

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 1
Nom, prénom : LAMONNIER Alexis		N° candidat : 02445945427
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : 28 / 04 / 2025
Organisation support de la réalisation professionnelle La Brasserie Terroir & Savoirs désire moderniser son image par le biais du numérique. Elle souhaite mettre en place un nouveau site web, ainsi qu'un outil pour gérer les ses stocks, sa base client et les réservations de commandes		
Intitulé de la réalisation professionnelle Le premier contexte de ce projet est la partie Web , elle a été réalisé avec les Framework Next (TypeScript) et Symfony pour l'API (PHP) Pour ce projet les fonctionnalités sont : <ul style="list-style-type: none">- Récupération et affichage des produits sur le site vitrine.- Connexion et déconnexion sur l'espace administrateur.- Création, modification et suppression d'un Compte.- Création, lecture, modification et suppression d'un Produit.- Création, modification et suppression d'une Réservation.		
Période de réalisation : 09/04/2025 - 21/05/2025 Lieu : EPSI Lille Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe		
Compétences travaillées <ul style="list-style-type: none"><input checked="" type="checkbox"/> Concevoir et développer une solution applicative<input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative<input checked="" type="checkbox"/> Gérer les données		
Conditions de réalisation ⁵ (ressources fournies, résultats attendus) Pour réaliser le projet, j'ai eu recours à : des cours sur les API REST, un cahier des charges. L'objectif étant de réaliser une Interface web ayant une vitrine, avec un espace administrateur permettant de gérer les comptes, les produits et les réservations qui communique via une API avec une Base de données à distance		
Description des ressources documentaires, matérielles et logicielles utilisées ⁶ IDE : Visual Studio Code Gestionnaire de version : Github BD : MySQL Technologies utilisées : Next / TypeScript , SQL Outils de test : Postman		
Modalités d'accès aux productions ⁷ et à leur documentation ⁸ Lien GitHub : https://github.com/alexkjzz/brasserie Login/mdp : alexis.lamonnier@gmail.com / racine pour l'administration & test@gmail.com / racine pour un utilisateur (bloqué pour ce contexte)		

⁵ En référence aux conditions de réalisation et ressources nécessaires du bloc « Conception et développement d'applications » prévues

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)

dans le référentiel de certification du BTS SIO.

⁶ Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

⁷ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁸ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Sommaire

1-présentation - 3

2-aperçu du visuel - 3 à 8

3-base de données - 8 à 9

4-détails techniques - 9 à 12

5-sources - 12

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)

Épreuve E6 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

1-Présentation

Dans le cadre de la préparation au épreuve E6, j'ai développé une interface web en Next.js (tsx), pour répondre au cahier des charges de notre client.

J'ai réalisé un site vitrine ayant une gestion admin. Il est possible de :

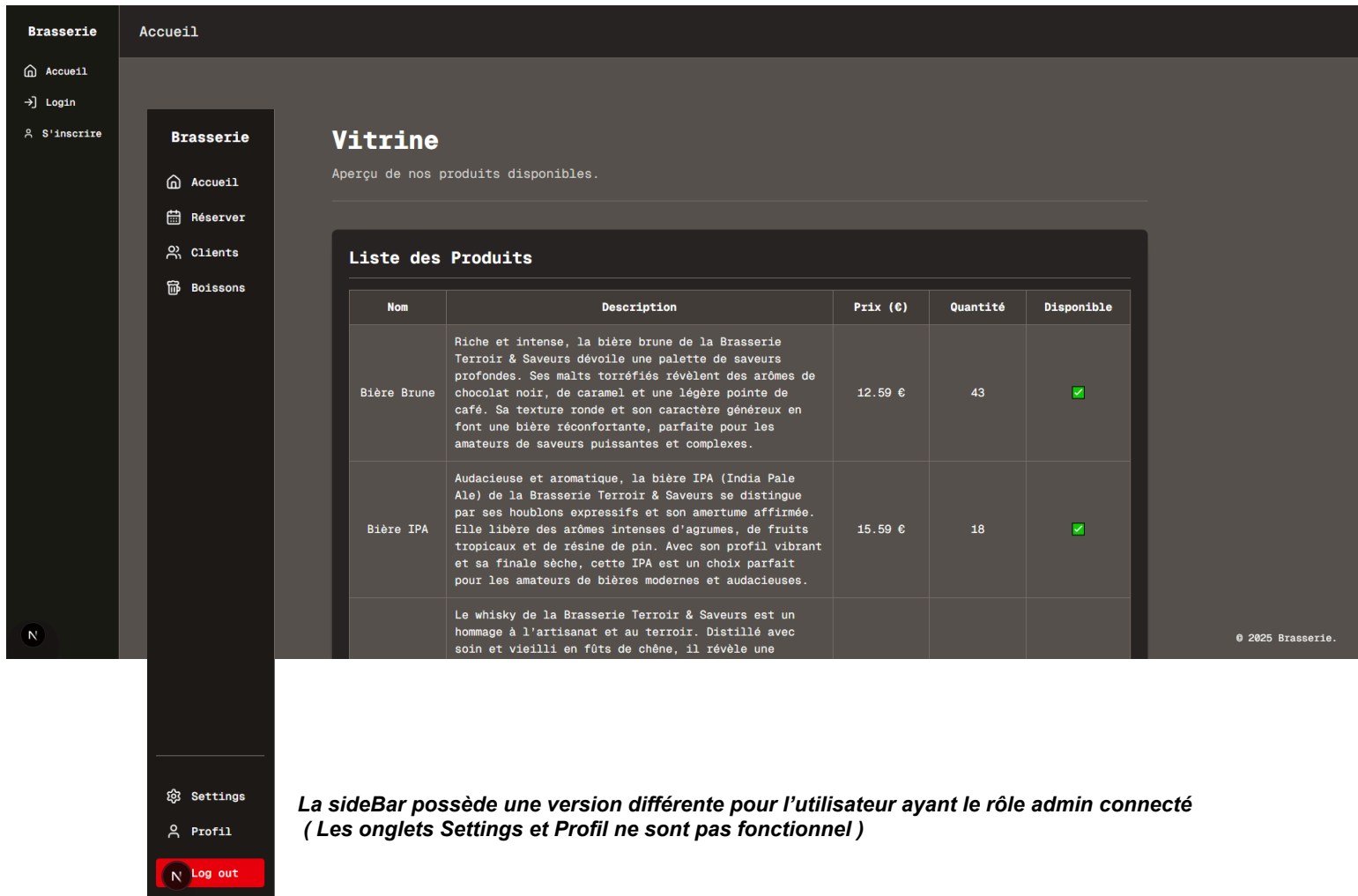
- Créer des produits, les modifier, les supprimer et les afficher.
- Créer des utilisateurs, les modifier, les supprimer et les afficher
- Créer des réservations, les modifier, les supprimer et les afficher

Le tout est lié à une base de données MySQL.

2- Aperçu visuel du projet

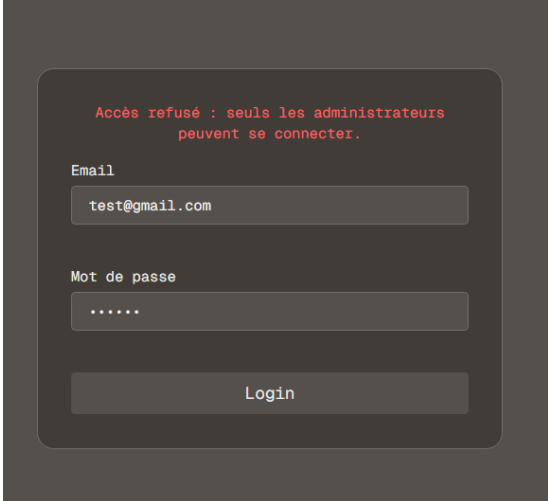
Pour commencer, je vous présente les différents écrans, ainsi que certains aspects visuelles,

À l'arrivée, les clients ont accès à la vitrine de nos produits, à l'aide de la sidebar il est possible de se login (admin) ou encore de se créer un compte utilisateur (ne donnant pas l'accès au dashboard admin)



Login et register

- à l'arrivée sur la page login , un formulaire se présente, permettant aux admins de se login (l'accès sera refusé aux utilisateurs classique) , la sidebar étant tout le temps disponible sur le site permet de retourner dans les autres pages.



Accès refusé : seuls les administrateurs peuvent se connecter.

Email

test@gmail.com

Mot de passe

.....

Login

- En allant dans “S'identifier” les visiteurs peuvent créer leurs compte pour permettre à l'admin de passer une commande pour eux. une fois le compte créé ils seront redirigé vers la vitrine



Nom

Entrez votre nom...

Prénom

Entrez votre prénom...

Email

Entrez votre email...

Mot de passe





Entrez votre mot de passe...

S'inscrire

Réservation

- Si un admin se login il aura accès au dashboard , une réplique de l'affichage qu'ont les visiteurs.

En naviguant via la sidebar ils ont accès à la gestion de réservation où ils peuvent voir les réservations en cours

Réservations en cours				
⊕ Ajouter une réservation				
Email	Date et Heure	Produits	Statut	Actions
test@gmail.com	29/04/2025 14:30	Bière Brune (2), Bière IPA (3), Whisky (1), Gin (1), Bière Blonde (1)	En cours	 
alexis.lamonnier@gmail.com	28/04/2025 21:54	Whisky (1)	Expédié	 

- En cliquant sur "Ajouter une réservation" les admins peuvent créer une commande en y stipulant l'utilisateur voulant passer une commande (utilisateur présent en bdd) , choisir un produit , si le produit est choisi il peut informer la quantité voulu (la quantité ne peut dépasser le stock présent) , il peut indiquer la date et heure de la prise de la commande par l'admin , et indiqué son statut (pending de base)

Ajouter une réservation

Sélectionnez un utilisateur ▼

Produits

- Bière Brune - 12.59€
- Bière IPA - 15.59€
- Whisky - 35.59€
- Gin - 32.59€
- Bière Blonde - 12.54€

Date et Heure

02/05/2025 20:28

Statut

Pending ▼

✓ ✕

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle

- Les admins peuvent modifier une commande via le bouton dans les actions de l'affichage des réservations, l'utilisateur / client ne peut être modifié, cependant les articles, leurs quantités, la date et le statut peuvent l'être. De plus, il est possible de supprimer une réservation.

Modifier une réservation

Produits

- ☒ Bière Brune - 12.59€ 2
- ☒ Bière IPA - 15.59€ 3
- ☒ Whisky - 35.59€ 1
- ☒ Gin - 32.59€ 1
- ☒ Bière Blonde - 12.54€ 1

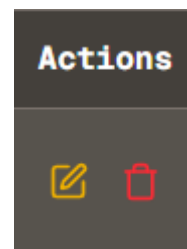
Date et Heure

29/04/2025 14:30

Statut

En cours


✓ ✕



Gestion des Utilisateurs

- Depuis la sidebar il est possible de naviguer vers la Gestion des Utilisateurs .

Les admins peuvent voir la liste des utilisateurs et à l'aide des boutons dans actions peuvent les modifier et supprimer

Liste des Utilisateurs			
Nom	Prénom	Email	Actions
lamonnier	alexis	alexis.lamonnier@gmail.com	 
test	test	test@gmail.com	 

- Les admins peuvent modifier les utilisateurs à l'aide du boutons dans actions, le nom, prénom et les emails peuvent être modifié

Modifier l'utilisateur "lamonnier"

Nom

lamonnier

Prénom

alexis

Email

alexis.lamonnier@gmail.com

✓ ✕

Gestion des Produits

- Depuis la sidebar il est possible de naviguer vers la Gestion des Produits .

Les admins peuvent voir la liste des produits et à l'aide des boutons dans actions peuvent les modifier et supprimer

Liste des Produits					
+ Ajouter un produit					
Nom	Description	Prix (€)	Quantité	Disponible	Actions
Bière Brune	Riche et intense, la bière brune de la Brasserie Terroir & Saveurs dévoile une palette de saveurs profondes. Ses malts torréfiés révèlent des arômes de chocolat noir, de caramel et une légère pointe de café. Sa texture ronde et son caractère généreux en font une bière réconfortante, parfaite pour les amateurs de saveurs puissantes et complexes.	12.59 €	0	✗	 
Autres produits et promotions - la bière IPA (Texte)					

- Les admins peuvent modifier les produits en spécifiant, le nouveau nom, description, prix et sa quantité. une fois validé la disponibilité du produit va être rétabli si la quantité est supérieur à 0 unité débloquent ainsi le fait de pouvoir les réserver

Modifier le produit "Bière Brune"



Nom du produit

Description

Riche et intense, la bière brune de la Brasserie Terroir & Saveurs dévoile une palette de saveurs profondes. Ses malts torréfiés révèlent des arômes de chocolat noir, de caramel et une légère pointe de café. Sa texture ronde et son caractère généreux en font une bière réconfortante, parfaite pour les amateurs de saveurs puissantes et complexes.

Prix (€)

Quantité

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle

- En cliquant sur ajouter un produit , les admins peuvent en ajouter , en spécifiant le nom, sa description, son prix et sa quantité

Ajouter un nouveau produit

Nom du produit

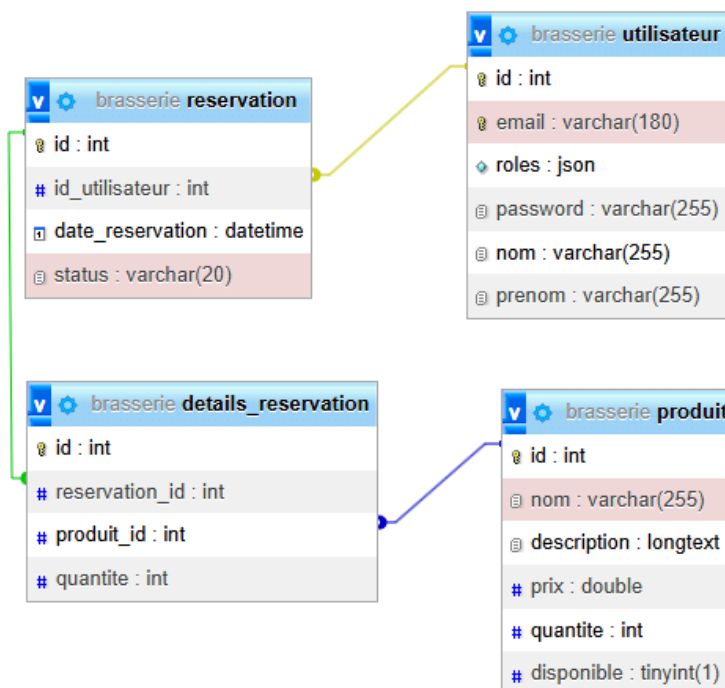
Description

Prix (€)

Quantité

3- Base de données

Schéma de la BDD



Lien github vers les controllers

Authentification : <https://github.com/alexkjzz/brasserie/blob/main/api/src/Controller/AuthController.php>

Détails des réservations : <https://github.com/alexkjzz/brasserie/blob/main/api/src/Controller/DetailsReservationController.php>

Produits : <https://github.com/alexkjzz/brasserie/blob/main/api/src/Controller/ProduitController.php>

Réservations : <https://github.com/alexkjzz/brasserie/blob/main/api/src/Controller/ReservationController.php>

Utilisateurs : <https://github.com/alexkjzz/brasserie/blob/main/api/src/Controller/UtilisateurController.php>

Lien github vers les appels API

ProduitApi : <https://github.com/alexkjzz/brasserie/blob/main/web/src/services/produitApi.ts>

RéservationApi : <https://github.com/alexkjzz/brasserie/blob/main/web/src/services/reservationApi.ts>

UtilisateurApi : <https://github.com/alexkjzz/brasserie/blob/main/web/src/services/utilisateurApi.ts>

4- Détails techniques

Mes pages disponibles uniquement aux admin ont une sécurité en plus qui est le composant **AdminRedirect** renvoyant les personnes vers le login si ils ne sont pas admin ou si ils ne sont pas connecté cela permet donc de s'assurer qu'il n'y ai pas d'intrusion via l'url possible.

```
return (  
  <AdminRedirect>  
    <main className="flex justify-center items-start w-full p-12">  
      <div className="gap-10 w-full max-w-6xl mt-12 flex flex-col items-start">  
        <section className="text-left w-full border-b border-stone-500 pb-6">  
          <h1 className="text-4xl font-extrabold text-white">Gestion des Produits</h1>  
          <p className="text-stone-300 mt-3 text-lg leading-relaxed max-w-3xl">  
            Ajoutez, modifiez ou supprimez les produits disponibles dans la boutique.  
          </p>  
        </div>  
      </main>  
    </AdminRedirect>  
  </div>  
)
```

```
useEffect(() => {  
  const token = localStorage.getItem("jwtToken");  
  if (!token) {  
    router.push("/"); // Redirige vers l'accueil si non connecté  
    return;  
  }  
  
  try {  
    const payload = JSON.parse(atob(token.split(".")[1]));  
    if (!payload.roles.includes("ROLE_ADMIN")) {  
      router.push("/"); // Redirige vers l'accueil si pas admin  
    }  
  }  
})
```

Lien vers **AdminRedirect** :

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle

<https://github.com/alexkjzz/brasserie/blob/main/web/src/components/AdminRedirect.tsx>

Ma Sidebar possède une détection de rôle.

Si la personne est un visiteur ou si il est admin les items ne seront pas les mêmes.

```
const guestItems = [
  { href: "/", label: "Accueil", icon: <Home size={20} /> },
  { href: "/login", label: "Login", icon: <LogIn size={20} /> },
  { href: "/register", label: "S'inscrire", icon: <User size={20} /> },
];

const adminItems = [
  { href: "/dashboard", label: "Accueil", icon: <Home size={20} /> },
  { href: "/reservations", label: "Réserver", icon: <CalendarDays size={20} /> },
  { href: "/users", label: "Clients", icon: <Users size={20} /> },
  { href: "/products", label: "Boissons", icon: <Beer size={20} /> },
];

const settingsItems = [
  { href: "/settings", label: "Settings", icon: <Settings size={20} /> },
  { href: "/profile", label: "Profil", icon: <User size={20} /> },
];
```

De plus elle possède une fonction de Déconnexion avec suppression de token pour permettre de réinitialiser les états de la Sidebar admin/users et de revenir à une utilisation tel que peuvent avoir les visiteurs

```
// Fonction de déconnexion
const handleLogout = () => {
  localStorage.removeItem("jwtToken"); // Suppression du token
  window.location.href = "/"; // Redirection vers l'accueil
};
```

Lien vers la **SideBar** :

<https://github.com/alexkjzz/brasserie/blob/main/web/src/components/Sidebar.tsx>

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle

Ici vous avez un exemple de comment les data peuvent être fetch (ici c'est pour "Produits") avec une vérification du token pour savoir si la personne a les droits de faire appel à l'API

```
useEffect(() => {
  const fetchData = async () => {
    try {
      const token = localStorage.getItem("jwtToken");
      if (!token) throw new Error("Token JWT requis.");
      const data = await fetchProduits(token);
      setProduits(data);
    } catch (err) {
      setError(err instanceof Error ? err.message : "Erreur inconnue.");
    } finally {
      setLoading(false);
    }
  };

  fetchData();
}, []);
```

de plus j'utilise des handler pour les changements , ajouts et suppressions (encore un exemple mais la méthode est partout pareille)

```
const handleSave = async () => {
  try {
    const token = localStorage.getItem("jwtToken");
    if (!token) throw new Error("Token JWT requis.");

    const method = editingProduit ? "PUT" : "POST";
    await saveProduit(token, formData, method, editingProduit?.id);
    closeModal();
    const data = await fetchProduits(token);
    setProduits(data);
  } catch (err) {
    console.error(err);
  }
};
```

Lien vers la page ayant servi d'exemple ici :

<https://github.com/alexkjzz/brasserie/blob/main/web/src/app/products/page.tsx>

J'utilise un dossier models avec un fichier contenant mes interfaces :
(exemple avec Produit)

Lien vers le dossier models :

<https://github.com/alexkjzz/brasserie/blob/main/web/src/models/types.ts>

```
export interface Produit {
  id: number;
  nom: string;
  description: string;
  prix: number;
  quantite: number;
  disponible: boolean;
}
```

5- Sources

<https://nextjs.org/docs>

<https://www.typescriptlang.org/>

<https://symfony.com/>

<https://www.php.net/>