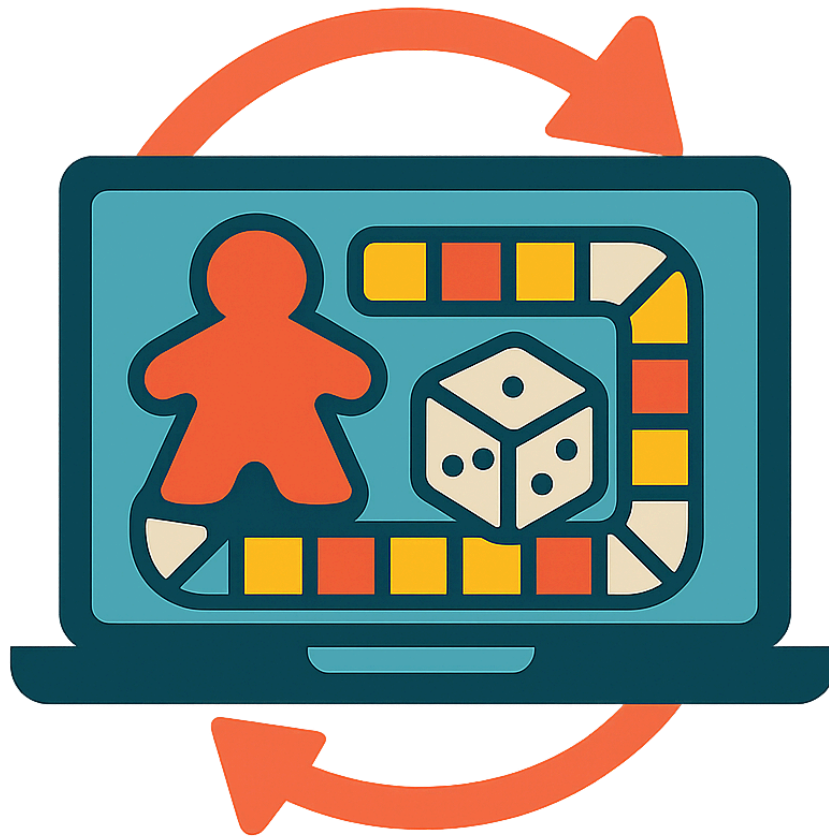


MUNIER Alexandre
ASSOUAD Adrien
CHAN PENG Julien
BIAL Thibault
NGANGO KOLOKO Catherine Lyne

Rent-a-game

Projet GreenIT



Site : <https://rent-a-game-lac.vercel.app/>

Github : <https://github.com/AdrienAssd/RentAGame>

Présentation du site :

Le site “Rent-a-game” est un site dédié à la location de jeux de sociétés. Il va permettre aux différents utilisateurs de louer des jeux de société à partir d’une base de données fournie. Son objectif est de permettre aux utilisateurs d’essayer différents jeux sans engagement financier important.

Public cible :

Bien que certains jeux soient tout public, la bonne utilisation du site nécessite d’avoir une carte de crédit pour louer les jeux, ainsi le public cible est majoritairement les jeunes adultes, ce qui leur permettra de jouer entre amis sans investissements coûteux, ou les familles, pour jouer avec les plus petits sans accumuler les boîtes...

Fonctionnalités du site :

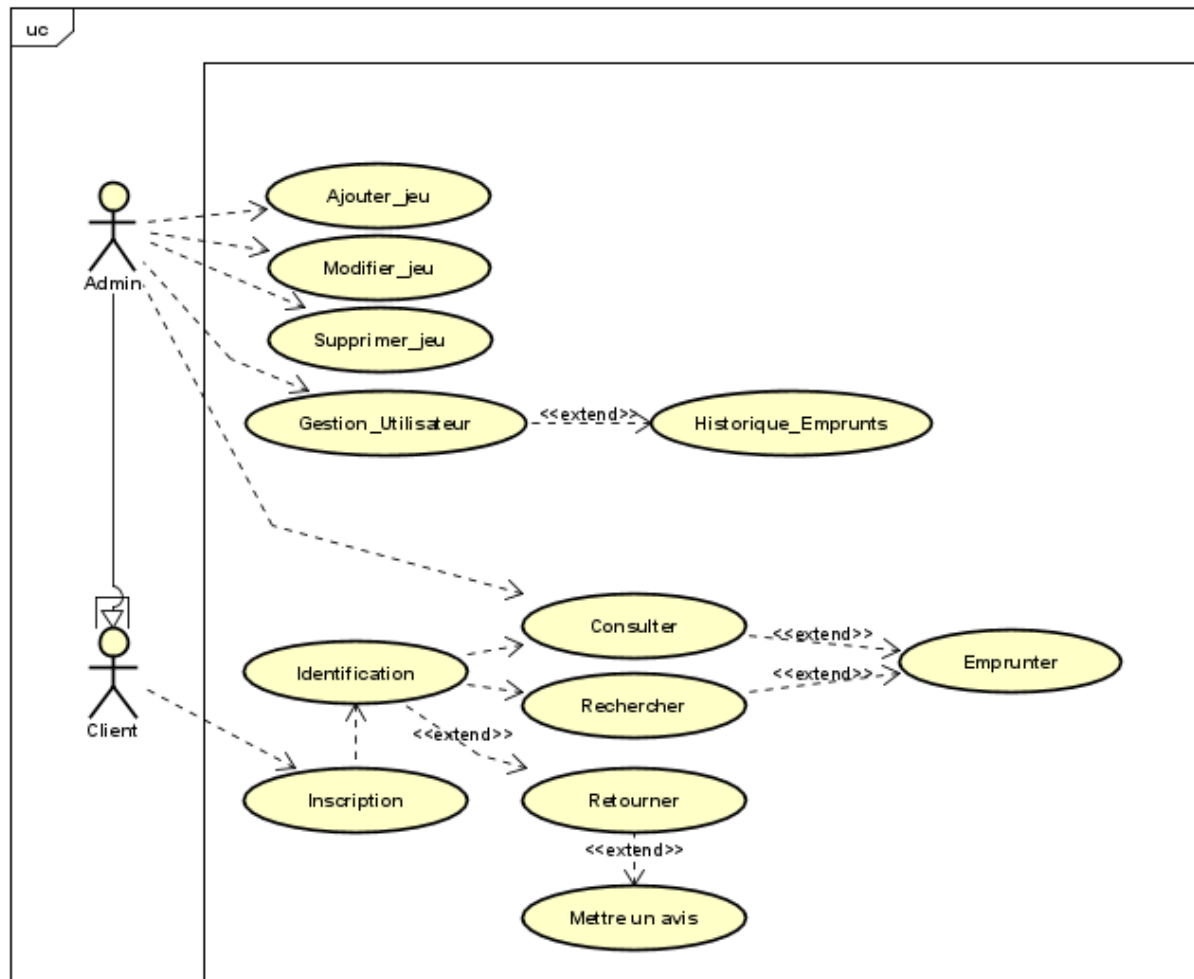
Le site va d’abord afficher une page d’accueil avec des jeux recommandés. Le site offrira aussi une liste des jeux qui sont éligible à la location.

L’utilisateur pourra se créer un profil, ou se connecter au site à l’aide d’une adresse mail. Une fois sur le site, il pourra donc, depuis la page d’accueil ou depuis la page de liste des jeux, consulter un jeu.

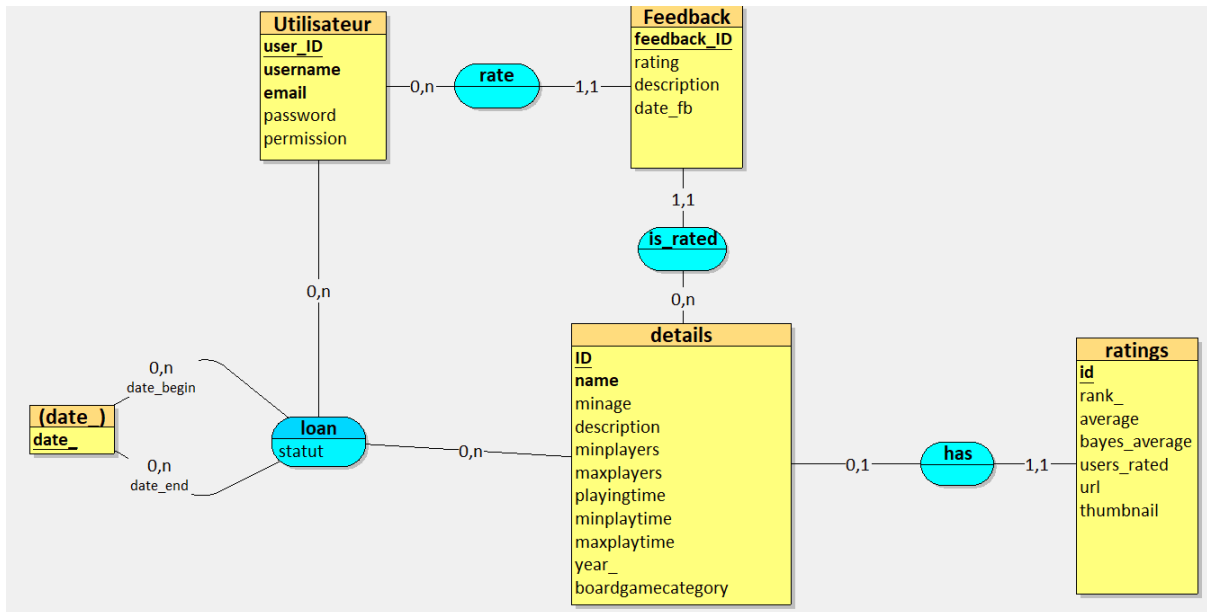
Sur la page du jeu, il pourra consulter :

- Des images du jeu
- La description du produit
- Les avis des autres utilisateurs sur ce produit
- Louer le jeu

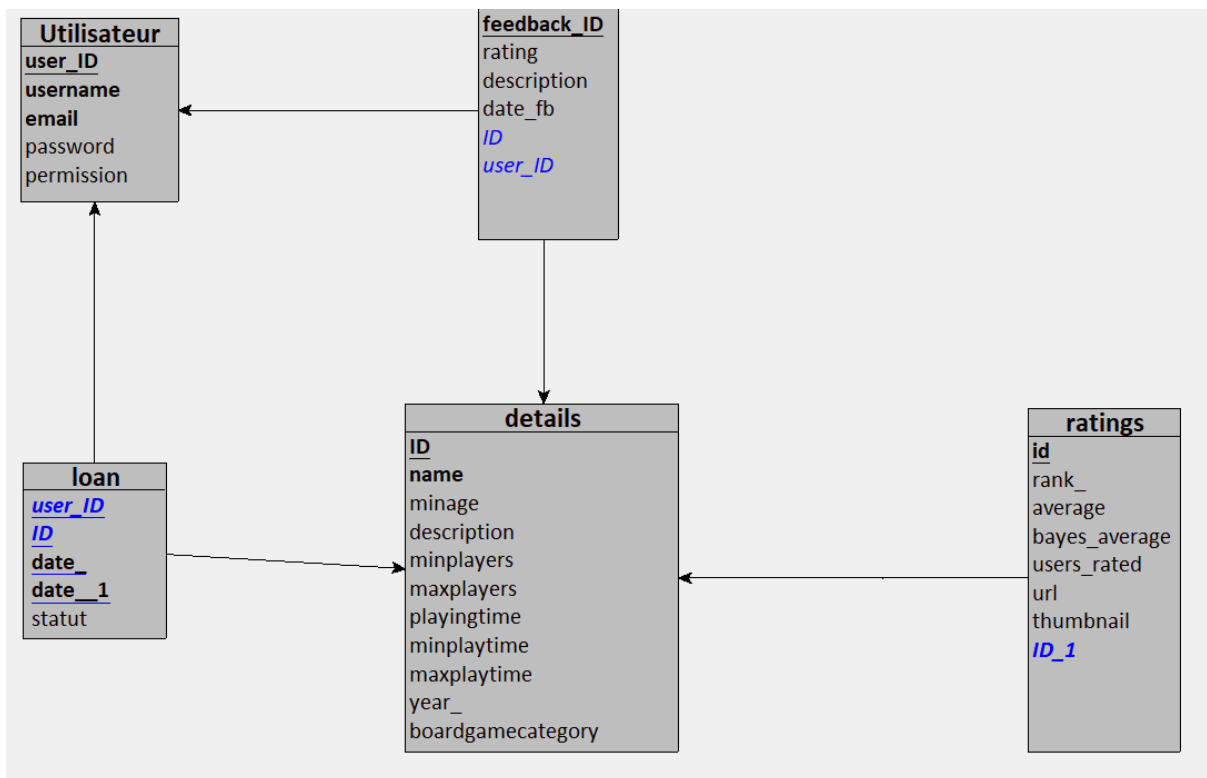
Diagramme de cas d'utilisation :



Modèle Conceptuel de Données :



Modèle Logique de Données (tables avec attributs, clés et relations)



Mockup du projet : [ici](#)

Nos choix techniques pour réduire l'empreinte carbone :

Frontend :

- Astro : génère du HTML statique par défaut (zéro JavaScript envoyé au client sauf si nécessaire), réduisant la bande passante et la consommation énergétique.
- JavaScript minimal (Astro permet du rendu partiel côté client), ce qui allège les calculs côté utilisateur.

Back-end :

- Node.js est léger, efficace en I/O, utilise peu de ressources serveur comparé à certains autres frameworks.
- Express permet une API minimaliste sans surcharge logicielle.
- CORS (Cross-Origin Resource Sharing) optimise les échanges entre services sans redondance.

Lazy Loading pour les images : Charge uniquement les images visibles dans le viewport : moins de requêtes réseau, bande passante réduite, CPU moins sollicité.

Zéro animations et transitions sur le site : Évite les calculs GPU inutiles → diminue la consommation électrique du terminal utilisateur.

Création des pages lorsque l'utilisateur clique sur le lien amenant à celle-ci : Évite le pré-rendu inutile de pages non visitées, ce qui diminue la charge serveur et réduit le stockage/cache utilisé.

Affichage de certains jeux puis "charger plus" : Moins de données à charger d'un coup → réseau et mémoire allégés.

Compression des images avec Squoosh

Hébergeur Vercel + Railway :

Ces plateformes déploient dynamiquement à la demande avec du scaling automatique, évitant des serveurs qui tournent à vide.

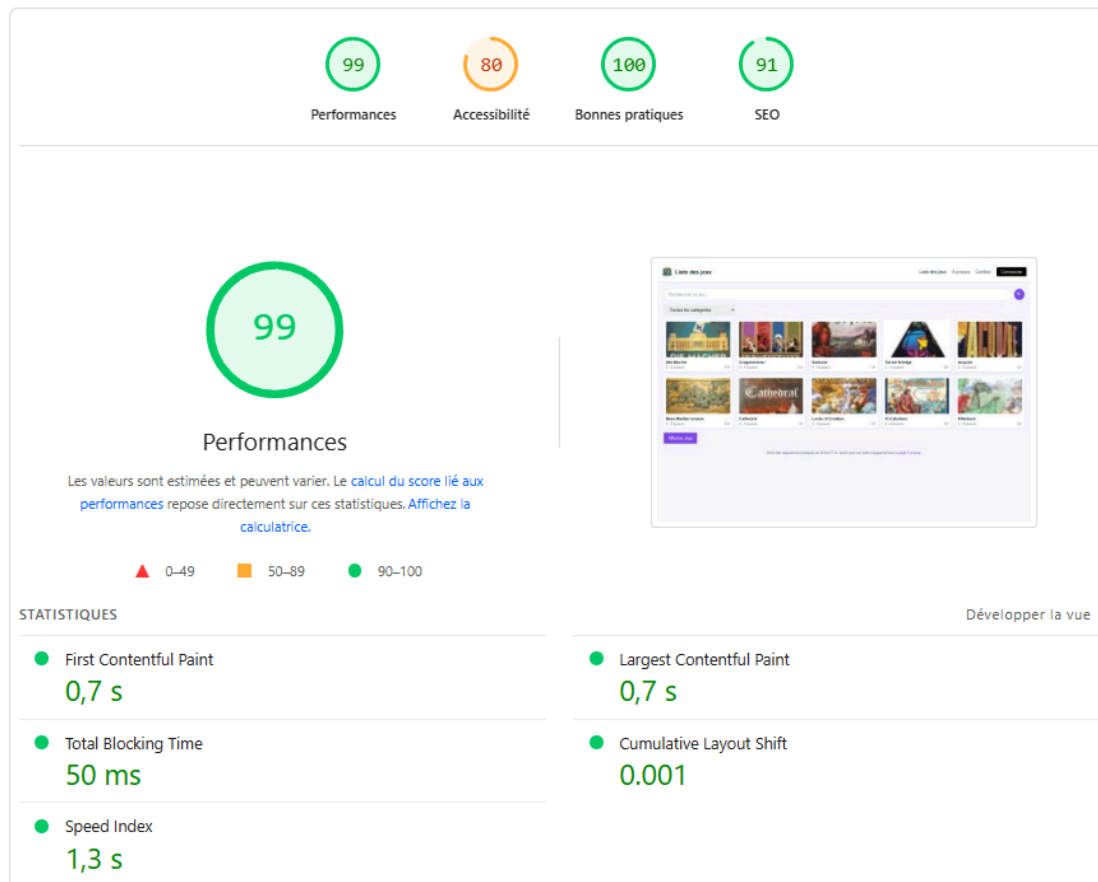
CDN intégré : les ressources sont servies rapidement depuis des nœuds proches des utilisateurs → baisse de la latence et de l'énergie utilisée.

Organisation de la base de données :

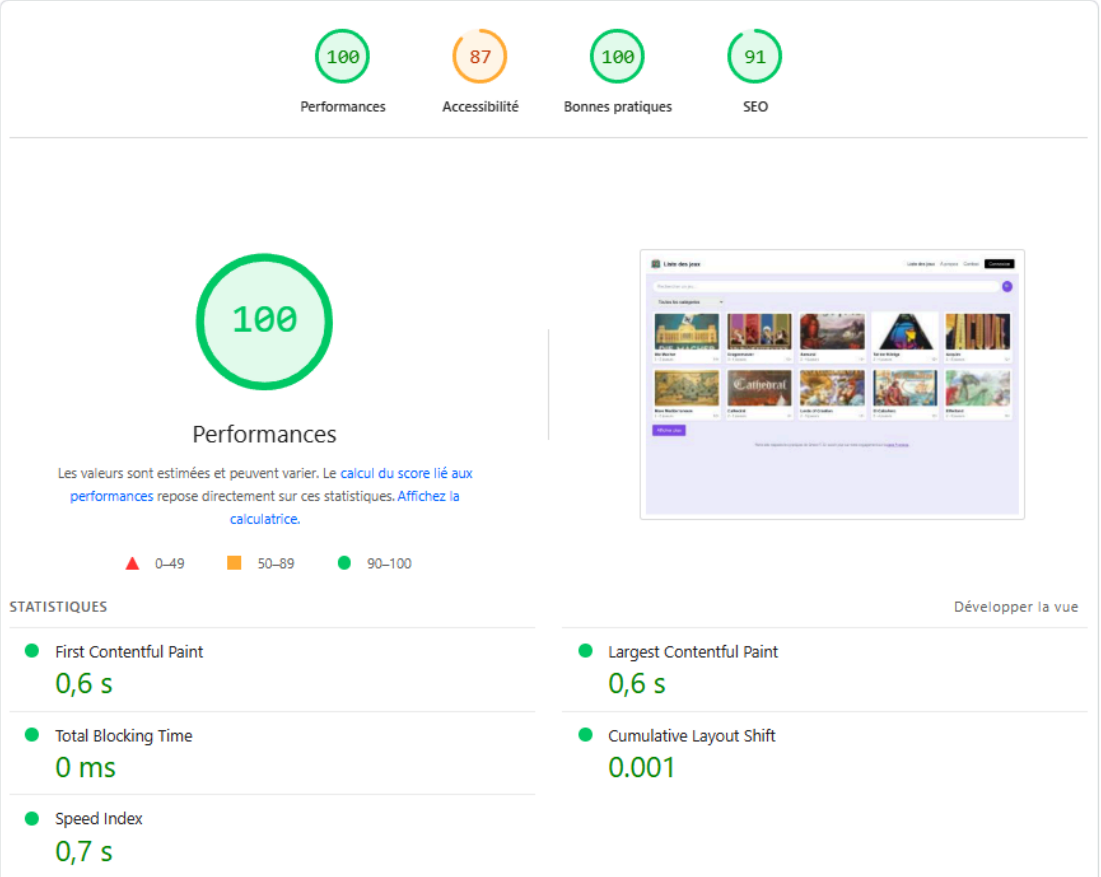
- Une base de données principale répertoriant les jeux de sociétés ainsi que leurs détails (table "details")
- Une base de données secondaire reliée à la table details concernant les notations des jeux
- Une base de données Utilisateur pour gérer les utilisateurs
- Ces derniers peuvent laisser des commentaires sur des jeux grâce à la table feedback
- Une table loan pour gérer les emprunts des jeux

Rapport d'analyse :

Tout d'abord nous nous devons d'analyser les performances de notre site avant l'optimisation de celui-ci (test Lighthouse) :



Voici les résultats du test Lighthouse après les dernières optimisations :



On remarque encore quelques difficultés au niveau de l'accessibilité et cela est en partie dû au choix de l'hébergeur.

Mais nous devons aussi réaliser des tests écologiques :

- Tout d'abord, grâce à l'extension GreenIT qui donne l'écoindex de chaque page :
Page d'accueil :

EcoIndex B		
EcoIndex	Eau (cl)	GES (gCO2e)
75.22	2.24	1.50
Nombre de requêtes	Taille de la page (Ko)	Taille du DOM
15	2971 (29)	197

Page d'un jeu :

EcoIndex A		
EcoIndex	Eau (cl)	GES (gCO2e)
81.86	2.04	1.36
Nombre de requêtes	Taille de la page (Ko)	Taille du DOM
7	2940 (28)	55






Page du profil utilisateur :

EcoIndex A		
EcoIndex	Eau (cl)	GES (gCO2e)
82.76	2.02	1.34
Nombre de requêtes	Taille de la page (Ko)	Taille du DOM
6	2937 (26)	46

- Puis le test Ecoping qui nous donne le score écologique global de notre site :

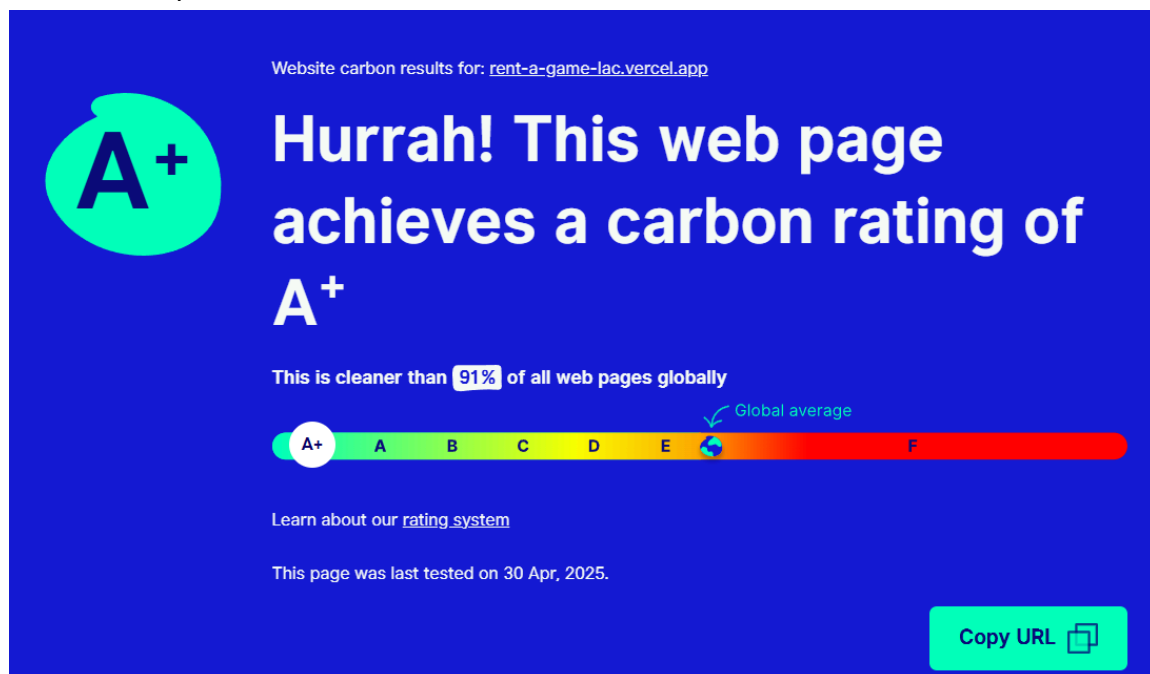
Your Sites

A list of the sites you've added on EcoPing sorted by EcoScore.

Site Name	EcoScore	CO2 (g)	Transfer size	Load time (s)	Actions	Status
 RentAGame	99	-	231.5 KIB	0.83	  	

Previous page 25 Next page

- Et d'après Website Carbon Simulator :



On remarque que notre site se positionne plutôt bien, néanmoins il est toujours possible de faire mieux, notamment en améliorant l'accessibilité de notre site et peut-être un hébergeur payant permettant un rendu plus rapide.

Discussion et conclusion

Ce projet s'inscrit dans une démarche de développement web éco-responsable en s'appuyant sur des outils et pratiques favorisant la sobriété numérique. Le choix d'un framework statique comme Astro, associé à un backend léger (Node.js + Express), permet de limiter le volume de données transférées et de réduire la consommation énergétique côté serveur comme côté client.

Des décisions comme l'absence d'animations et le lazy loading des images renforcent cette logique d'optimisation des ressources. Elles visent à améliorer l'efficacité énergétique sans compromettre l'expérience utilisateur de base.

L'un des enjeux majeurs rencontrés fut de maintenir un bon équilibre entre les fonctionnalités modernes attendues par les utilisateurs et les contraintes de sobriété. Certaines fonctionnalités visuellement attrayantes ont été volontairement laissées de côté afin de respecter cet objectif.