Rapport Technique : Portfolio React Next.js

AZANGUE LEONEL DELMAT

27 mars 2025

Table des matières

Chapitre 1

Introduction

Ce portfolio a été développé avec React Next.js pour répondre aux besoins d'un étudiant en génie informatique spécialisé en science des données et cybersécurité. Il intègre une interface publique et une section admin protégée.

Chapitre 2

Architecture Globale

2.1 Stack Technique

— **Framework** : Next.js 14 (React Server Components)
— **Styling** : Tailwind CSS
— **Authentification** : Firebase Authentication
— **Base de données** : Firebase Firestore
— **Déploiement** : Vercel (publique) + Firebase Hosting (admin)

Chapitre 3

Pages Publiques

3.1 Accueil

3.1.1 Fonctionnalités

— Hero section avec présentation personnelle
— Grille de projets filtrables
— Call-to-action vers la section contact

3.1.2 Technologies Utilisées

— `getStaticProps` pour le SEO
— Composants réutilisables (Header/Footer)
— Animations CSS avec Tailwind

3.1.3 Extrait de Code

Listing 3.1 – Hero Section

```
const HomePage = () => {
  return (
    <section className="bg-gradient-to-r from-blue-600 to-blue-900 text
       -white min-h-screen">
      <div className="container mx-auto p-8 pt-24 md:p-16">
        <div className="max-w-3xl mx-auto text-center">
          <h1 className="text-4xl md:text-6xl font-bold mb-4">
            Votre Nom
          </h1>
          <p className="text-xl md:text-2xl mb-8">
              tudiant   en G nie Informatique    l'ENSPY | Passionn
                par la Science des Donn es et la Cybers curit
          </p>
          <a href="#contact" className="bg-blue-800 px-8 py-3 rounded-
             lg hover:bg-blue-900 transition-colors">
            Contactez-moi
          </a>
        </div>
      </div>
    </section>
```

```
  );
};
```

3.2 Projets

3.2.1 Fonctionnalités

— Cartes de projets avec détails techniques

— Filtres par catégorie (Data Science/Cybersécurité)

— Liens vers démos GitHub

3.2.2 Technologies Utilisées

— `ProjectCard` : Composant réutilisable

— Grille responsive avec `grid-template-columns`

— Progress bars animées pour les compétences

3.2.3 Extrait de Code

Listing 3.2 – Composant ProjectCard

```
interface ProjectCardProps {
  title: string;
  description: string;
  techStack: string[];
  link: string;
}

const ProjectCard = ({ title, description, techStack, link }:
   ProjectCardProps) => {
  return (
    <div className="project-card">
      <h3 className="font-bold mb-2">{title}</h3>
      <p className="text-gray-600 mb-4">{description}</p>
      <div className="flex flex-wrap gap-2 mb-4">
        {techStack.map((tech) => (
          <span key={tech} className="bg-gray-100 px-3 py-1 rounded-md
             text-sm">
            {tech}
          </span>
        ))}
      </div>
      <a
        href={link}
        target="_blank"
        rel="noopener noreferrer"
        className="text-blue-600 hover:text-blue-800 transition-colors"
      >
        Voir le projet
      </a>
```

```
      </div>
    );
};
```

3.3 Formations

3.3.1 Fonctionnalités

— Timeline éducative
— Section "En cours" pour formations actuelles
— Certifications avec badges

3.3.2 Technologies Utilisées

— Timeline CSS personnalisée
— Composants `SkillBadge`
— Intégration de PDF (CV)

3.3.3 Extrait de Code

Listing 3.3 – Timeline CSS

```
.timeline {
  @apply flex flex-col gap-8;
}

.timeline-item {
  @apply relative pl-8;
}

.timeline-content {
  @apply bg-white p-4 rounded-lg shadow-sm;
}
```

3.4 Contact

3.4.1 Fonctionnalités

— Formulaire de contact avec validation
— Carte interactive (Google Maps API)
— Liens sociaux

3.4.2 Technologies Utilisées

— `Formik` pour la gestion des formulaires
— Validation côté client
— Intégration d'API externes

3.4.3 Extrait de Code

Listing 3.4 – Formulaire Contact

```
const ContactPage = () => {
  return (
    <form className="space-y-4">
      <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
        <input
          type="text"
          placeholder="Nom"
          className="w-full p-2 border rounded-md"
        />
        <input
          type="email"
          placeholder="Email"
          className="w-full p-2 border rounded-md"
        />
      </div>
      <textarea
        rows={5}
        placeholder="Message"
        className="w-full p-2 border rounded-md"
      />
      <button
        type="submit"
        className="bg-blue-600 text-white px-8 py-3 rounded-lg hover:bg
           -blue-700 transition-colors"
      >
        Envoyer
      </button>
    </form>
  );
};
```

Chapitre 4

Section Admin

4.1 Dashboard

4.1.1 Fonctionnalités

— CRUD pour les formations

— Interface sécurisée avec Firebase Auth

— Historique des modifications

4.1.2 Technologies Utilisées

— `react-firebase-hooks` pour l'authentification

— `Formik` avec validation

— Firebase Security Rules

4.1.3 Extrait de Code

Listing 4.1 – CRUD Formations

```
const Dashboard = () => {
  const [user] = useAuthState(auth);
  const db = getFirestore();

  const formik = useFormik({
    initialValues: {
      nom: '',
      dateDebut: '',
      dateFin: '',
      description: ''
    },
    onSubmit: async (values) => {
      await addDoc(collection(db, 'formations'), {
        ...values,
        userId: user?.uid,
        dateAjout: new Date().toISOString()
      });
    }
  });

```

```
  return (
    <form onSubmit={formik.handleSubmit}>
      <div className="grid grid-cols-1 md:grid-cols-2 gap-4 mb-4">
        <div>
          <label className="block mb-2">Nom de la formation</label>
          <input
            type="text"
            name="nom"
            onChange={formik.handleChange}
            value={formik.values.nom}
            className="w-full p-2 border rounded-md"
          />
        </div>
      </div>
      <button type="submit">Enregistrer</button>
    </form>
  );
};
```

4.2 Login

4.2.1 Fonctionnalités

— Authentification par email/password

— Redirection conditionnelle

— Gestion des erreurs

4.2.2 Technologies Utilisées

— `getServerSideProps` pour l'authentification

— Firebase Auth SDK

— Protection des routes

4.2.3 Extrait de Code

Listing 4.2 – Authentification

```
const Login = () => {
  const [user, setUser] = useState(null);
  const [error, setError] = useState(null);

  const handleSubmit = async (email: string, password: string) => {
    try {
      const userCredential = await signInWithEmailAndPassword(auth,
          email, password);
      setUser(userCredential.user);
    } catch (error) {
      setError(error.message);
    }
  };
```

```
13
14   return (
15     <form onSubmit={(e) => e.preventDefault()}>
16       <input
17         type="email"
18         placeholder="Email"
19         className="w-full p-2 border rounded-md"
20       />
21       <input
22         type="password"
23         placeholder="Mot de passe"
24         className="w-full p-2 border rounded-md"
25       />
26       <button
27         type="submit"
28         onClick={() => handleSubmit('email', 'password')}
29       >
30         Se connecter
31       </button>
32     </form>
33   );
34 };
```

Chapitre 5

Bonnes Pratiques

5.1 Cybersécurité

— Chiffrement des données avec Firebase
— Validation côté serveur
— Protection contre les XSS

5.2 Optimisation

— Pré-réndering statique (Next.js)
— Lazy loading des images
— Compression des assets

5.3 Maintenabilité

— Architecture modulaire
— Typage TypeScript
— Documentation via JSDoc

Chapitre 6

Conclusion

Ce portfolio démontre une intégration complète des compétences techniques en développement web moderne, avec une attention particulière à la sécurité et l'expérience utilisateur.