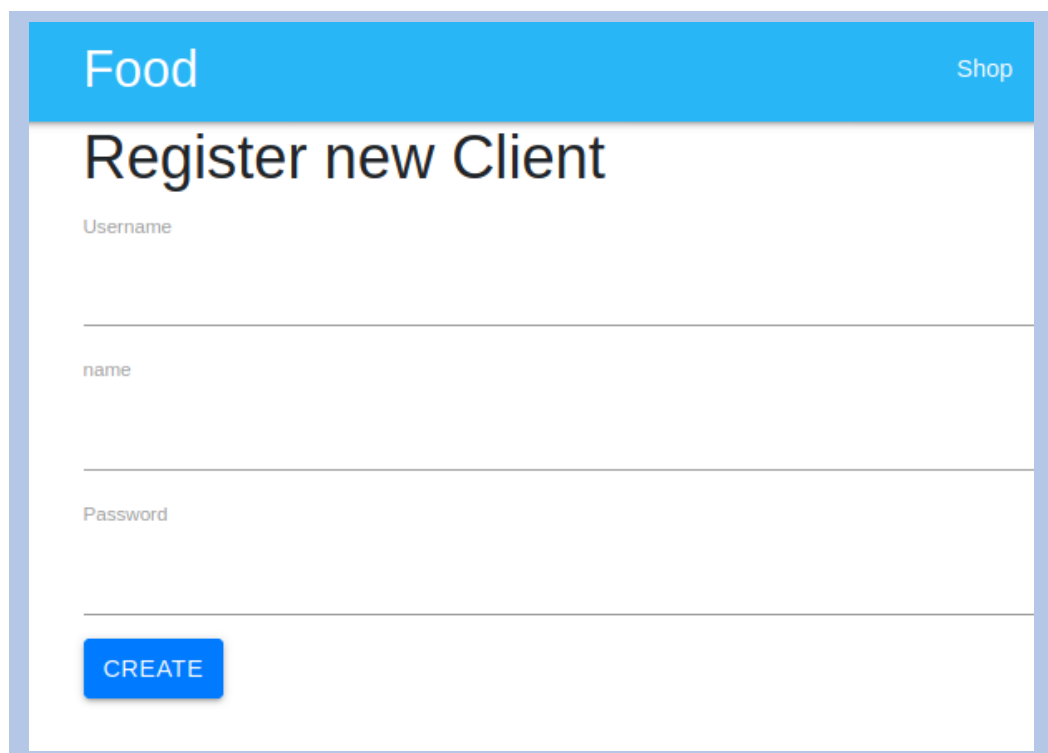
Les fonctionnalités du site et Explication

ReactJS

1. Côté Client

L'image ci-dessus représente l'interface d'utilisateur du site qui met à disposition les produits à vendre sur le site E-Commerce.

Avant de faire des commandes, le client doit créer un compte utilisateur en cliquant sur le lien **Register** et il sera rediriger sur la page suivante :

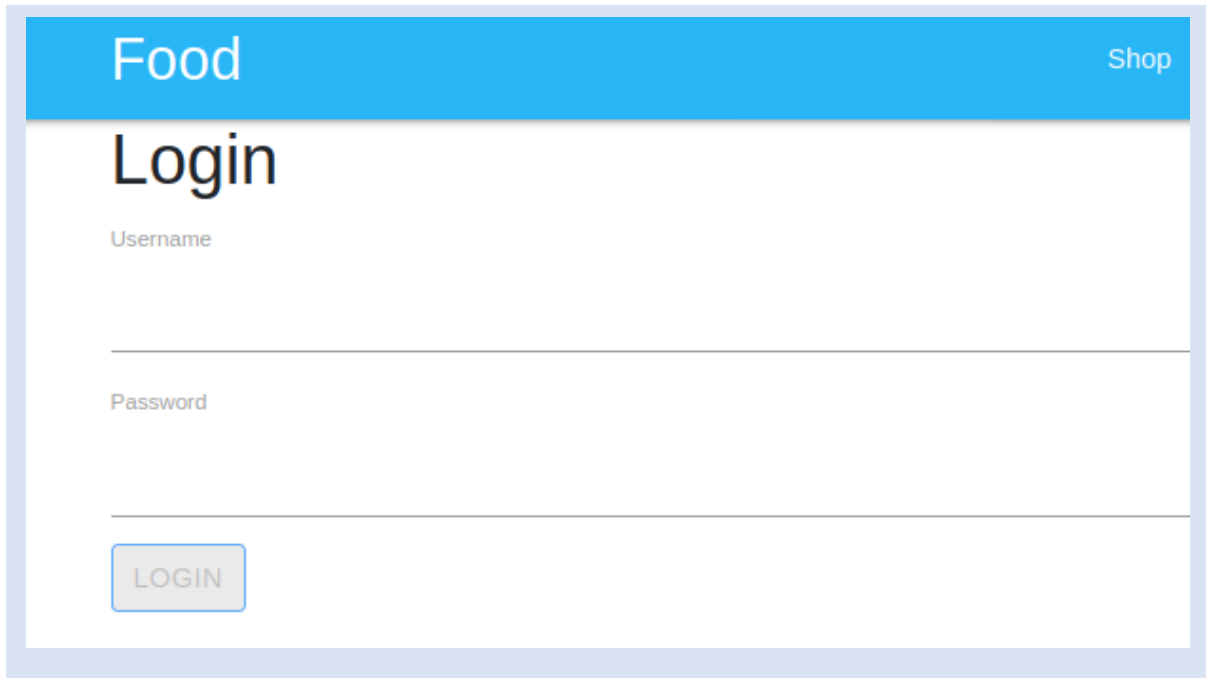


The image shows a web form for registering a new client on a website titled 'Food'. The form has a blue header bar with the word 'Food' on the left and a 'Shop' link on the right. Below the header, the title 'Register new Client' is displayed. The form contains three input fields: 'Username', 'name', and 'Password'. At the bottom of the form is a blue button labeled 'CREATE'.

A partir de laquelle, il peut créer un compte pour pouvoir faire les achats ensuite.

N.B : notons que le Password est crypté (voir détail section sécurité).

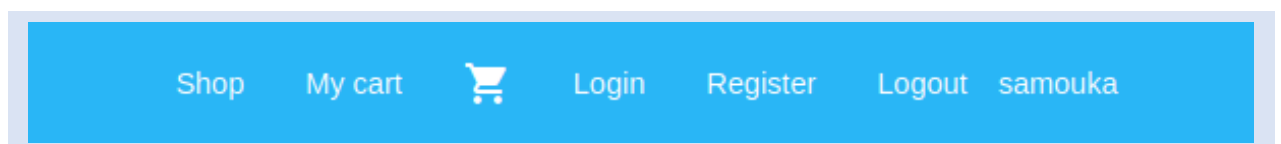
Après d'avoir créé un compte, le client doit se loguer (obligatoire pour faire des achats) en cliquant sur le lien Login, il est redirigé sur la page suivante :



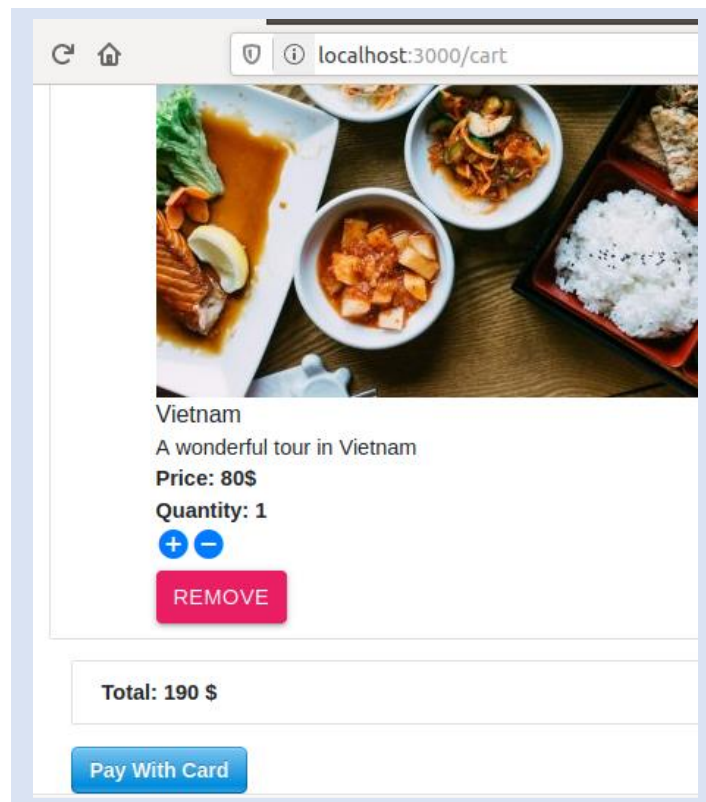
The screenshot shows a web page for a shop named 'Food'. The header is blue with 'Food' in white text on the left and 'Shop' in white text on the right. The main content area is white and has a large 'Login' title. Below the title are two input fields: 'Username' and 'Password'. Below the password field is a blue button with the text 'LOGIN' in white.

N.B : termes utiles, validation du formulaire, Token, autorisation (voir section sécurité).

Ainsi l'utilisateur est directement reconnu par le site et son nom est affiché en haut à droite du NavBar.



Le client peut ainsi faire les achats en sélectionnant les articles qu'il a besoin et mettre dans son panier (My Cart)



En Cliquant sur Button **Pay With Card** une certaine information lui est demandé avant de valider la commande.

This screenshot shows a payment form with the following fields: an email field containing 'client@gmail.com', a checked checkbox for 'Same billing & shipping info', a name field with 'client', an address field with 'street client', a zip code field with '18000', a city field with 'bourges', and a country dropdown menu set to 'France'. At the bottom is a blue button labeled 'Payment Info' with a right-pointing arrow.

This screenshot shows a payment form with the following fields: a card number field containing '4242 4242 4242 4242' with a Visa logo, an expiration date field with '04 / 20', and a CVV field with '123'. Below these is an unchecked checkbox for 'Remember me'. At the bottom is a large blue button labeled 'Pay \$190.00'.

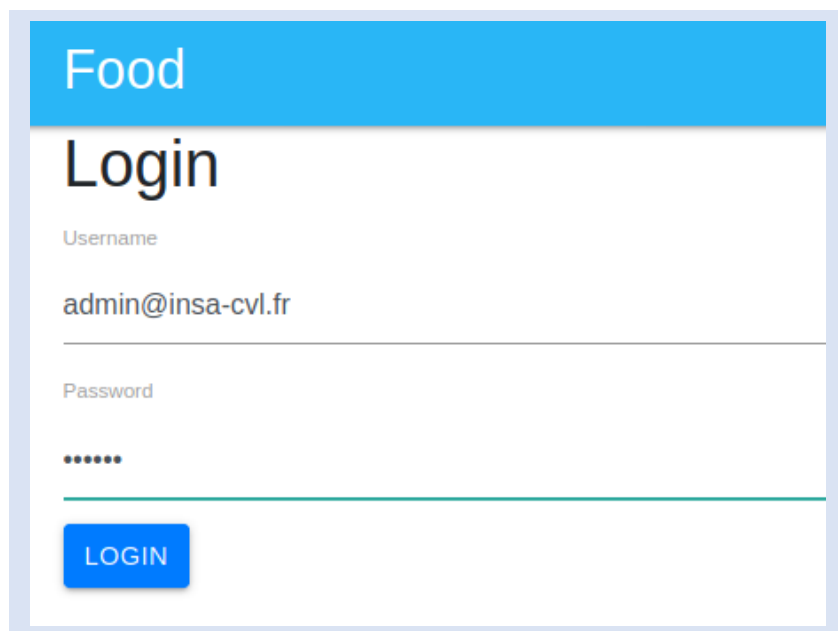
N.B : l'administrateur pourrait ainsi voir les commandes avec un compte administrateur

2. Côté administrateur

Comme le client, Administrateur doit aussi avoir un compte administrateur et il peut créer aussi d'autre compte Administrateur avec son Token. Mais le problème c'est la création du premier compte administrateur sans compromettre la sécurité du site.

Alors pour des raisons de sécurité, avant de se loguer en tant qu'administrateur, il faut exécuter le script **firstAdmin.sh** stocker dans le dossier du Projet, ce script permet de créer de premier compte administrateur pour pouvoir administrer le site après.

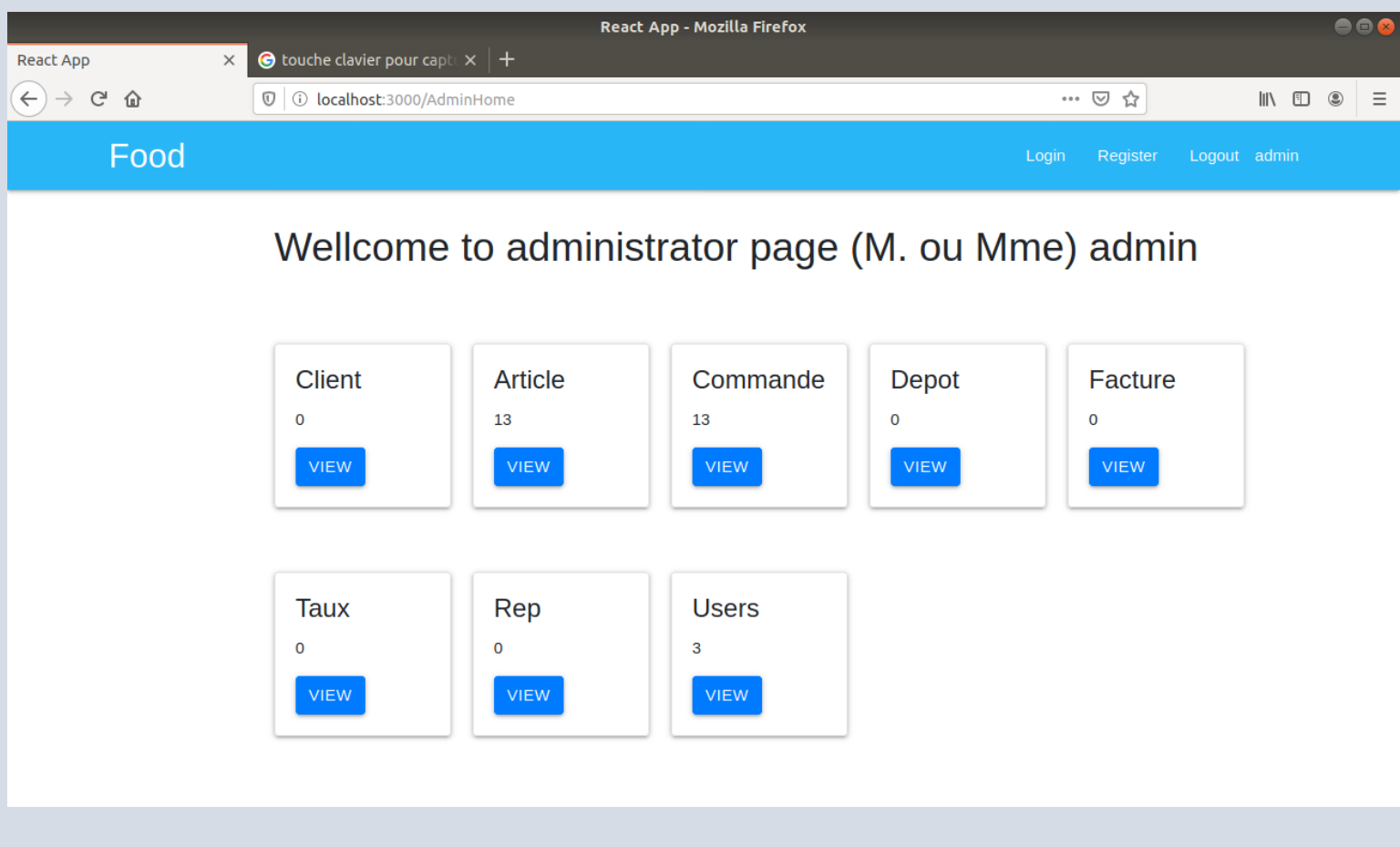
Ainsi on pourrait se loguer avec le
Username : **admin@insa-cvl.fr** et le Password : **azerty**



The image shows a web application interface for logging in. At the top, there is a blue header bar with the word "Food" in white. Below the header, the word "Login" is prominently displayed in a large, black, sans-serif font. Underneath "Login", there are two input fields. The first field is labeled "Username" in a small, grey font and contains the text "admin@insa-cvl.fr". The second field is labeled "Password" in a small, grey font and contains seven dots, indicating a masked password. Below these fields is a blue rectangular button with the word "LOGIN" in white, uppercase letters. The entire login form is enclosed in a light blue border.

N.B : il est identifié sur le site avec un Token administrateur qui est généré

Après l'authentification, Admin est redirigé directement sur la homePage Administrateur



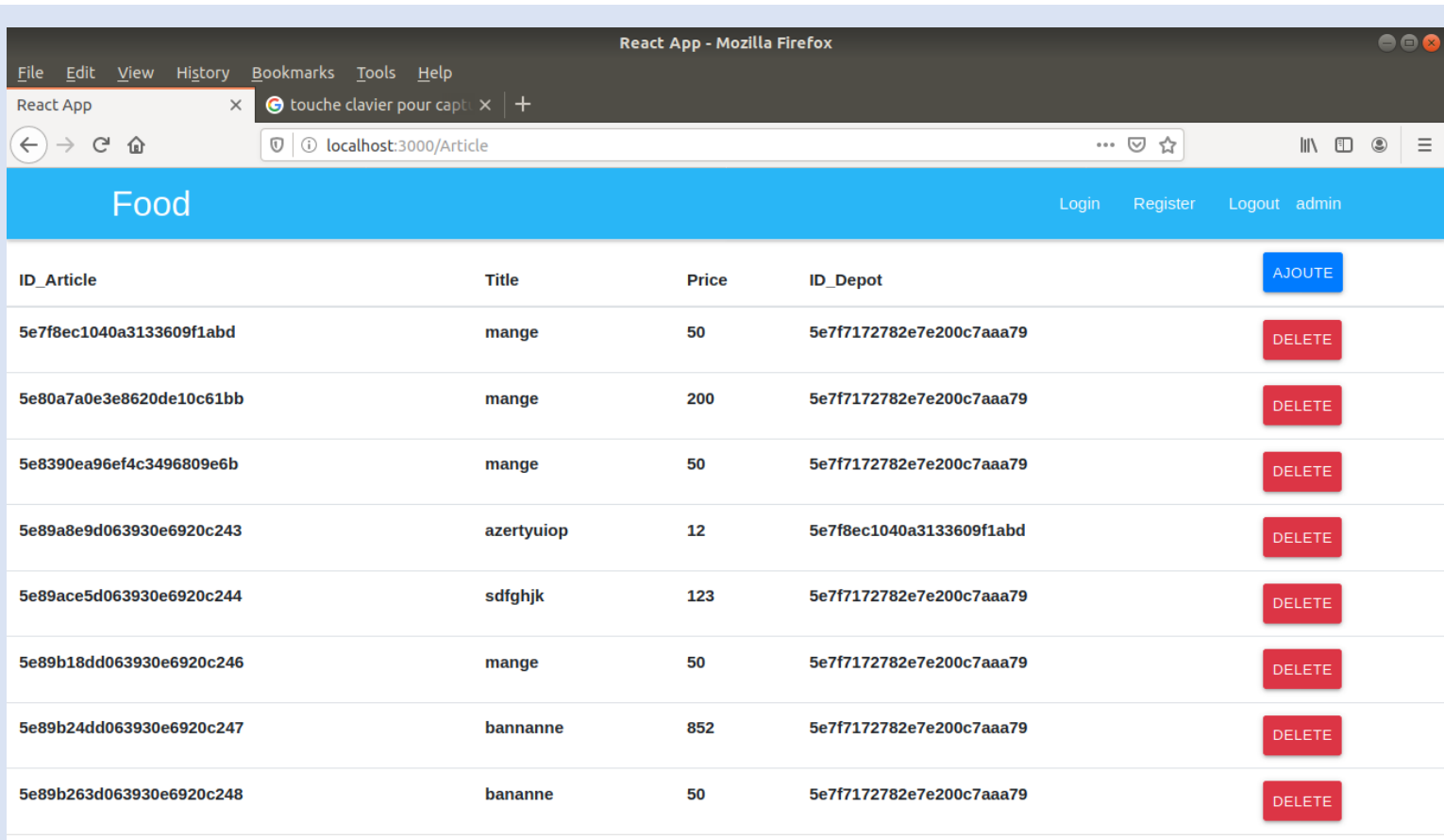
N.B : notons que les liens **shop** et **MyCart** ont disparu car l'administrateur n'a pas besoin de faire des achats avec un compte administrateur par contre il peut toujours voir le homePage avec le lien **Food**

Une fois sur la page principale administrateur, l'administrateur peut créer d'autre compte administrateur en cliquant sur Register

The screenshot shows a web form titled "Food Register new Admin". It contains three input fields for "Username", "name", and "Password", each with a corresponding label above it. At the bottom of the form is a blue button labeled "CREATE".

Il peut également consulter la base de données (Articles, Commande, Users, Client, ...)

Exemple la page Article en cliquant sur VIEW



ID_Article	Title	Price	ID_Depot	
5e7f8ec1040a3133609f1abd	mange	50	5e7f7172782e7e200c7aaa79	AJOUTE
5e80a7a0e3e8620de10c61bb	mange	200	5e7f7172782e7e200c7aaa79	DELETE
5e8390ea96ef4c3496809e6b	mange	50	5e7f7172782e7e200c7aaa79	DELETE
5e89a8e9d063930e6920c243	azertyuiop	12	5e7f8ec1040a3133609f1abd	DELETE
5e89ace5d063930e6920c244	sdfghjk	123	5e7f7172782e7e200c7aaa79	DELETE
5e89b18dd063930e6920c246	mange	50	5e7f7172782e7e200c7aaa79	DELETE
5e89b24dd063930e6920c247	bannanne	852	5e7f7172782e7e200c7aaa79	DELETE
5e89b263d063930e6920c248	bananne	50	5e7f7172782e7e200c7aaa79	DELETE

Sur la page article, on peut supprimer un article ou ajouter d'autre, en cliquant sur ajouter, un modal nous est proposé pour ajouter les informations d'article.



Add a new article

Title

Price

Depot

EX : 5e7f7172782e7e200c7aaa79

SUBMIT CLOSE

N.B : le projet n'est pas totalement fini, il reste des problèmes de synchronisation, donc dans le champ Depot, il faut mettre un identifiant **object.ID** valide manuellement pour pouvoir ajouter un article.

Le front n'est pas fini complètement, certaine partie (component) du front n'ont pas été programmée comme par exemple le component (client ou facture) mais tout le backend est faite (voir la section problème).

3. Sécurité du site

Comme indiquer ci-haut, pour éviter qu'un utilisateur crée un compte administrateur, il faut au préalable se loguer en tant qu'administrateur pour pouvoir créer un compte administrateur car en se loguant en tant administrateur, un Token est générer par le serveur et stocké sur le localStorage du navigateur, ce token contient le booléen « **isAdmin : true** » qui sert à reconnaître un compte administrateur (toute la sécurité du site repose sur ce token).

Dans les champs de formulaire (Register, Login), les données saisies dans le formulaire sont soumises à une procédure de validation (par exemple le Button de validation reste inactif tant que l'un des champs est vide, le Username doit être une adresse email valide et au moins 5 caractères.

Food

Register new Client

Username

sdore

"Username" must be a valid email

name

"name" length must be at least 5 characters long

Password

"Password" length must be at least 5 characters long

CREATE

Food

Login

Username

azerty

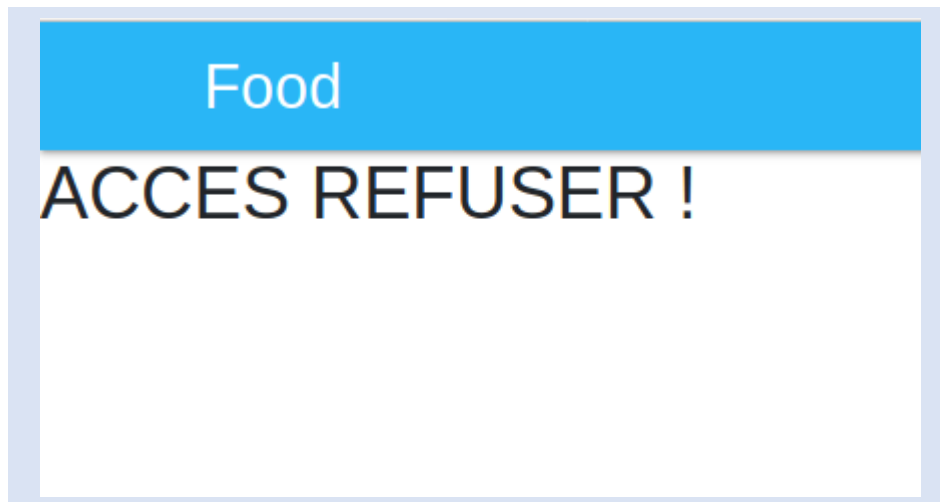
Invalid email or password.

Password

.....

LOGIN

Pour les authaurisations, un utilisateur qui veut accéder illégalement une page



Un utilisateur qui est authentifier en tant que client qui essaie d'accéder illégalement aux pages administrateurs se voit refuser ses accès et s'il n'est pas authentifié, il est redirigé vers la page de connexion.

Les Passwords sont chiffrés avec l'algorithme bcrypte et un salt de longueur 10.

Food					Login	Register	Logout	admin
ID User	Username	Name	Password	IsAdmin				
5e8cf00e1963646602f9274c	samouka@insa	samouka	\$2b\$10\$DpKrd0RC3FctPsiamP/EreQtegE2KXa7Lv83j.ICbMXMIA.YUZV/6	False	DELETE			
5e8cf0811963646602f9274d	admin@insa-cvl.fr	admin	\$2b\$10\$yCLYvJCDroxbMesVaalo6urmQC8LRco5KQPpyFbtVh6Wpc7nhZhgf.	True	DELETE			
5e8dba7fe627450c1ad8aa00	dore@insa	samouka	\$2b\$10\$OsJSTH/sN37I5Niz0Xvne.x238MPE3y/v9eM379/4QsJuTWikwGMi	False	DELETE			

4. Problèmes.

Le coté front du site n'est pas fini encore, il y a des composants qui n'ont pas été programmé, des fonctionnalités initialement prévues mais qui ne sont pas finit êtres programmer, certaine partie et fonctions utilisée ne sont pas optimums et d'autres données sont manuellement ajouter sur la page coté commande.

5. Essentielle sur le backend (ExpressJs, nodeJS)

La modélisation de la base de données a été faite, les routes ont été créé, sur chaque collection il existe la méthode (POST, GET, PUT, DELETE).

Authentification et autorisation sont faite en backend, les passwords sont cryptés et on utilise le json-web-token pour générer les token. On a utilisé le principe de middleware.

La méthode utilisée est le MVC (model, view, controller).

6. Annexe :

Le projet est sur le lien

<https://github.com/sdore2021/SAM-TUAN-QUENTIN-PROJET-WEB>