

Analyse empreinte carbone / Tests écologiques

Résultats obtenus (avant optimisation)



- **Note carbone** : F
- **Émissions par visite** : 1,42 g de CO₂
- **Performance** : Plus polluant que 83% des sites web testés
- **Type d'énergie utilisé** : Énergie standard (non verte)

EcoIndex A

EcoIndex	Eau (cl)	GES (gCO2e)
89.52	1.81	1.21

Nombre de requêtes	Taille de la page (Ko)	Taille du DOM
2	1 (11)	180

Bonnes pratiques

Ajouter des expires ou cache-control headers	✓	
Compresser les ressources (>= 95%)	✓	
Limiter le nombre de domaines (<6)	✓	1 domaine(s) trouvé(s)
Ne pas retailer les images dans le navigateur	✗	5 image(s) retaillée(s) dans le navigateur
Externaliser les css et les js	✓	Pas de inline stylesheet ou inline javascript
Eviter les requêtes en erreur	✓	0 erreur(s) HTTP
Limiter le nombre de requêtes HTTP (<27)	✓	2 requête(s) HTTP
Ne télécharger pas des images inutilement	✗	2 image(s) téléchargée(s) mais non affichée(s) dans la page
Valider le javascript	✓	Javascript validé
Taille maximum des cookies par domaine(<512 Octets)	✓	Pas de cookies
Minifier les css et js	✗	5/5 css or js non minifiés
Pas de cookie pour les ressources statiques	✓	Aucun cookie
Eviter les redirections	✓	0 redirection(s)
Optimiser les images bitmap	✓	Pas d'images bitmap à optimiser
Optimiser les images svg	✓	Pas de svg à optimiser
Ne pas utiliser de plugins	✓	Aucun plugin
Fournir une print css	✗	Pas de print css
N'utilisez pas les boutons standards des réseaux sociaux	✓	Pas de bouton standard de réseau social trouvé
Limiter le nombre de fichiers css (<=10)	✓	Pas plus de 7 fichiers css
Privilégier HTTP/2 à HTTP/1	✓	0/2 ressources utilisant HTTP/1
Utiliser des polices de caractères standards	✓	Pas de polices de caractères spécifiques

Consommations identifiées

Les éléments identifiés comme responsables de cette empreinte élevée sont :

- **Poids des pages important** : plusieurs images en haute résolution (PNG non compressés)
- **Nombre de requêtes HTTP élevé** : chargement de plusieurs ressources (CSS, JS, images)
- **Scripts JavaScript** : certains scripts lourds, notamment pour la gestion dynamique du carrousel et de l'interface utilisateur
- **Base de données et back-end** : non hébergés sur un serveur écologique

Analyse de l'impact de la base de données

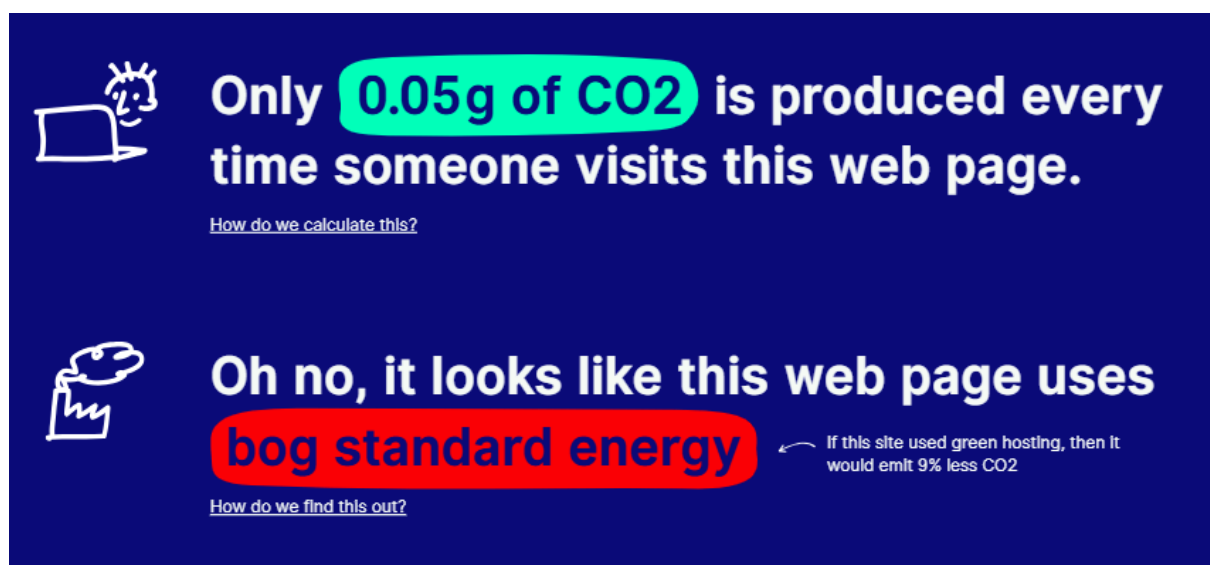
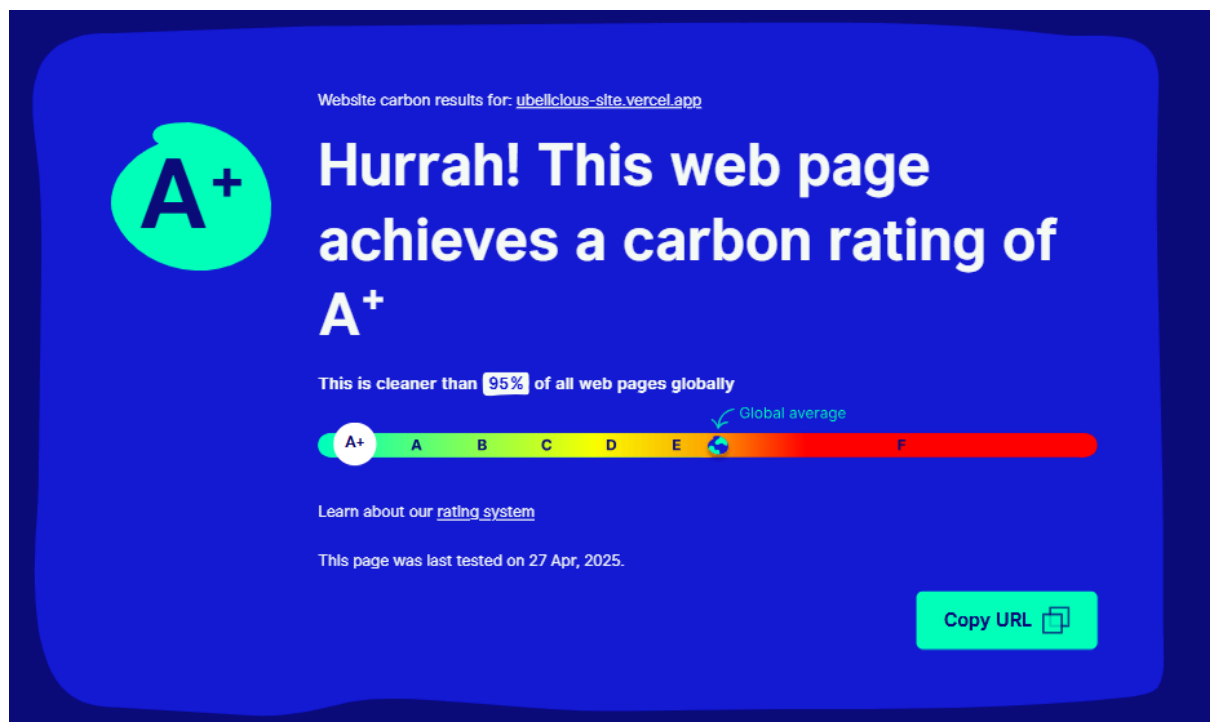
Notre base de données SQLite est légère, mais l'accès en lecture/écriture pour la récupération des recettes et des recommandations génère tout de même un trafic serveur supplémentaire. Le back-end utilisé (Node.js + SQLite) n'est pas particulièrement optimisé pour l'économie d'énergie dans un déploiement sur Render.

Solutions proposées pour réduire l'empreinte carbone

Pour réduire l'impact environnemental, nous avons identifié plusieurs axes d'amélioration :

- **Compression des images** : Compression des PNG et les convertir en webp
- **Réduction du poids des pages** : Optimisation du CSS et du JavaScript (minification, suppression du code inutile)
- **Amélioration du serveur** : Recherche d'une alternative plus verte à Render, car Render ne propose pas d'hébergement certifié écologique

Résultats obtenus (après optimisation)



EcoIndex A

EcoIndex	Eau (cl)	GES (gCO2e)
90.98	1.77	1.18
Nombre de requêtes	Taille de la page (Ko)	Taille du DOM
0	0	204

Bonnes pratiques

Ajouter des expires ou cache-control headers	✓	
Compresser les ressources (>= 95%)	✓	
Limiter le nombre de domaines (<6)	✓	
Ne pas retailler les images dans le navigateur	✗	6 image(s) retaillée(s) dans le navigateur
Externaliser les css et les js	✓	Pas de inline stylesheet ou inline javascript
Eviter les requêtes en erreur	✓	
Limiter le nombre de requêtes HTTP (<27)	✓	
Ne télécharger pas des images inutilement	✓	0 image(s) téléchargée(s) mais non affichée(s) dans la page
Valider le javascript	✓	Javascript validé
Taille maximum des cookies par domaine(<512 Octets)	✓	Pas de cookies
Minifier les css et js	✗	1/5 css or js non minifiés
Pas de cookie pour les ressources statiques	✓	Aucun cookie
Eviter les redirections	✓	
Optimiser les images bitmap	✓	Pas d'images bitmap à optimiser
Optimiser les images svg	✓	Pas de svg à optimiser
Ne pas utiliser de plugins	✓	Aucun plugin
Fournir une print css	✗	Pas de print css
N'utilisez pas les boutons standards des réseaux sociaux	✓	Pas de bouton standard de réseau social trouvé
Limiter le nombre de fichiers css (<=10)	✓	Pas plus de 7 fichiers css
Privilégier HTTP/2 à HTTP/1	✓	
Utiliser des polices de caractères standards	✓	Pas de polices de caractères spécifiques

Avant optimisation, notre site obtenait une note carbone F avec 1,42 g de CO₂ par visite. Cette empreinte élevée s'expliquait par le poids important des images non compressées et par le choix initial d'un hébergeur non certifié écologique, ce qui a fortement impacté le score global, indépendamment du contenu du site.

Notre site est désormais plus propre que 95% des sites web testés mondialement, atteignant une note A+ sur Website Carbon. Cela a été obtenu grâce à l'optimisation des images (conversion WebP), la minification des fichiers CSS/JS/HTML, le nettoyage du code et une migration vers un hébergeur plus performant Vercel.

Déploiement

Le serveur Node.js ainsi que la base SQLite ont été déployés sur la plateforme Render. Il nous avait initialement été recommandé d'utiliser Vercel pour héberger le site. Cependant, Vercel est principalement destiné au déploiement d'applications front-end statiques, et ne prend pas en charge directement l'hébergement de serveurs Node.js avec base de données locale. C'est pourquoi nous avons choisi Render, qui permet l'hébergement complet du back-end (API Node.js + base SQLite). Toutefois, nous avons identifié que Render ne propose pas d'hébergement certifié écologique, ce qui constitue un axe d'amélioration pour notre projet.

Lors du déploiement final, nous avons rencontré une problématique liée à notre choix initial de base de données : SQLite.

En effet, Vercel, plateforme sur laquelle nous avons optimisé et déployé notre site statique pour réduire son empreinte carbone, ne supporte pas la persistance d'une base de données locale comme SQLite. Vercel est conçu pour accueillir principalement des projets statiques ou des APIs serverless, et ne fournit pas d'environnement persistant pour héberger une base de données de ce type.

Cela a engendré une situation particulière, le site vitrine (HTML, CSS, JavaScript minifiés et optimisés) fonctionne parfaitement sur Vercel. En revanche, l'accès à la base de données pour la gestion dynamique des recettes et des recommandations n'est pas fonctionnel sur cette plateforme.

Étant donné le temps limité avant la date de rendu et la soutenance, nous n'avons pas pu adapter toute notre architecture vers une base de données compatible (comme PostgreSQL).

Nous avons donc choisi de maintenir :

- Le site vitrine optimisé sur Vercel, pour respecter notre objectif principal de réduire l'impact environnemental du projet.
- Une version fonctionnelle sur Render, où toutes les fonctionnalités dynamiques liées à la base de données (ajout, suppression, modification) sont pleinement opérationnelles.

Ce choix réfléchi permet de démontrer à la fois :

- L'optimisation écologique du projet (noté A+ sur Website Carbon Calculator),
- La bonne réalisation technique de l'ensemble des fonctionnalités prévues.

lien Render : <https://www.websitecarbon.com/website/greenit-hryk-onrender-com/>

lien Vercel : <https://ubelicious-site.vercel.app/>