

GESTION
D'ELECTRO X

RAPPORT PROJET

REALISER PAR : ACHRAF BOUHIA

ENCADRE PAR :
MME MOUSSAMMI

SOMMAIRE

Remerciement	1
Introduction	2
Au Cœur du Projet	3
Interfaces	4
Conclusion	5

REMERCIEMENT

01



Je tiens à exprimer mes sincères remerciements à ma professeure MME MOUSSAMMI pour son soutien constant tout au long de ce projet. Son expertise et ses précieux conseils m'ont permis d'améliorer considérablement mes compétences techniques . Je souhaite également remercier pour l'excellent cahier des charges, qui a fourni une direction claire et détaillée, m'aidant à développer mes connaissances et à perfectionner mes compétences pratiques. Ce projet a été une véritable opportunité d'apprentissage et une expérience enrichissante.

INTRODUCTION

02



Le projet de développement du dashboard ADMIN pour le site web Electrolux a été conçu dans le but d'offrir une interface moderne et intuitive permettant une gestion efficace des produits et des utilisateurs. Ce tableau de bord centralise les informations essentielles, telles que les statistiques des produits, les quantités disponibles, ainsi que des données relatives à l'équipe et aux utilisateurs. Le design responsive et l'intégration d'outils comme les graphiques dynamiques, les carrousels de produits et les fonctionnalités de recherche permettent une gestion fluide et une prise de décision rapide. L'interface est construite avec des technologies modernes telles que Bootstrap, Tailwind CSS et Chart.js, garantissant ainsi une expérience utilisateur agréable et performante.

AU CŒUR DU PROJET

03

PROBLÉMATIQUE :

Dans un environnement commercial dynamique, la gestion efficace des produits et des Administrateurs essentielle pour assurer le bon fonctionnement d'une entreprise. Pour un site e-commerce tel qu'Electrolux, il devient crucial de disposer d'un tableau de bord intuitif et performant, permettant aux administrateurs de suivre les stocks, d'analyser les performances des produits, et de gérer les utilisateurs de manière optimale. L'absence d'une telle interface peut rendre la gestion complexe, prendre du temps et entraîner des erreurs humaines.

POURQUOI ?

Ce projet a été développé pour répondre à cette problématique en proposant une solution complète : un tableau de bord administrateur centralisé, où toutes les informations pertinentes peuvent être consultées et manipulées de manière simple et efficace. L'objectif est de faciliter la prise de décision en temps réel grâce à une vue d'ensemble claire des stocks, des produits, ainsi que des utilisateurs. En intégrant des fonctionnalités modernes et dynamiques telles que des graphiques interactifs et une interface responsive, le tableau de bord permet d'améliorer la gestion des opérations, d'optimiser les performances et de garantir une expérience utilisateur de qualité.

OUTILS DE DÉVELOPPEMENT



tailwindcss

Pour le développement de ce tableau de bord administrateur, plusieurs outils et technologies modernes ont été utilisés pour garantir une interface réactive, esthétique et fonctionnelle. Le projet repose principalement sur HTML pour la structure de la page, permettant une organisation claire des éléments du dashboard. Le CSS a été employé pour la mise en forme et la personnalisation visuelle des composants, assurant une présentation soignée et professionnelle. En complément, Bootstrap a été intégré pour accélérer le processus de développement en utilisant ses classes pré-définies et sa grille flexible, permettant de créer un design responsive et cohérent. Enfin, Tailwind CSS, une bibliothèque utilitaire, a été utilisée pour affiner le style visuel avec une approche modulaire et personnalisable, offrant ainsi une grande flexibilité dans le design sans compromettre la performance du site. Ensemble, ces technologies assurent une interface moderne, rapide et facilement maintenable.

GESTION DE VERSION AVEC

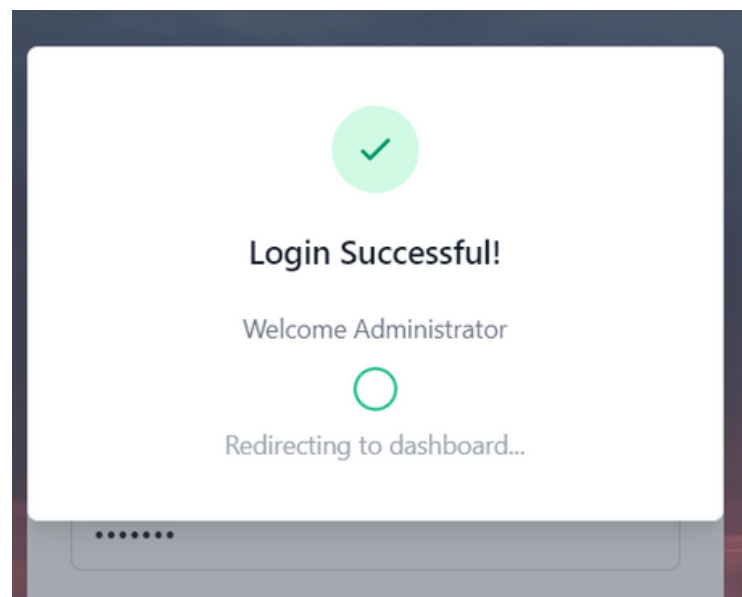
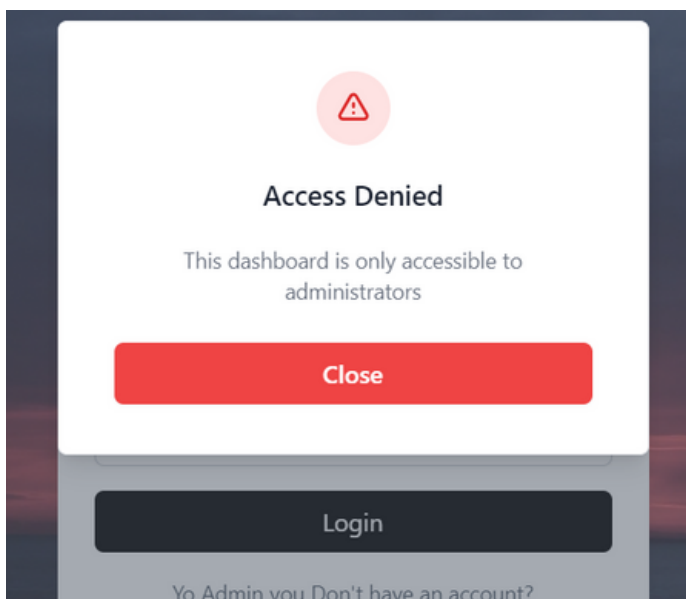
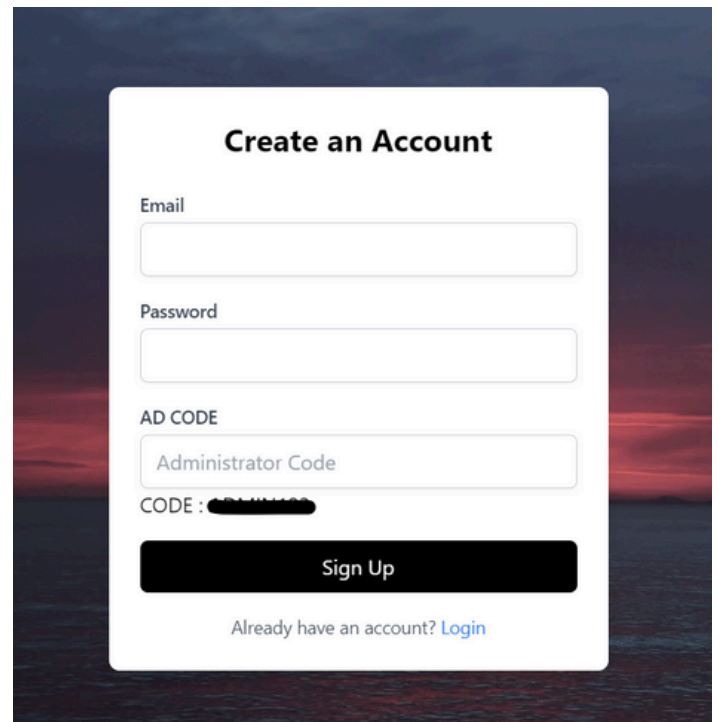
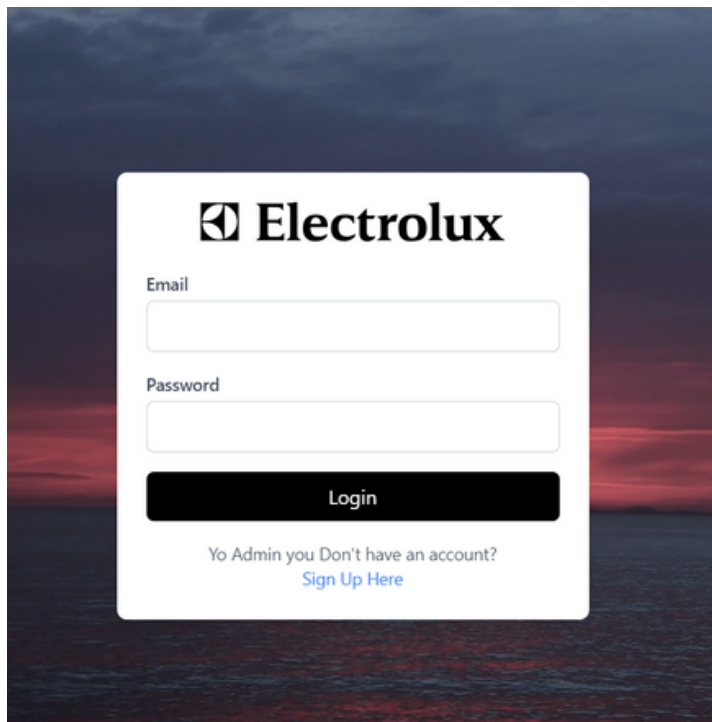


Pour assurer un développement collaboratif et une gestion efficace des différentes versions du projet, GitHub a été utilisé comme plateforme de gestion de version. En utilisant Git, les modifications apportées au code sont suivies de manière systématique, permettant de revenir à une version antérieure en cas de besoin, et facilitant la gestion des différentes branches de développement. Chaque fonctionnalité ou amélioration du projet a été réalisée sur une branche dédiée, puis fusionnée avec la branche principale après validation. Cette méthode garantit une gestion propre et ordonnée du code, tout en permettant de travailler simultanément sur plusieurs aspects du projet sans risquer de conflits. GitHub a également été utilisé pour héberger le projet et faciliter les collaborations, en offrant un espace de stockage sécurisé et en permettant une révision facile des contributions à travers des pull requests. Cela a non seulement amélioré la qualité du code, mais aussi facilité la gestion des versions et des déploiements.

INTERFACES

04

INTERFACE NO 1



L'interface de connexion admin offre une expérience utilisateur fluide grâce à :

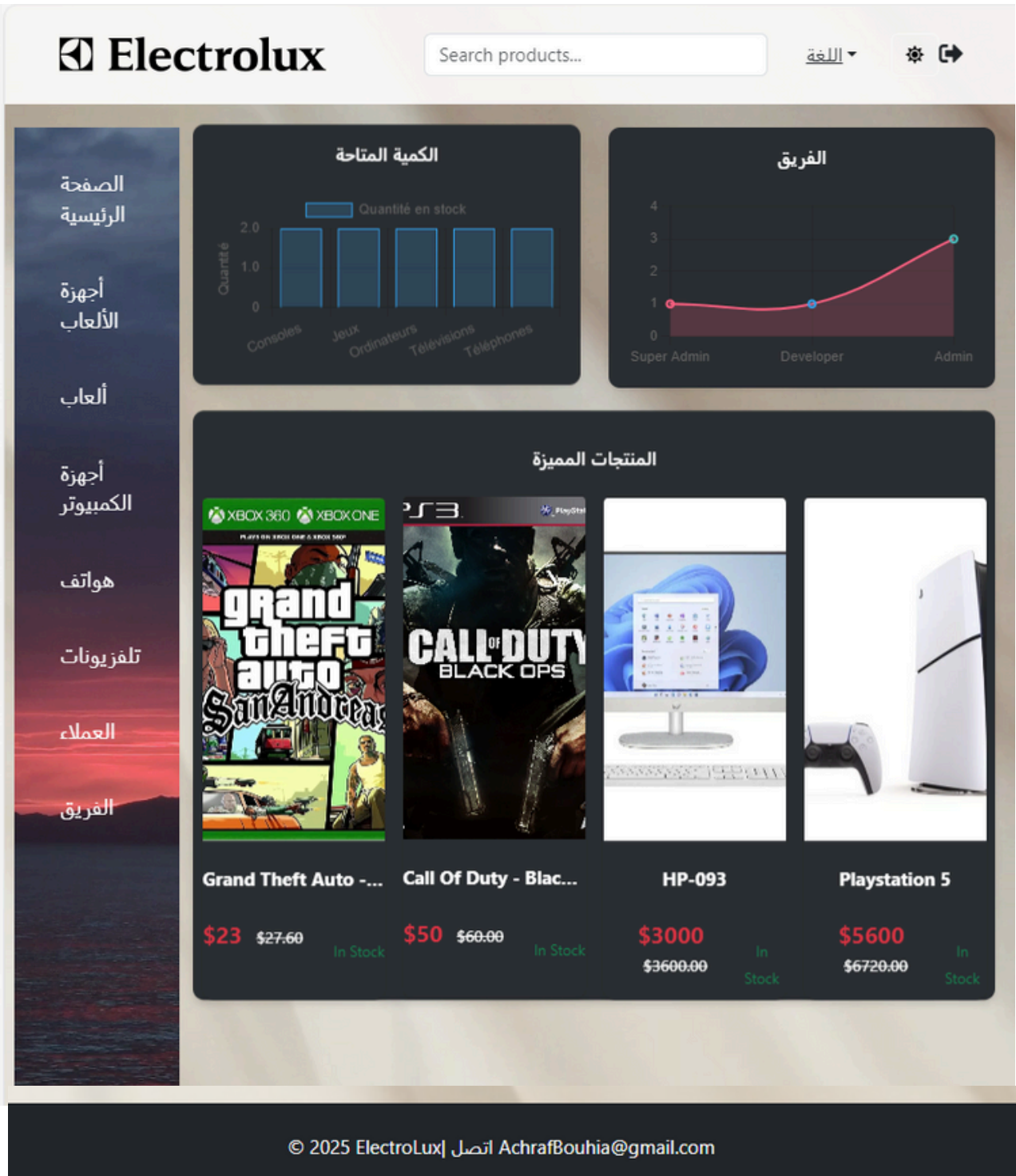
Une gestion claire des états entre connexion et inscription.

Des validations robustes pour garantir la sécurité.

Des animations modernes pour des alertes d'erreur et de succès.

Un design esthétique avec une vidéo d'arrière-plan pour une meilleure présentation visuelle.

INTERFACE NO 2



PAGE D'ACCUEIL DU DASHBOARD :

Une vue d'ensemble de la page principale du tableau de bord, mettant en évidence les statistiques clés des produits et les graphiques interactifs.

SIDEBAR

Présentation du menu latéral, avec des liens vers les différentes catégories de gestion (Consoles, Jeux, Ordinateurs, Télévisions, etc.) et le fond vidéo dynamique pour une meilleure esthétique.

SECTION DE RECHERCHE :

Capture montrant la barre de recherche avec son design moderne, permettant une recherche rapide des produits et l'affichage des résultats en temps réel.

SECTION DES PRODUITS EN VEDETTE

Une vue détaillée du carrousel des produits en vedette, mettant en avant les produits sélectionnés avec leurs images et descriptions.

GRAPHIQUES DYNAMIQUES

Illustrations des graphiques générés à l'aide de Chart.js, présentant des données telles que les quantités disponibles et les performances de l'équipe.

MODALE DE DÉTAIL DU PRODUIT :

Capture de la fenêtre modale affichant les détails d'un produit sélectionné, incluant son image, son prix, et sa quantité disponible.

MODE SOMBRE ET CLAIR

Capture comparant les deux thèmes disponibles, mettant en avant la fonctionnalité de basculement entre le mode clair et sombre pour améliorer l'expérience utilisateur.

FOOTER

Vue du pied de page contenant les informations de contact et les crédits.

QUELQUE SCRIPTS :

```
<section>
  <section>
    <div class="container-fluid">
      <div class="row d-flex">
        <!-- Sidebar -->
        <div id="sidebar" class="col-md-2 text-white vh-100 px-4 sidebar">
          <div class="sidebar-video-container">
            <video autoplay muted loop class="sidebar-video">
              <source src="assets/videos/sidebar-bg.mp4" type="video/mp4">
            </video>
          </div>
          <div class="sidebar-overlay"></div>
        </div>
        <div class="nav flex-column">
          <br>
          <li class="nav-item mb-3">
            <a data-translate-key="home" href="index.html" class="nav-link text-white"><i class="fas fa-home me-2">
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="consoles" href="pages/Consoles.html" class="nav-link text-white"><i class="fas fa-di
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="games" href="pages/Games.html" class="nav-link text-white"><i class="fas fa-di
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="computers" href="pages/Pc.html" class="nav-link text-white"><i class="fas fa-l
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="phones" href="pages/phones.html" class="nav-link text-white"><i class="fas fa-
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="televisions" href="pages/Tv.html" class="nav-link text-white">Televisions</a>
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="clients" href="pages/Clients.html" class="nav-link text-white">Clients</a>
          </li>
          <li class="nav-item mb-3">
            <a data-translate-key="team" href="pages/users.html" class="nav-link text-white">Team</a>
          </li>
        </ul>
      </div>
    </div>
  </section>
</div>
```

```
const admins = JSON.parse(localStorage.getItem('admins')) || [];
const ADMIN_SECRET_CODE = 'ADMIN123';

const loginForm = document.getElementById('loginForm');
const signupForm = document.getElementById('signupForm');
const formContainer = document.getElementById('formContainer');

document.getElementById('switchToSignup').addEventListener('click', () => {
  loginForm.classList.add('hidden');
  signupForm.classList.remove('hidden');
});

document.getElementById('switchToLogin').addEventListener('click', () => {
  signupForm.classList.add('hidden');
  loginForm.classList.remove('hidden');
});

document.getElementById('signupButton').addEventListener('click', () => {
  const email = document.getElementById('signupEmail').value;
  const password = document.getElementById('signupPassword').value;
  const adminCode = document.getElementById('adminCode').value;

  if (!email || !password || !adminCode) {
    alert('Please fill in all fields');
    return;
  }

  if (adminCode !== ADMIN_SECRET_CODE) {
    alert('Invalid administrator code');
    return;
  }

  if (admins.find(admin => admin.email === email)) {
    alert('Email is already registered');
    return;
  }

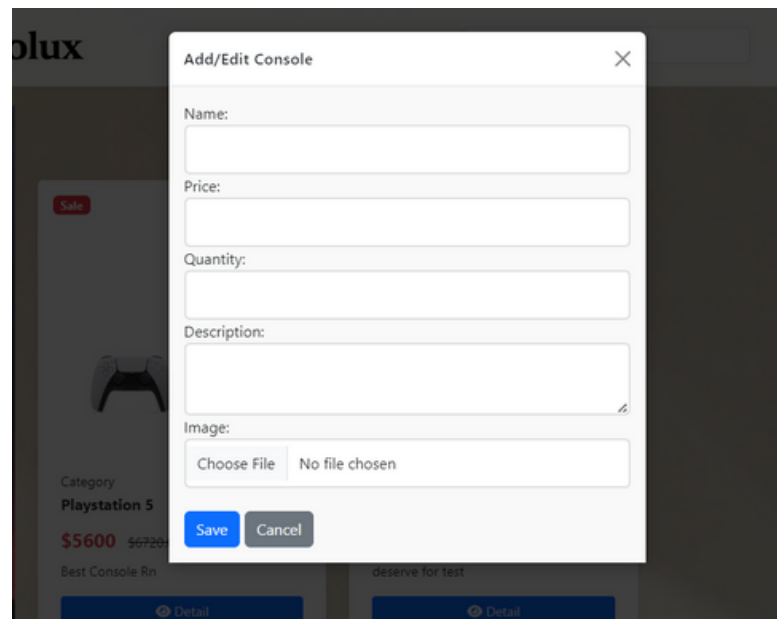
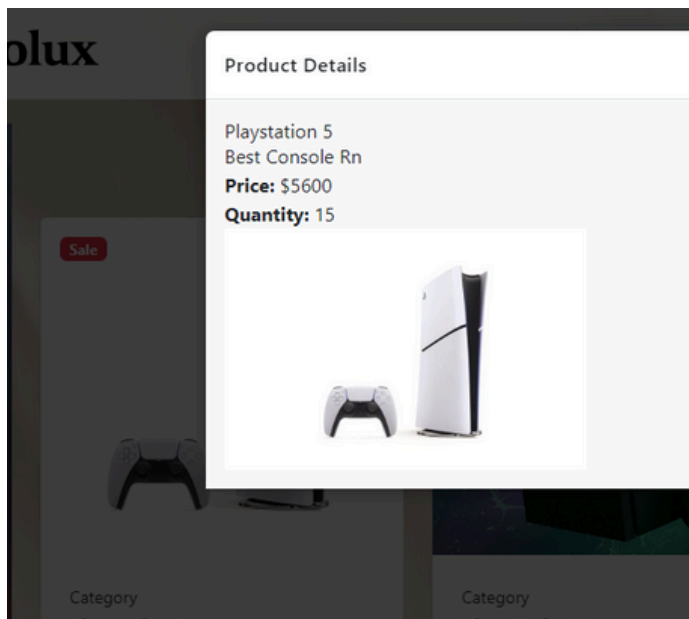
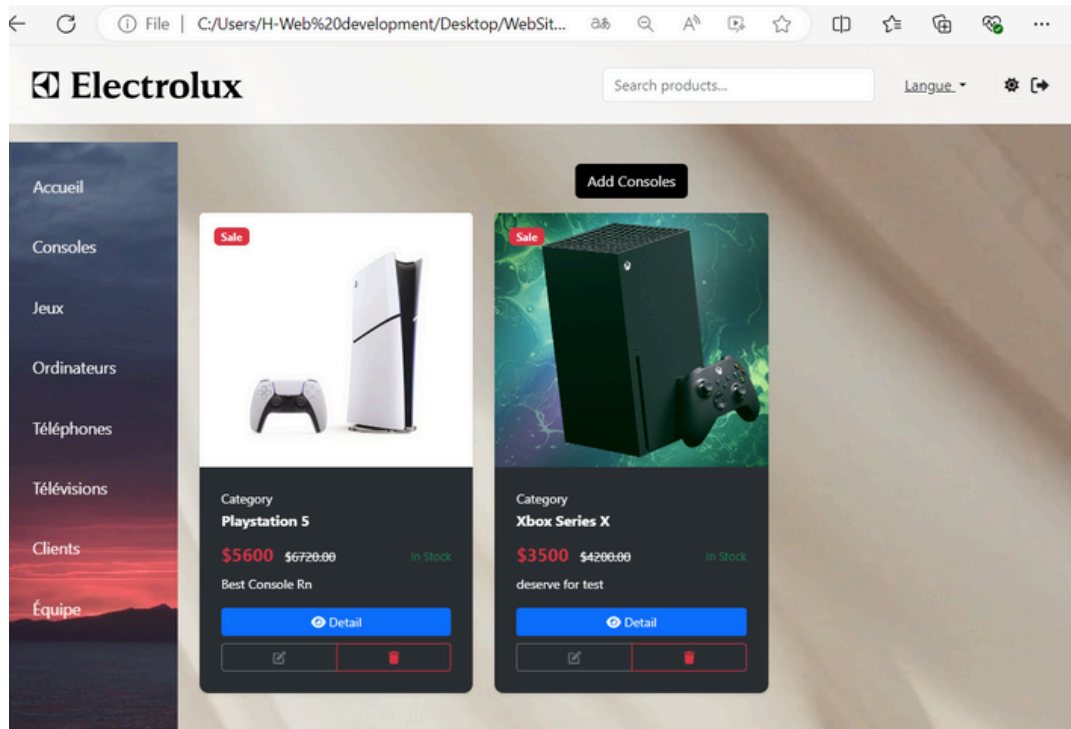
  admins.push({ email, password, role: 'admin' });
  localStorage.setItem('admins', JSON.stringify(admins));
  alert('Administrator account created successfully');
  signupForm.classList.add('hidden');
  loginForm.classList.remove('hidden');
});

document.getElementById('loginButton').addEventListener('click', () => {
  const email = document.getElementById('loginEmail').value;
  const password = document.getElementById('loginPassword').value;

  if (!email || !password) {
    alert('Please fill in all fields');
    return;
  }

  const admin = admins.find(admin => admin.email === email && admin.password === password);
  if (!admin) {
    alert('Invalid email or password');
    return;
  }
  // Login logic continues here
});
```


INTERFACE NO 3



INTERFACE PRINCIPALE :

AFFICHE UNE GRILLE DE CARTES PRODUITS AVEC IMAGE, NOM, PRIX, QUANTITÉ, ET BADGES DYNAMIQUES POUR LE STOCK. BOUTONS POUR VOIR LES DÉTAILS, MODIFIER OU SUPPRIMER UN PRODUIT.

MODAL DE DÉTAILS :

PRÉSENTE LES INFORMATIONS COMPLÈTES D'UN PRODUIT (IMAGE, NOM, DESCRIPTION, PRIX, QUANTITÉ).

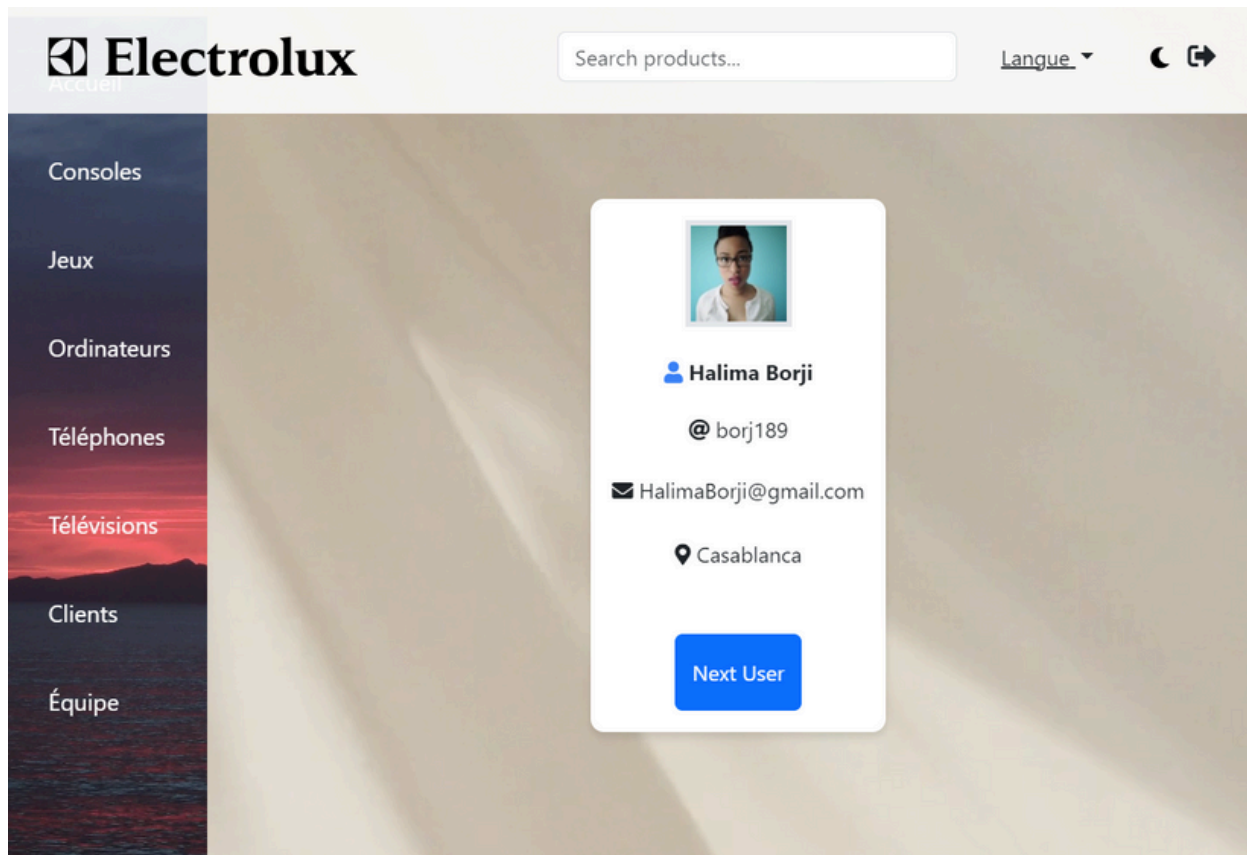
MODAL D'AJOUT/MODIFICATION :

PERMET D'AJOUTER OU DE MODIFIER UN PRODUIT VIA UN FORMULAIRE INTERACTIF.

INTERFACE DE SUPPRESSION :

OPTION POUR SUPPRIMER UN PRODUIT SPÉCIFIQUE OU TOUS LES PRODUITS APRÈS CONFIRMATION.

INTERFACE NO 4



- Une carte affiche les informations d'un client tirées aléatoirement via une API.
- Les données incluent :
 - Photo de profil (avatar).
 - Nom complet (prénom et nom de famille).
 - Nom d'utilisateur.
 - Adresse e-mail.
 - Ville de résidence.
- Bouton de Rafraîchissement :
- Un bouton permet de charger un nouveau client aléatoire à chaque clic.
- Interaction Dynamique :
- Lors du clic, les informations de la carte sont mises à jour grâce à une requête fetch vers l'API Random User.
- Gestion des Erreurs :
- Les erreurs HTTP ou de traitement de données sont capturées et affichées dans la console pour le débogage.

INTERFACE NO 5

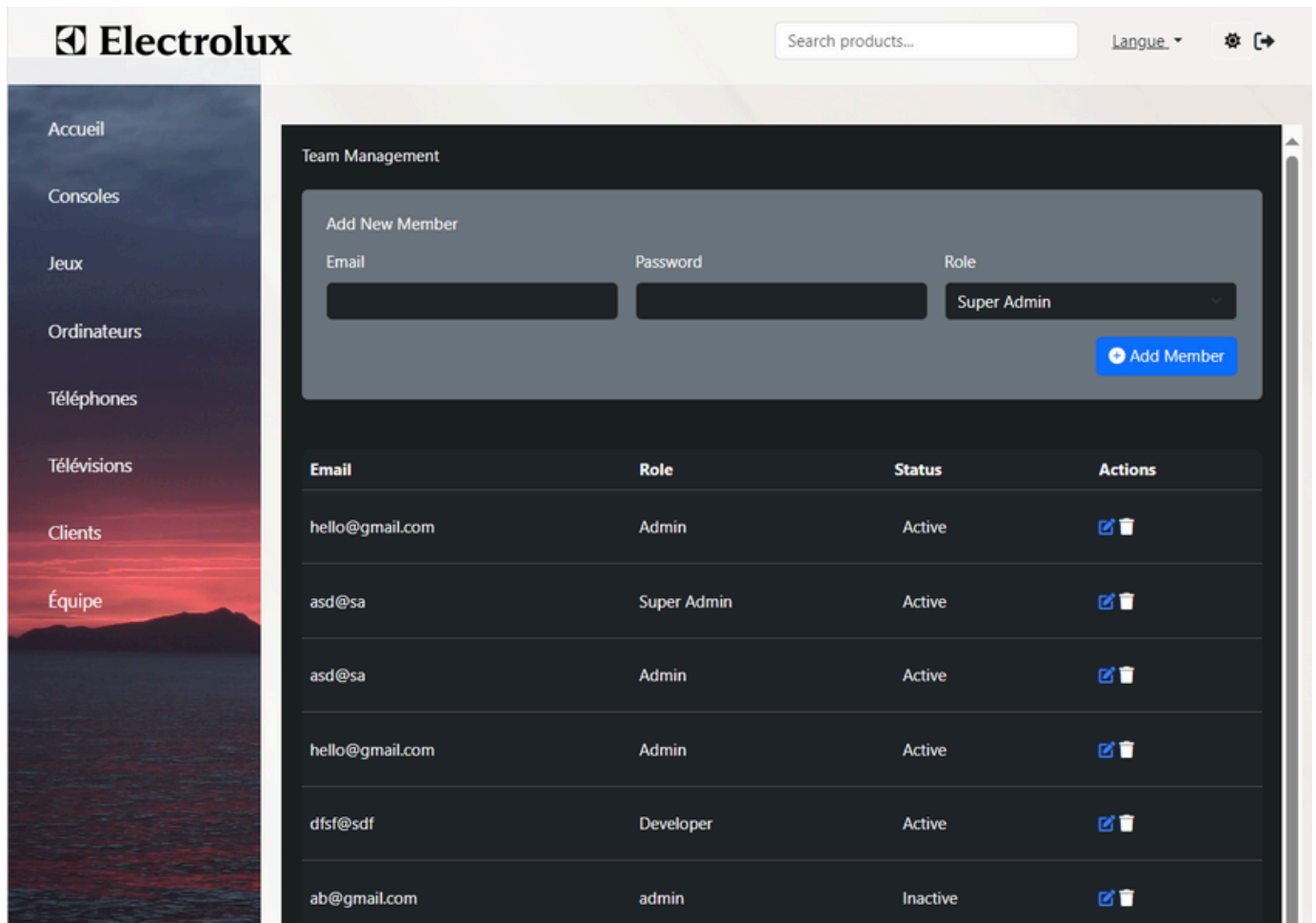


Tableau des Administrateurs :

Affiche une liste des administrateurs avec les colonnes :
Email.

Rôle (Admin, Super Admin, Developer).

Statut Actif/Inactif (indiqué par des badges colorés).

Actions : Modifier ou Supprimer.

Ajout d'un Administrateur :

Formulaire pour ajouter un nouvel administrateur avec les champs :

Email.

Mot de passe.

Rôle.

L'administrateur ajouté est actif par défaut.

Modification d'un Administrateur :

Affichage d'un formulaire modal pré-rempli.

Possibilité de :

Modifier l'email.

Changer le mot de passe (facultatif).

Mettre à jour le rôle.

Activer/Désactiver l'utilisateur.

Suppression d'un Administrateur :

Bouton pour supprimer un administrateur avec une confirmation.

Stockage et Persistance :

Les données des administrateurs sont sauvegardées dans le localStorage pour persistance.

Validation et Gestion des Erreurs :

Alertes pour champs manquants lors de l'ajout.

Gestion des erreurs lors de la modification avec affichage dans la console.

CONCLUSION

05

CONCLUSION

Le projet "Tableau de Bord - Gestion d'ELECTRO " est une application web intuitive et interactive conçue pour simplifier la gestion et la visualisation des données commerciales. L'objectif principal de ce projet est de fournir un outil performant et accessible qui centralise la gestion des produits, des équipes, et des données de marché.

Ce projet améliore considérablement l'efficacité de la gestion des données, offrant une interface utilisateur intuitive et des fonctionnalités avancées. Il établit une base solide pour les futures mises à jour ou intégrations, en mettant l'accent sur l'accessibilité, l'esthétique et la performance.