

## Cahier des Charges - Portfolio de MELL CANAC

### 1. Présentation du projet

- **Nom du projet :** Portfolio de MELL CANAC.
- **Objectif :** Présenter mon parcours, mes compétences techniques (Full Stack - Back-end, Front-end, DevOps) et mes réalisations à travers un portfolio interactif et moderne. Attirer des recruteurs, des entreprises et des clients potentiels en mettant en avant mon expertise et ma personnalité (ancien rugbyman reconvertis). Format "One-Page" dynamique.
- **Public cible :**
  - Recruteurs et entreprises recherchant un développeur Full Stack.
  - Clients potentiels pour des prestations freelance.
  - Réseau professionnel souhaitant découvrir mon travail.

### 2. Fonctionnalités principales

- **Page d'accueil :**
  - Présentation immersive avec une **carte néon personnalisée** (image card-last.png) affichant mon nom, mon rôle (Développeur Full Stack, Apprenti Chef de Projet), mes compétences clés (Back-end, Front-end, DevOps) et mes stats (60+ projets, 15+ stacks).
  - Liens vers les sections internes (À Propos, Projets, Contact).
  - Design avec un thème **néon cyan** sur fond sombre.
- **Section "Mon Parcours" :**
  - Présentation de mon profil ("À Propos de Moi") et de mon expérience professionnelle et sportive ("Mon parcours et Expérience").
  - Disposition côté à côté en desktop avec un **effet d'escalier** (l'une en avant-plan, l'autre en retrait).
- **Section "Compétences Techniques" :**
  - Affichage de mes technologies sous forme de **cartes hexagonales ou carrées** avec des **icônes** et des **niveaux de maîtrise** (optionnel).
  - Thème néon cyan.
- **Section "Mes Projets" :**
  - Grille de **4 cartes de projets** (2x2 sur desktop).
  - Chaque carte affiche le **titre**, une **description courte**, et un **effet visuel néon** au survol.
  - **Clic sur la carte :** Ouvre une **modale** détaillée avec :
    - Image du projet.
    - Description complète.
    - Bouton "**Voir sur GitHub**".

- **Section de commentaires** (voir section 3).
- **Section "Soft Skills"** :
  - Présentation de mes qualités transversales (Leadership, Communication, etc.) via des cartes avec icônes.
- **Section "Contact"** :
  - Formulaire de contact avec **nom, email, message**.
  - L'envoi du formulaire déclenche une **API route Next.js** qui envoie le message à mon adresse e-mail via **Resend**.

### 3. Système de commentaires

- **Technologie utilisée** : API développée en **Next.js (API Routes)** et base de données **Supabase**.
- **Fonctionnement** :
  - Un bouton "Laisser un commentaire" est disponible dans la **modale du projet**.
  - Cela ouvre un formulaire de commentaire (nom, texte) **dans la même modale**.
  - Les données (nom, commentaire, ID du projet, date) sont stockées dans une table comments sur **Supabase**.
  - Les commentaires existants pour le projet sont **récupérés et affichés** dans la modale via une requête Supabase.
  - L'ajout d'un commentaire est **immédiatement visible** dans l'interface (sans recharge complet).

### 4. Design & UX/UI

- **Style visuel** : Moderne, **thème néon cyan, effets de lumière, glassmorphism** (arrière-plan flou transparent).
- **Palette de couleurs** : Principalement des **nuances de cyan néon** sur un **fond sombre** (noir profond).
- **Typographie** : Polices lisibles et modernes (ex: Geist Mono, Inter, ou similaire) pour un rendu clair et professionnel.
- **Interactivité** :
  - Animations fluides au scroll (framer-motion).
  - Effets de **pulsation** ou de **glow** légers sur les éléments interactifs.
  - Transitions douces entre les sections.
  - **Menu burger** responsive pour mobile.
  - **Design responsive** adapté à tous les écrans (mobile, tablette, desktop).

### 5. Technologies utilisées

- **Frontend** : **Next.js** (App Router), **React, Tailwind CSS**.
- **Styling** : **Tailwind CSS** avec configuration personnalisée pour le thème cyan néon, **styled-components** (pour Card.jsx).

- **Animations** : `framer-motion`.
- **Gestion du scroll** : `react-intersection-observer`.
- **Backend / API** : `API Routes Next.js`.
- **Base de données** : `Supabase` (PostgreSQL).
- **Envoi d'e-mails** : `Resend`.
- **Déploiement** : `Vercel`.
- **Versionning** : `GitHub`.

## **6. *Contraintes et Planning (Résumé)***

- **Compatibilité** : Responsive design obligatoire.
- **Performance** : Optimisation des images (`next/image`), gestion fluide des animations, chargement rapide.
- **Accessibilité** : Bon contraste, navigation claire, balises sémantiques.
- **SEO** : Balises meta de base.
- **Déploiement** : Hébergé sur `Vercel` avec liaison à GitHub pour mise à jour continue.