

TI616 : Green IT

Mini-Projet Hackathon : « Le site web le plus Green »

Composition d'équipe de projet : 4 à 5 étudiants par équipe.

Objectif : concevoir, implémenter et analyser l'empreinte carbone d'un site web. Le but est de créer un site web aussi écologique que possible, en minimisant son impact environnemental.

Description : Dans ce projet, il vous est demandé de concevoir, implémenter et analyser l'empreinte carbone d'un site web de votre choix. Vous pouvez donc choisir d'implémenter le site web de vos rêves, celui qui vous rendra riches ou celui le plus drôle, à condition de respecter les conditions suivantes :

- Le site web doit réaliser une véritable fonctionnalité : un site web vraiment utilisable
- Il peut être simple, mais doit être fonctionnel
- Il doit manipuler des données stockées dans une base de données : données de votre choix, mais avec au moins la gestion (création, affichage, modification, suppression) des utilisateurs
- L'interface utilisateur doit être ergonomique, optimisée pour des temps de chargement faibles et minimiser la consommation de ressources.

Notez bien :

- En dehors de **GitHub** (qui sera utilisé pour le suivi de votre projet), vous n'êtes pas obligés d'utiliser les outils ou technologies suggérées. Mais assurez-vous que celles que vous choisissez sont les plus adaptées et les moins énergivores possibles.
- Vous pouvez aussi vous faire aider par ce référentiel pour faire des choix judicieux
<https://www.eco-conception-web.com/conseils-pro-eco-conception-web/>

Phase initiale : brainstorming des idées de sites Web et des fonctionnalités majeures

1. Brainstorming des idées de site web :
 - Quel type de site web (par exemple : e-commerce, vitrine, éducatifs et plateforme d'apprentissage, site de streaming et de divertissement, etc.) ?
 - Qui sont les utilisateurs cibles ?
 - Quels services ou informations seront fournis ?
2. Fonctionnalités du site web
 - Identifiez les fonctionnalités principales de votre site web.
 - Définissez les rôles des utilisateurs (par exemple, administrateur, utilisateur régulier, invité, etc.) et leurs droits.

1^{ère} phase : création et collaboration efficace avec GitHub

1. Création d'un dépôt GitHub commun :
 - Un membre de l'équipe crée un dépôt sur GitHub.
 - Ce dépôt sera le point central pour le suivi du projet, le versionnement du code, la gestion des tâches et la collaboration.
 - Paramétrage du dépôt en privé ou public en fonction de vos préférences
2. Ajout des membres au dépôt :
 - Le propriétaire du dépôt ajoute les autres membres de **l'équipe ainsi que l'enseignant comme collaborateurs**.
 - Chacun doit configurer son environnement local avec Git :
`git clone https://github.com/username/repo-name.git`. Notez que **username** et **repo-name** sont respectivement le nom de chaque collaborateur et le nom du dépôt.
3. Mise en place de la structure du projet :
 - Créer une branche principale (ex : main ou master).
 - Chaque fonctionnalité importante (conception, back-end, front-end, gestion des données, etc.) sera développée dans des branches spécifiques. Par exemple :
 - dev-front pour le développement du front-end.
 - dev-back pour le développement du back-end.
 - database pour la gestion des accès à la base de données.
4. Processus de collaboration et révision :
 - Une fois qu'une fonctionnalité est terminée dans une branche, l'étudiant fait un commit et push sa branche vers le dépôt distant :

```
git add .
git commit -m "Ajout du header au front-end"
git push origin dev-front-header
```
 - Pour intégrer des modifications dans la branche principale, chaque étudiant ouvre un pull request.
 - Les autres membres peuvent examiner le code, ajouter des commentaires et approuver ou demander des modifications avant de fusionner la branche dans main.
5. Vous êtes également libres d'utiliser (ou pas) toute autre fonctionnalité de Git et GitHub.
 - Par exemple chaque étudiant peut travailler sur une tâche distincte dans sa propre branche
 - `git checkout -b dev-front-header`
 - Cela permet à chaque membre de travailler de manière indépendante sur différentes parties du projet.
 - Vous pouvez utiliser la fonctionnalité “GitHub Issues” pour suivre les tâches et les bugs à résoudre.
 - GitHub Projects peut être utilisé pour organiser les tâches en colonnes (To Do, In Progress, Done) et suivre l'avancement.

- Assigner des issues spécifiques à chaque membre et suivre l'avancement.
6. Utilisation de GitHub Actions (optionnel mais conseillé) :
- Configurer GitHub Actions pour automatiser certaines tâches, comme l'exécution des tests ou des outils d'analyse statique (linting).
 - Automatisation des vérifications de performance pour chaque pull request afin de garantir que les optimisations GreenIT sont respectées.

2^{ème} phase : conception

1. Concevoir une architecture minimalistre et éco-responsable :
 - Diagrammes de conception du site web (outil suggéré : <https://plantuml.com/fr/>)
 - Diagramme de classes, pour représenter la structure des objets et des interactions.
 - Diagramme de séquence, pour montrer l'échange de messages entre les composants du site.
 - Un Diagramme de cas utilisation, pour modéliser les interactions entre les acteurs et le système.
 - Wireframes (par exemple, via **Figma**, **Canva**, **Sketch** ou **Balsamiq**).
 - Concevez un mockup de l'interface utilisateur.
 - Incluez des écrans pour les fonctionnalités clés (par exemple, page d'accueil, ...).
 - Préciser les pages à développer, avec au moins :
 - Une page d'accueil informant sur les mesures écologiques du site.
 - Une page de gestion des utilisateurs (ajout, suppression, modification).
 - Une page de présentation des autres données récupérées de la base de données.
 - Etc.
2. Choisir les technologies :
- Technologies de front-end et de back-end en tenant compte de leur consommation de ressources.
 - Favoriser des frameworks légers (ex. Jekyll, Hugo, ou des solutions basées sur HTML/CSS basiques).

3^{ème} phase : implémentation front-end, back-end et accès à la base de données

1^{ère} étape : développement front-end

À obtenir : développer la partie visible du site en intégrant les concepts de green IT, notamment l'optimisation (minification, compression, etc.) des fichiers CSS, JavaScript et des images.

- **Technologies suggérées :**
 - **HTML5/CSS3** pour la structure et le style.

- **JavaScript** pour l'interactivité (utilisation minimale de bibliothèques lourdes comme **jQuery**).
- Framework léger comme **TailwindCSS** ou **Bootstrap** si nécessaire.
- **Consignes clés :**
 - Optimisation du poids des pages web en minimisant les ressources inutiles (minification des fichiers, limitation des images lourdes).
 - Compression des images via des outils comme **TinyPNG** ou **Squoosh**.
 - Utilisation du lazy loading pour charger les éléments (images, vidéos) uniquement quand nécessaire.
 - Limiter l'utilisation des animations et transitions complexes (pour réduire la consommation de CPU).

2^{ème} étape : développement back-end et accès à la base de données

À obtenir : Implémenter la logique côté serveur, avec gestion des données stockées dans une base de données relationnelles.

- **Technologies suggérées :**
 - **Node.js, Django** ou **PHP** pour le back-end.
 - **MySQL, PostgreSQL** ou **SQLite** pour la base de données.
- **Consignes clés :**
 - Mise en place d'une base de données pour gérer les utilisateurs et toute autre donnée pertinente (produits, informations sur la durabilité, etc.)
 - CRUD (Create, Read, Update, Delete) des utilisateurs et des autres entités :
 - Création d'une table utilisateur avec des champs comme nom, email, date d'inscription.
 - Ajout de nouvelles entrées via un formulaire sur le site.
 - Modification ou suppression d'utilisateurs depuis une interface dédiée.
 - Affichage des utilisateurs inscrits via une liste connectée à la base de données.
 - Minimiser les requêtes et optimisations des accès à la base de données (pagination des résultats, requêtes optimisées).
 - Ne récupérer que les données nécessaires à l'affichage.
 - Utiliser des connexions persistantes pour minimiser la consommation d'énergie lors des accès à la base de données.

4^{ème} phase : analyse de l'empreinte carbone

À obtenir : mesurer l'empreinte carbone du site web développé et proposer des solutions pour la réduire.

- **Outils suggérés :**
 - **Website Carbon Calculator** : pour estimer l'empreinte carbone des pages du site.

- **EcoPing, GreenIT Analysis** : pour analyser l'efficacité énergétique du site.
- Tout autre outil vu pendant les TPs et tout autre outil de votre choix
- **Consignes clés :**
 - Identifier les éléments les plus consommateurs en énergie (poids des pages, nombre de requêtes HTTP, scripts JavaScript lourds).
 - Analyser l'impact de la base de données sur la performance énergétique.
 - Proposer et appliquer des solutions pour réduire davantage l'impact environnemental (par exemple, compression plus agressive des ressources, réduction du nombre de scripts ou images).

5^{ème} phase : tests

À obtenir : tester le site pour s'assurer qu'il est fonctionnel et documenter les choix techniques, les bonnes pratiques et les optimisations réalisées.

- **Tests à réaliser :**
 - Tests fonctionnels : Vérifier que les formulaires fonctionnent correctement (CRUD).
 - Tests de performance : Utiliser des outils comme **Lighthouse** pour tester la rapidité de chargement des pages.
 - Tests écologiques : Vérifier à nouveau l'empreinte carbone après optimisations.

6^{ème} phase : livrables finaux et documentation

- **Livrables finaux** : sur votre dépôt GitHub, vous devez :
 - Documenter le projet, c'est-à-dire, maintenir un fichier README.md détaillé avec :
 - Les instructions pour cloner, configurer et exécuter le projet.
 - Les étapes pour contribuer au projet (ex: conventions de commit).
 - Une brève description des différentes parties du site et des fonctionnalités.
 - Pull requests : historique des révisions de code et des contributions de chaque membre
 - **Wireframes** des principales pages du site.
 - **Code source** (front-end et back-end) du site web avec un accès fonctionnel à la base de données.
 - **Rapport d'analyse** détaillant l'empreinte carbone du site avant et après optimisation.
 - **Documentation technique** de l'implémentation.
- **Documentation attendue (rapport au format PDF) :**
 - **Présentation du projet** :
 - Présenter le site web, ses fonctionnalités, et l'approche suivie pour minimiser l'impact environnemental.
 - Les choix techniques réalisés pour réduire l'empreinte carbone.
 - L'architecture de la base de données et du code.

- **Discussion et conclusion**

- Réflexion sur les pratiques éco-responsables dans le développement web.
- Défis rencontrés pour équilibrer les fonctionnalités du site et la réduction de l'empreinte carbone.
- Idées pour l'amélioration future et intégration des principes écologiques dans d'autres projets technologiques.

7^{ème} phase : présentation finale du projet

- Présentation du projet, des objectifs, de la conception, des choix technologiques
- Démonstration en live
- Présentation de l'analyse de l'empreinte carbone
- Réponse aux questions de l'enseignant

À propos de l'évaluation du projet :

1. Conception (30%) :

- Simplicité et efficacité des choix techniques.
- Prise en compte de l'impact environnemental dès la conception.

2. Implémentation et accès à la base de données (37,5%) :

- Qualité du code, respect des pratiques de clean code, et optimisation des requêtes et des ressources.
- Utilisation d'outils pour mesurer et réduire la consommation d'énergie.

3. Analyse de l'empreinte carbone (10%) :

- Exactitude de l'analyse et pertinence des ajustements pour réduire l'empreinte carbone.
- Comparaison avec d'autres sites de même type.

4. Tests & Intégration (10%)

- Mise en place de tests automatisés.
- Détection et correction des bugs

5. Présentation finale (10%) :

- Clarté de la présentation, efficacité des conclusions tirées du projet, et justifications des choix des technologies utilisées

6. Qualité de rédaction du rapport (2,5%)

Il est important de noter que les membres d'un même groupe peuvent avoir des notes différentes. Tout est fonction du degré d'implication de chacun.

Bonus : le meilleur projet, se verra attribuer des points supplémentaires.