Florentin Rollet Théophane Gué Lucien Labet Abla Sefrioui

Groupe 24

Match&Burn:

Rapport de projet web de Start-up Nation

Concept du projet:

Le but de Start-up Nation était de créer une plateforme web sur laquelle conceptfinder et inventeur pourrait proposer des projets, de nouveaux concept à des professionnels et échanger dessus. Ainsi cette plateforme augmenterait la visibilité de concepts de start-ups ingénieuses et permettrait à ces concepts de se réaliser plus aisément.

Ressources:

Pour réaliser ce site web, on a utilisé du CSS dont certains outils Bootstrap pour l'esthétique du site, du PHP (version 7.1 et plus), du Javascript ainsi que du html 5 pour le code source.

Les fonctionnalités de la plateforme:

*: nécessite d'être connecté à un compte utilisateur ou administrateur

**: nécessite d'être connecté à un compte administrateur

-Visionner et rechercher des annonces et leurs commentaires par mots-clés.
-Se connecter à un compte existant
-Créer un compte utilisateur
-Modifier son compte*
-Supprimer son compte*
-Poster une annonce*
-Modifier une annonce que l'on a posté*
-Supprimer une annonce que l'on a posté*
-Commenter une annonce *
-Signaler une annonce*
-Rechercher un utilisateur*
-Supprimer une annonce que l'on n'a pas posté**
-Supprimer un compte qui ne nous appartient pas**
-Voir les annonces signalées**

Base de Données:

Pour l'accès à la base de donnée,on a utilisé l'objet de données PHP: PDO. Ainsi, le fichier DbAdaperfactory.php -qui fait le lien entre la base de donnée et le site - retournera un PDO.
Lors d'une requête SQL, on se connecte à la BDD grâce au PDO, puis on réalise la requête en utilisant : les fonctions prepare et execute ou query, les deux ayant été utilisé dans le projet.
L'avantage de *prepare+execute* par rapport à *query* est le contrôle de la requête, ce qui permet de ne pas insérer n'importe quoi dans la BDD,un contrôle que *query* n'a pas. Toute requête qui inclut le mot-clef d'un utilisateur aura été préparée pour éviter les injections SQL.

En effet avec *prepare+execute,* on peut utiliser la fonction bindParam qui renvoie une chaîne de caractère brute ce qui évite les injections sql.
Une fois la requête reçu, on utilise une fonction PDO (fetch ou fecthAll) qui permet de récupérer tout ou une partie du résultat de la requête

Nous avons voulu construire ce projet sur le modèle d'un réseau social, il allait ainsi de soi que l'on ait les fonctionnalités de base d'un réseau social. C'est pourquoi on a implémenté un SCRUD pour les annonces et un SCRUD pour les utilisateurs.

Si l'utilisateur veut créer, modifier ou supprimer une annonce ou un compte, un formulaire va prendre en compte la requête de l'utilisateur et va la rediriger vers un fichier php qui va la traiter.
Pour le SCRUD utilisateur, on peut :
-Créer un compte via la page inscription.php
-Modifier un compte via espace_membre.php
-Voir ou rechercher un compte via index.php
-Supprimer un compte via espace_membre.php

Pour le SCRUD annonce, on peut :
-Créer un compte via la page espace_membre.php

-Modifier un compte via espace_membre.php

-Voir ou rechercher une annonce via index.php

-Voir en détail via ad.php

-Supprimer un compte via espace_membre.php

Signalement:

La sécurité et la réputation d'une plate-forme étant primordiales, nous avons décidé d'implémenter une fonction de signalement des annonces avec la raison pour laquelle l'annonce est signalée. Dès lors, une annonce est marquée comme signalée aux yeux des administrateurs, et ceux-ci auront aussi accès à la raison de chaque signalement qu'elle aurait reçu.

Moteur de recherche:

Il fallait également que l'utilisateur puisse non seulement rechercher des annonces mais aussi des utilisateurs c'est pourquoi nous avons implémenté un système de moteur de recherche des annonces par mots-clés,lequel n'affiche que les 30 premiers résultats dans l'ordre ante-chronologique.

Pour cela, on a utilisé dans notre requête SQL la recherche de motif («LIKE%mot%»). Par défaut, le site recherche automatiquement toutes les annonces et donc affiche les 30 dernières annonces à avoir été postées. Un clic sur l'annonce de votre choix vous envoie sur add.php pour la voir en détails.

Pour rechercher des utilisateurs,2 options s'offrent à vous:

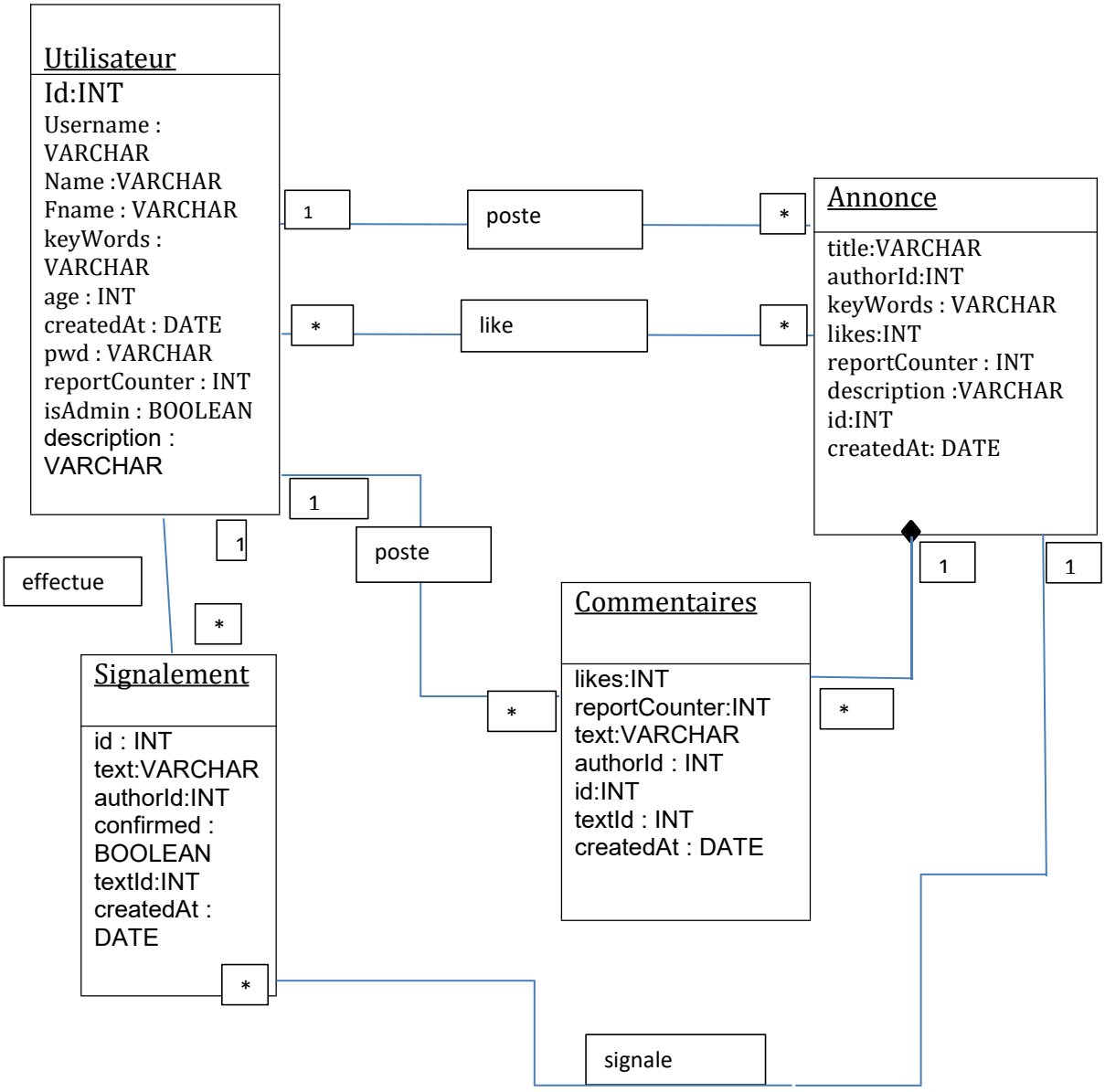
-Si vous êtes déjà connecté au site, vous pouvez rechercher les utilisateurs par nom, prénom,nom d'utilisateur, e-mails ou mots-clés .

-Sinon vous pouvez aller sur le profil d'un utilisateur à partir d'une de ses annonces, en cliquant sur son nom.

Florentin Rollet Théophane Gué Lucien Labet Abla Sefrioui

Groupe 24

Diagramme UML du site:



Javascript:

Comme demandé, nous avons utilisé du javascript dans notre code. Son utilisation regroupe 3 cas. Le javascript permet d'une part d'afficher en direct une vérification des données des formulaires (insciption.php par exemple). Il permet aussi de vérifier si on peut envoyer ce formulaire. Javascript permet aussi d'afficher des fenêtres pop-ups de confirmation lorsqu'on souhaite faire une action importante (supprimer son compte ou celui d'un utilisateur par exemple) Enfin, nous avons utilisé javascript pour

afficher des formulaires ou les cacher selon les envies de l'utilisateur. Ce cas apparaît lorsque l'utilisateur veut modifier son mot de passe par exemple.

CSS:

Pour designer le site, nous avons utilisé les fichiers de Bootstrap en complément des nôtres qui se trouvent dans le dossier css. Les fichiers Css permettent de pouvoir designer élégamment notre site en ajustant les designs bootstrap à nos pages.