

# RAPPORT DESIGN4GREEN TEAM SQUIRREL - TEAM 28 20 NOVEMBRE 2019 - 22 NOVEMBRE 2019

# Squirrel Electrical Management



CAMPAN Nathaniel CHAUMEIL Thomas FIEBER Théo MATHIEN Thomas MOUSSEAU Louise

# Table des matières

1	Contexte	2
2	Choix des technologies  2.1 Langage back-end	2
3	Design         3.1       User expérience : structure du site et ergonomie	
4	Accessibilité	3
5	Sécurité, RGPD et intégrité des données	4
6	Optimisation	4

1. Contexte 2

#### 1 Contexte

Ce rapport s'inscrit dans le contexte du challenge Design4Green. Nous avons créé une équipe, la Team Squirrel, composée de 5 personnes : Nathaniel CAMPAN, Thomas CHAUMEIL, Théo FIEBER, Thomas MATHIEN et Louise MOUSSEAU.

Le sujet proposé était : « : Eco-concevoir numériquement une version simplifiée d'un des outils (analyse de la consommation électrique des ménages) du carnet numérique du logement ».

Ce document a pour but d'expliquer notre stratégie pour répondre au mieux au cahier des charges.

Les objectifs du projet sont de réaliser une solution la moins impactante possible sur l'environnement, tout en restant ergonomique, accessible et respectueuse des données personnelles.

Nous avions carte blanche pour le choix des technologies et disposions d'un serveur pour héberger notre solution.

# 2 Choix des technologies

#### 2.1 Langage back-end

Nous avons identifié deux langages pour réaliser le traitement de données sur le serveur en back-end : Python et PHP.

Après avoir réfléchi sur les avantages et inconvénients de chacun des langages, nous avons décidé de choisir Python.

Le principal point qui nous a décidés sont nos connaissances plus poussées de ce langage qu'en PHP nous permettant ainsi un gain de temps considérable. Le résultat, en python, est d'après nous plus complet et mieux optimisé. [5] [2]

PHP7 est plus rapide que Python3, cependant, pour un volume de données faible nous avons estimé qu'il s'agissait d'un point mineur par rapport à la sécurité et sa facilité d'utilisation.

En effet, le langage python comprend des caractéristiques pouvant répondre à notre besoin : il s'agit d'un langage épuré et léger, avec une sécurité plus simple à implémenter, notamment car le serveur en python protège des injections SQL et facilite l'utilisation de SHA256 pour le hashage des mots de passe.

#### 2.2 Technologies de la base de données

Nous avons dans un premier temps recherché la solution la plus adéquate entre les bases de données relationnelles (SQLite ou encore MySQL) et le NoSQL. Le désavantage des bases de données relationnelles est la difficulté de répartir la charge de requêtes sur plusieurs machines ainsi que la rigidité des structures de données. Pour la base de données fournie, notre schéma est fixe et nous n'avons pas besoin de répartir la charge de requêtes sur plusieurs machines. Nous ne sommes pas impactés par les aspects négatifs des technologies relationnelles, donc l'utilisation d'une technologie NoSQL n'est pas avantageux.[7]

Nous avons ensuite réfléchi à la technologie à utiliser entre SQLite, MySQL ou Post-greSQL. SQLite est clairement la technologie la plus légère avec l'empreinte carbone la

plus faible des trois. MySQL est légèrement plus rapide et plus adaptée aux bases de données de grande taille.

En raison de la taille de notre base de données, nous avons donc opté pour l'utilisation de SQLite.[6][7]

#### 2.3 Langage front-end

Pour la structure et l'interface graphique du site web, nous avons décidé d'utiliser HTML5 et CSS3.

Pour dynamiser notre site web et minimiser le nombre de requêtes vers le serveur, nous avons utilisé JavaScript pour déléguer une partie de la charge serveur sur le client.

# 3 Design

#### 3.1 User expérience : structure du site et ergonomie

La solution consiste à fournir un site web de type administratif de consultation de données. Nous avons donc voulu prioriser un accès rapide et intuitif aux services fournis.

Dans cette optique, le site est structuré avec des pages sobres en faible nombre. Chaque service est clairement identifiable et accessible en un nombre de clics limité.

Les intitulés des pages sont clairement identifiables.

Le haut de page contient deux bandeaux : le premier regroupe le nom du site et le logo, le second regroupe les différentes fonctionnalités pour l'accessibilité ainsi que les liens vers les autres pages.

#### 3.2 Identité visuelle

Nous avons créé une charte graphique pour notre site web. Nous avons utilisé un nuancier de bleu car cette couleur est associée à la fiabilité et au professionnalisme.[4] Cette couleur est en général utilisée pour les sites web des banques, sites dans le domaine de la santé ou encore high-tech.

Nous avons associé à cette couleur un design minimaliste, agréable, intuitif et rapide. Le logo a été créé en respectant cette charte graphique. Il est minimaliste, rapide à afficher et clair. Aucun dégradé de couleur n'est utilisé pour la clarté et minimiser le poids de l'image.

#### 4 Accessibilité

Nous avons rendu notre site accessible aux malvoyants et aux personnes aveugles, l'accessibilité aux malentendants et sourds étant inutile car nous n'avons aucun contenu audio ou vidéo à proposer. Pour cela, nous avons développé 3 points :

- Ajout de balises descriptives si besoin
- Deux thèmes disponibles avec des couleurs contrastées
- Personnalisation de la taille du contenu

Au niveau des graphiques, les balises descriptives ont été ajoutées pour permettre aux personnes aveugles de pouvoir bénéficier de toutes les informations nécessaires.

Les deux thèmes, un thème sombre et un thème normal, ont été développés pour modifier le contraste sur l'ensemble du site. Au niveau du daltonisme, les couleurs utilisées ne devraient pas poser de problème.

Enfin, la personnalisation de la taille du contenu avec des icônes zoom avant et arrière permet un confort de lecture aux personnes malvoyantes.

# 5 Sécurité, RGPD et intégrité des données

Pour respecter la RGPD, nous avons décidé de mettre en place les recommandations du CNIL sur la complexité des mots de passe. De plus, nous avons précisé dans les mentions légales que l'utilisateur détiens tous ses droits sur ses données qu'il peut supprimer ou modifier en s'adressant à l'administrateur.[1]

Les mots de passes sont stockés en hashé dans la base de donnée en utilisant SHA256 pour protéger l'accès aux données des utilisateurs.

La réinitialisation du mot de passe ainsi que la validation du compte après création utilisent la validation par lien temporaire envoyé par mail.

Par mesure de sécurité, nous vérifions également côté client aux données envoyées à l'aide d'expressions régulières pour assurer l'intégrité des données. Côté serveur, nous avons également utilisé des fonctions permettant de se protéger des injections SQL.

Nous avons un certificat SSL auto-signé. Nous n'avons pas pu obtenir de certificat signé à cause d'une restriction sur le nombre de certificats accordés. Le fournisseur de clés SSL let's encrypt à en effet atteint sa limite de certificats pour les noms de domaines ovh.net. Nous avons cependant dû le désactiver car il rendait les services inconsistant en fonction du protocole utilisé entre http et https.

# 6 Optimisation

Afin d'optimiser notre solution, nous avons limité le nombre de requêtes SQL adressées à la base de données ainsi que leur poids.

La durée du cache des fichiers statiques ont été allongées à la durée maximum pour éviter un surplus de requête.

Pour réduire le temps de chargement des pages, nous chargeons les scripts Javascript en asynchrone.

Nous avons également compressé tous les fichiers. De plus, nous avons limité l'utilisation d'images et privilégié l'utilisation de caractère unicode pour l'icône de changement de thème par exemple.

Nous nous sommes basés sur les 115 bonnes pratiques.[3]

RÉFÉRENCES 5

### Références

[1] Authentification par mot de passe : les mesures de sécurité élémentaires.

- [2] Php versus python 3 fastest programs.
- [3] Ecoconception web, les 115 bonnes pratiques Doper son site et réduire son empreinte écologique. Eyrolles, 2019.
- [4] Anthedesign. Comment bien choisir les couleurs de votre site internet?, 23 mars 2018.
- [5] E. Boersma. Php vs python: Which should you choose in 2019?, 25 mai 2019.
- [6] M. Drake. A comparison of nosql database management systems and models, 9 aout 2019.
- [7] M. Drake. Sqlite vs mysql vs postgresql: A comparison of relational database management systems, 19 mars 2019.