



Prompt (Refonte Rapport Interactif Futuriste)

Contexte : Je dispose d'un rapport interactif web (Next.js + React + TypeScript) pour mon site, qui affiche des informations détaillées sur un bien (ex: véhicule) à partir de données fournies. La version actuelle du rapport reprend un design standard existant, et je souhaite **changer complètement de style** pour quelque chose de beaucoup plus « **WOW** », immersif et **futuriste (style année 2030)**. L'objectif est que l'utilisateur, une fois redirigé vers son rapport, soit **épaté** visuellement et ait envie de naviguer longtemps dedans pour explorer les infos de son véhicule. Le nouveau rapport doit évoquer une **application mobile moderne** ultra soignée (mobile-first), avec des animations fluides et un look & feel high-tech.

Technologies & Contraintes :

- Utiliser **React 18+** (framework Next.js) et **TypeScript**.
- Mise en page **responsive mobile-first** (optimisée pour mobile, tablette, desktop).
- Utiliser **Tailwind CSS** (pour rester cohérent avec le reste du site) pour le style, éventuellement couplé à des effets CSS avancés (ex: `backdrop-filter` pour du glassmorphism).
- Conserver/adapter les dépendances utiles déjà en place : par ex. **Framer Motion** (animations) ¹, **Recharts** (graphiques), **Radix UI** (composants d'interface), **lucide-react** (icônes). Tu peux en ajouter d'autres si nécessaire pour le thème futuriste, en privilégiant des assets open-source trouvables sur GitHub (par ex. bibliothèques UI au style sci-fi comme *Arwes* ² ou *Cosmic UI* ³ pour inspiration). Pas de builder de site ou d'assets propriétaires payants – uniquement du **codage pur et propre**.

Exigences de Design (Rapport « WOW » futuriste) :

- **Look & Feel Futuriste** : Le design doit évoquer l'an 2030 — **moderne, épuré mais avec des touches high-tech**. Par exemple, utilisation de **néons** ou reflets futuristes, de la **glasmorphism** (panneaux semi-transparents floutés) ou **neomorphism**, des effets de **lumières dynamiques** et d'animations subtiles pour donner vie au rapport. Palette de couleurs dans des tons technologiques (bleu électrique, violet vibrant, noir/gris futuriste, vert cyberpunk, etc.) plutôt qu'un simple thème sombre générique. (Éviter le thème « nuit claire » basique ; il faut de l'originalité visuelle pour impressionner.)
- **Immersion & Interactivité** : Intégrer des **animations fluides** et réactions au scroll ou aux interactions utilisateur pour encourager à “pianoter” dans le rapport. Par ex. transitions en fondu/défilement entre sections, éléments qui se révèlent au scroll, feedback visuel au toucher sur mobile (effet de **ripple** ou **haptic feedback** simulé). Si pertinent, incorporer un **fond animé** discret (ex: légère animation géométrique ou halo lumineux) pour donner de la profondeur sans distraire de la lecture.
- **Structure type App Mobile** : Concevoir la mise en page comme une application mobile. Par ex. un **en-tête fixe** avec le titre du rapport et actions clés, une **navigation par onglets ou sections** bien visible (éventuellement un menu burger ou des icônes pour sections sur mobile), et un **bouton flottant** ou footer fixe pour les actions principales (télécharger PDF, partager, revenir à l'accueil – icônes Material/Lucide appropriées). Utiliser les **safe areas** pour les bords d'écran sur mobile. Les sections du rapport doivent être facilement **swipeables** sur mobile (ex: geste de balayage gauche/droite pour changer de section, en plus des onglets) pour renforcer l'expérience applicative.
- **Typographie & Iconographie** : Utiliser une police moderne, lisible et élégante. Les titres/valeurs importantes peuvent avoir un effet de texte stylé (p.ex. un **dégradé néon** ou un léger glow). Inclure des **icônes futuristes** pour chaque section ou type d'info (réutiliser lucide-react ou autres icônes open-source avec un style cohérent). Par ex. une icône de voiture stylisée pour la section véhicule, un bouclier high-tech pour la section sécurité, etc.
- **Mise en page dynamique** : Plutôt que de simples listes à puces statiques, présenter les informations sous forme de **cartes interactives, graphiques et indicateurs visuels** : par exemple, une **gauge**

circulaire animée pour le score de fiabilité du véhicule (avec effets de glow autour du pourcentage)
4, un **radar chart** stylisé pour les critères d'évaluation, une **timeline horizontale** pour l'historique des propriétaires (avec une frise chronologique animée), etc. Chaque section du rapport devrait avoir un **visuel fort** (icône, graphique ou illustration) en plus du texte, afin de captiver le regard.

- **Utilisation maximale des données** : Si de nombreuses données sont disponibles, les exploiter **toutes de manière intelligente** pour enrichir visuellement le rapport. Par exemple:

- S'il y a une **photo du véhicule**, l'afficher en haut (style hero) avec éventuellement un effet parallax ou 3D léger (ex: le visuel bouge légèrement avec le gyroscope).
- S'il y a des **analyses IA** (synthèse, technique, environnement, etc.), les présenter sous forme de **capsules ou encarts futuristes** (ex: une carte avec icône IA ou un robot) incluant un résumé texte généré par l'IA, dans un style valorisant (cadre en surbrillance, fond légèrement animé).
- Pour les données chiffrées (kilométrage, puissance, émissions...), utiliser des **barres de progression stylées, des graphiques ou des badges** plutôt que du texte brut. Par ex. un badge "⚡ 90 CV" avec une couleur ou icône adaptée.
- Inclure la **vignette environnementale** Crit'Air sous forme d'icône ou image stylisée (e.g. petit médaillon coloré) automatiquement, et la **logo de la marque** du véhicule en haut à côté du nom (récupéré via l'URL GitHub comme actuellement).
- **Gestion des données manquantes** : Le rapport doit rester bluffant même si certaines infos manquent. Prévoir des **éléments de remplacement** créatifs lorsque des données sont absentes : par ex. afficher une silhouette de voiture stylisée en lieu et place de la photo manquante, remplacer un graphique vide par une illustration ou un message suggérant que l'info est en cours d'analyse, etc. L'idée est de **ne jamais avoir de "trou vide"** dans la mise en page : chaque section doit paraître complète et engageante, quitte à montrer des valeurs par défaut ou des conseils, pour **faire rêver l'utilisateur** et lui donner du contenu à se mettre sous la dent même si les données réelles sont partielles.

❖ Consignes de Réalisation (Technique) :

- Fournis le **code complet React/TypeScript** du nouveau composant de rapport interactif (par exemple un composant `FuturisticReportView.tsx` ou similaire) ainsi que la page Next.js l'utilisant, avec tous les styles intégrés (utilise Tailwind CSS pour les classes utilitaires). Inclure aussi tout sous-composant nécessaire (par ex. un composant séparé pour un graphique spécifique ou un bouton de partage s'il le faut).
- **Respecte la structure des données existante** pour alimenter le rapport (les données proviennent d'une API `/api/report/interactive/[orderId]` renvoyant `sections`, `vehicleInfo`, `pdfUrl`, etc. comme actuellement). Le nouveau composant doit accepter les props (`sections`, `vehicleInfo`, `ai`, `reportId`, `pdfUrl`...) et les utiliser pour afficher les infos, comme dans l'implémentation précédente, mais en appliquant le nouveau design. **Important** : ne pas enlever de fonctionnalités existantes (ex: bouton de partage, téléchargement PDF 5, etc.), simplement les ré-intégrer dans le nouveau design (éventuellement sous forme d'icônes dans un menu flottant, etc.).
- **Qualité du code** : Le code doit être propre, bien organisé et **commenté en français** pour expliquer les choix importants. N'hésite pas à découper en fonctions utilitaires ou composants plus petits si besoin pour la clarté. Pas de contenu "lorem ipsum" ou de design généré aléatoirement sans cohérence - tout doit sembler fait sur mesure.
- **Originalité** : Bien que tu puisses t'inspirer de composants existants ou de frameworks futuristes, le rendu doit être **unique et adapté** à mon cas d'usage. Par exemple, tu peux t'inspirer de l'esthétique "sci-fi" de frameworks comme Arwes ou Cosmic UI, mais il faut intégrer ça de manière subtile et élégante dans le contexte d'un rapport de véhicule (donc garder une certaine lisibilité et sérieux, tout en étant spectaculaire).
- **Pas de régression** : Assure-toi que le rapport reste entièrement fonctionnel. Teste avec différents scénarios de données (complètes, partielles) pour vérifier l'affichage. Le design doit être compatible **dark mode / light mode** éventuellement, ou au minimum avoir assez de contraste pour être lisible en toutes circonstances 6.

En sortie : Retourne le code React/TypeScript **complet** du nouveau rapport interactif (composant + styles + éventuellement extrait de la page Next.js qui l'utilise). Veille à bien formatter le code (indentation, etc.) et à inclure des commentaires explicatifs pour que je puisse comprendre et intégrer facilement. Aucune étape ne doit être omise. Merci !

2 3

1 4 5 RAPPORT-INTERACTIF-FINAL.md

<https://github.com/silencieux99/verifiemavoiture/blob/5b25bd266c6ed766f717922e74cd439fe2dd1500/RAPPORT-INTERACTIF-FINAL.md>

2 GitHub - arwes/arwes: Futuristic Sci-Fi UI Web Framework.

<https://github.com/arwes/arwes>

3 GitHub - rizkimuhammada/cosmic-ui: A collection of Sci-Fi themed components and futuristic design elements for modern web applications.

<https://github.com/rizkimuhammada/cosmic-ui>

6 TODO-RAPPORT-INTERACTIF.md

<https://github.com/silencieux99/verifiemavoiture/blob/5b25bd266c6ed766f717922e74cd439fe2dd1500/TODO-RAPPORT-INTERACTIF.md>