

DOCUMENTATION

+

DOCUMENTATION
TECHNIQUE



Pour une communauté Minecraft francophone
sûre et agréable.

Table des matières

Présentation du projet	2
Moodboard	2
Fonctionnalités du site.....	2
Création de compte.....	2
Photo de profil automatique.....	2
Édition du profil utilisateur	2
Dénoncer un utilisateur	3
UUID Finder	3
Image du skin de l'utilisateur dénoncé.....	3
Voir les utilisateurs dénoncés	3
Panel administrateur	3
Mode maintenance	3
Personnalisation de la maintenance	3
Mode sombre et clair	3
Technologies utilisées	3
MVC.....	4
PHP	4
HTML, CSS et JavaScript	5
Base de données.....	5
API Externes	5
Difficultés et solutions	5
Installation	5
Niveaux de permissions	6
Administrateur.....	6
Gérant de serveur	6
Membre	6
Visiteur	6
API Externes	7
Base de données	7
Fonctionnalités futures.....	8
Remerciements.....	9
Crédits	10

Présentation du projet

Je joue à Minecraft depuis 2013, et depuis le premier jour je crée des serveurs pour mes amis ou pour des gens que je ne connais pas. J'ai tout de suite adoré ce jeu et son univers, mais une chose revenait trop régulièrement et me déplaisait vraiment : les mauvais joueurs.

Il existe plusieurs manières de s'en débarrasser, comme le bannissement, le kick, le ban-ip, etc. mais pourquoi ne pas couper le problème à la racine et s'en débarrasser par défaut ?

Snitch répertorie tous les signalements faits par des administrateurs ou modérateurs de serveur dans une base de données et permet aux serveurs s'en servant de chercher si un utilisateur suspect a déjà été dénoncé pour tricherie ou mauvais comportement sur d'autres serveurs.

Notre but : une communauté Minecraft francophone sûre et agréable.

Moodboard

Trois couleurs principales ont été choisies pour ce site. Une couleur sombre (Eerie black), une claire (White smoke) et une principale (Neon blue).

Ces trois couleurs permettent d'avoir des bons contrastes et le bleu néon permet de rester dans un univers jeu vidéo.

La typologie utilisée pour le site web est la typo officielle du jeu Minecraft. Cette police est carrée, comme le jeu et a cet aspect 8Bit qui nous renvoie à l'univers du jeu vidéo.

Fonctionnalités du site

Création de compte

Les utilisateurs peuvent créer un compte puis se connecter.

Photo de profil automatique

La photo de profil de l'utilisateur connecté sera le skin lié au même nom d'utilisateur. Cela permet d'éviter un stockage inutile supplémentaire et d'avoir une photo différente pour tous par défaut.

Édition du profil utilisateur

L'utilisateur a la possibilité de changer de mot de passe, d'adresse mail ou de nom d'utilisateur via une page de profil.

Dénoncer un utilisateur

Une page permettant de signaler un joueur, avec plusieurs paramètres comme le nom, le UUID ou les raisons de cette dénonciation.

Cette page a aussi une fonctionnalité qui permet de trouver un UUID à partir d'un pseudo.

UUID Finder

Fonctionnalité qui permet, grâce à l'API de PlayerDB de trouver un UUID en se basant sur le nom d'utilisateur fourni par le dénonciateur.

Image du skin de l'utilisateur dénoncé

Afin d'illustrer la dénonciation, l'API de Crafatar permettra d'afficher le skin du joueur en volume isométrique directement sur la page de dénonciation et sur la page du post.

Voir les utilisateurs dénoncés

Une liste des joueurs dénoncés, du plus récent au plus ancien.

Panel administrateur

Un panel d'administration réservé aux utilisateurs de niveau 1 (voir plus bas les niveaux de permissions).

Mode maintenance

Possibilité de désactiver les fonctionnalités principales du site et d'afficher un message de maintenance.

Personnalisation de la maintenance

Le message affiché lors de la maintenance peut être personnalisé via le panel administrateur, ainsi que la description. Cela évite un tour dans les fichiers.

Mode sombre et clair

Par défaut le site est affiché en mode clair mais un petit bouton en haut à droite permet de basculer entre le mode sombre et le mode clair. Ce bouton s'adapte au mode activé.

Technologies utilisées

Snitch est un CMS codé en MVC, c'est-à-dire « modèle-vue-contrôleur ».

Les technologies utilisées sont le PHP, HTML, CSS, JavaScript, MVC et les bases de données. Nous avons aussi eu recours à des API externes, comme PlayerDB et Crafatar.

La librairie SweetAlert2 a aussi été utilisée pour faire des messages d'alerte ou de succès.

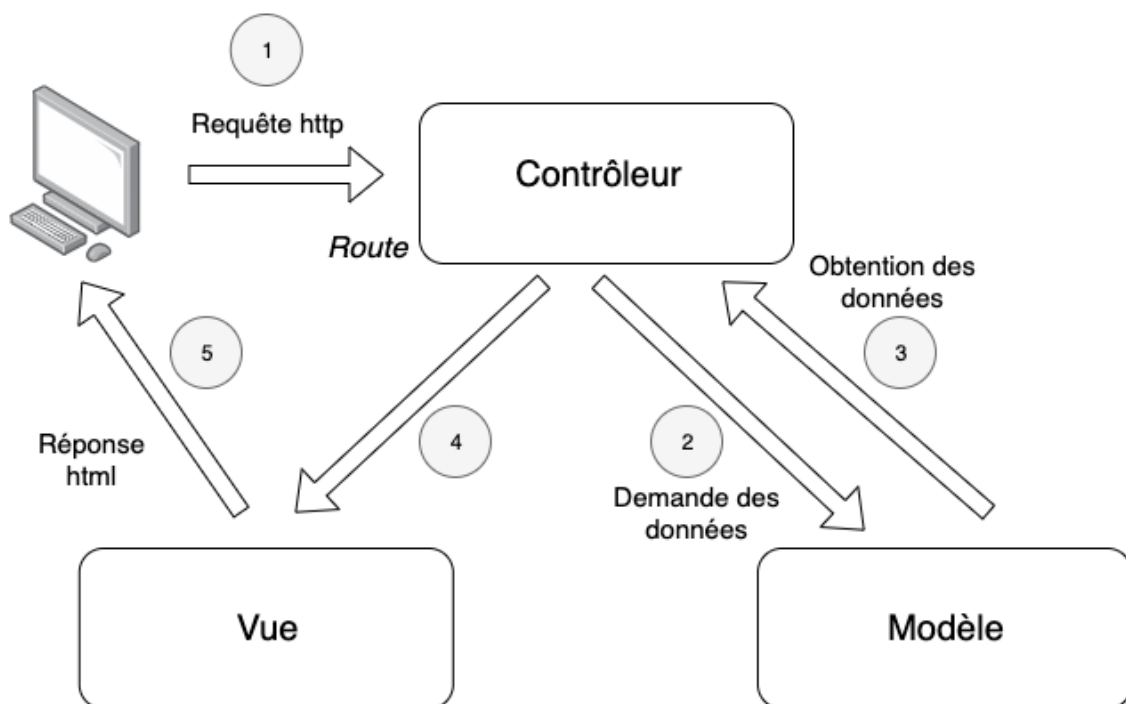
MVC

Le Modèle-Vue-Contrôleur (MVC) est un concept essentiel en développement logiciel, visant à organiser les applications de manière claire et efficace. Il se compose de trois composants principaux :

Le "Modèle" représente les données et la logique métier de l'application. Il traite la gestion des données et les règles de fonctionnement.

La "Vue" est responsable de l'affichage des données au sein de l'interface utilisateur. Elle communique avec le modèle pour récupérer les informations à afficher.

Le "Contrôleur" agit comme un intermédiaire entre le modèle et la vue, traitant les actions de l'utilisateur. Il met à jour le modèle en fonction des interactions de l'utilisateur et rafraîchit la vue en conséquence. Cette séparation des préoccupations facilite la maintenance, la réutilisation du code et permet une meilleure évolutivité de l'application.



PHP

Le site utilise PHP dans sa plus grande partie, pour permettre un dynamisme et l'affichage des données selon des conditions. Cette technologie est considérée comme « back-end », elle fonctionne grâce au serveur.

HTML, CSS et JavaScript

Tout ce qui concerne l’affichage utilisateur. Le javascript a été utilisé pour fonctionner avec les API et les interactions utilisateurs.

Base de données

Notre base de données permet de garder les utilisateurs ayant été dénoncés et d’avoir des membres et des niveaux de permissions pour les dénonciations ou l’affichage des données utilisateurs.

Cette base permet aussi la configuration du site web, notamment le mode maintenance.

API Externes

Pour le bon fonctionnement du site, trois API sont utilisées. La première, PlayerDB permet de trouver le UUID d’un utilisateur lorsqu’il s’agit de le reporter. Cette API permet aussi d’éviter qu’un utilisateur non-existant soit reporter.

La deuxième, MC-Heads permet d’afficher la tête de l’utilisateur connecté, soit dans la barre de navigation soit dans la page profile.

La dernière API est Craftar, qui nous permet d’afficher un skin isométrique lorsque l’utilisateur est trouvé sur PlayerDB avant d’envoyer la dénonciation.

Difficultés et solutions

Il a été difficile pour moi d’intégrer les messages d’erreur et les messages de succès. C’est pourquoi j’ai décidé d’utiliser la librairie SweetAlert2 qui permet d’afficher des alertes en plein écran avec une large liberté de personnalisation des alertes.

Comprendre le MVC n’était pas non plus une partie de plaisir, mais au final c’est une compétence que je suis très content d’avoir. Ma solution a été de ne pas abandonner le MVC et de chercher tant que je pouvais, ça a porté ses fruits.

L’utilisation des bases de données et des requêtes préparées n’était pas non plus le plus facile pour moi, j’ai donc fait appel à un ami qui a su répondre à toutes mes questions.

Installation

Pour installer le CMS sur votre serveur, veuillez suivre ces instructions :

1. Copiez le projet GitHub dans la racine de votre site web.
2. Redirigez votre serveur PHP vers /public au chargement
3. Créez une nouvelle base de données en important le fichier SQL présent dans le projet (/Core/snitch.sql)
4. Modifiez le fichier (/Core/cred_example.php) avec les bonnes données de connexion vers votre base de données puis renommez le fichier credentials.php

Les informations de connexions par défaut sont dans le fichier d’exemple.

Le site fonctionne maintenant avec des utilisateurs factices et des posts factices. Vous pouvez importer une base de données vierge en prenant le fichier snitch_empty.sql au lieu de snitch.sql.

Pour vous connecter aux comptes factices, utilisez les deux premières lettres de chacun de leur email comme mot de passe et le tour est joué.

Après ça vous pouvez supprimer les deux fichiers SQL.

Niveaux de permissions

Il existe 4 niveaux d'utilisateurs qui confèrent certaines permissions ou non.

Administrateur

Le premier niveau est l'administrateur (permission = 1) qui peut tout faire sur le site.

- Panel administrateur
- Bypass maintenance
- Signaler des joueurs
- Voir les signalements
- Gestion du profil
- Accéder au site web

Gérant de serveur

Le deuxième niveau est le Gérant de serveur (permission = 2)

- Signaler des joueurs
- Voir les signalements
- Gestion du profil
- Accéder au site web

Membre

Le troisième niveau est le Membre (permission = 3)

- Voir les signalements
- Gestion du profil
- Accéder au site web

Visiteur

Le dernier niveau est pour les visiteurs du site, non-inscrits

- Accéder au site web

Pour modifier les accès selon les niveaux, vous devez modifier les contrôles effectués dans les Controllers.

API Externes

L'API la plus importante sur Snitch est PlayerDB, elle permet de retrouver le UUID d'un utilisateur lorsque l'on souhaite le dénoncer sans devoir passer par NameMC ou les commandes de notre serveur.

En utilisant JavaScript, lorsque le dénonciateur souhaite trouver le UUID, l'API va être contactée pour mettre en relation le nom d'utilisateur et un UUID.

Si un nom d'utilisateur non-existant est donné, un message d'erreur s'affichera.

Si un nom d'utilisateur pas possible est donné (contrôle regex), un message d'erreur s'affichera.

Cette fonctionnalité ne permet pas de faire l'inverse et de passer d'un UUID à un utilisateur, merci d'utiliser NameMC pour ça, pour le moment.

Nous n'autorisons que la dénonciation de joueurs premium, c'est-à-dire de joueurs ayant un compte minecraft mojang, **pour des raisons évidentes d'usurpation d'identités, ne dénoncez pas des joueurs en version crack.**

Base de données

La base de données à 4 tables : users, reported_users, posts et settings.

La première table « users » permet de stocker les utilisateurs inscrits sur le site.

La seconde table « reported_users » permet de stocker tous les utilisateurs ayant été dénoncés au moins une fois sur le site et de stocker leur dernier nom d'utilisateur connu, afin d'éviter de trop contacter PlayerDB. Si un utilisateur est dénoncé une seconde fois et que son nom d'utilisateur a changé, il sera actualisé dans la base de données sans créer de nouvelle colonne.

La troisième table « posts » permet de stocker toutes les dénonciations faites sur le site.

La dernière table « settings » permet de stocker les éléments de modification du site web.

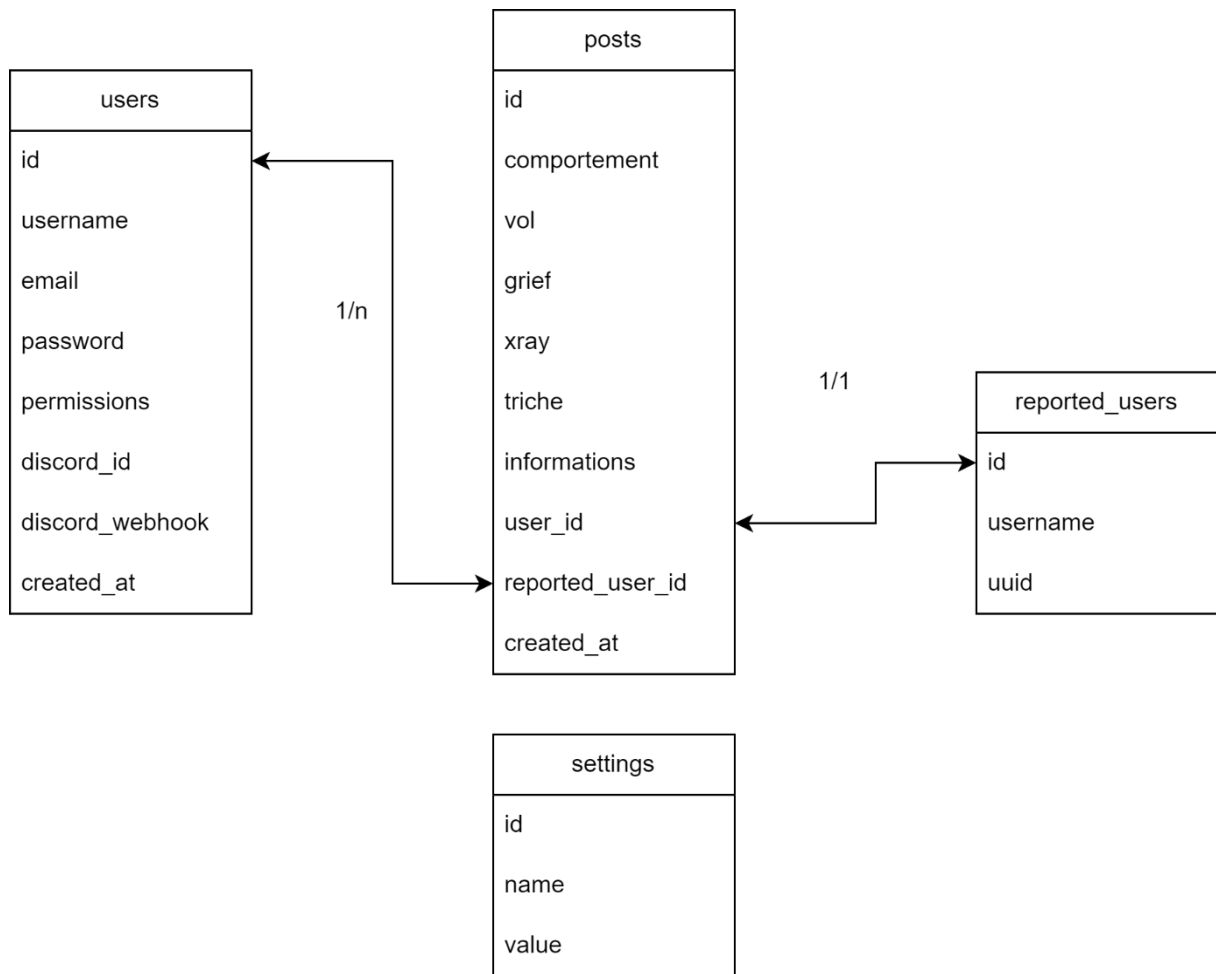


Figure 1: Schéma de la base de données de Snitch

Fonctionnalités futures

- Trouver un pseudo avec le UUID
- Plugin minecraft pour les serveurs
 - o Permet de notifier un utilisateur lorsqu'un joueur déjà dénoncé rejoint le serveur
 - o Permet de dénoncer un joueur directement depuis le serveur
 - o Permissions avancées
- Fonction de recherche sur le site, pour trouver un utilisateur dénoncé
- Filtres pour voir les utilisateurs dénoncés selon certaines conditions
- Formulaire de réinitialisation de mot de passe
- Validation par courriel lors de l'inscription

Remerciements

Je remercie chaleureusement mes professeurs, qui m'ont donné la possibilité de réaliser ce site web et de développer cette large palette de nouvelles compétences.

Je remercie particulièrement Jenna L. pour sa flexibilité et son approche humaine face aux situations plus complexes.

Je remercie particulièrement Julien R. qui nous a donné avec énormément de détails les outils nécessaires pour aller plus loin dans ce projet.

Merci à la SAE de Genève.

Crédits

- SweetAlert2 <https://github.com/sweetalert2/sweetalert2>
- PlayerDB <https://playerdb.co/>
- MC-Heads <https://mc-heads.net/>
- Minecraft Font <https://mojang.com/>
- Crafatar <https://crafatar.com/>
- VanillaTilt <https://micku7zu.github.io/vanilla-tilt.js/>
- jQuery <https://jquery.com/>