

Projet WEB 2018-2019

Lucas HERFORT

Antony Fred BACILIO SALAZAR

Imane EL MOUL

14/05/19

Table des matières

1. Présentation de l'application.....	2
2. Cahier des charges et répartition des tâches.....	2
3. Développement de l'application.....	3
3.1 Mise en place de la base de donnée3	
3.2 La page d'accueil	
3.2.1 Inscription	
3.2.2 Connexion	
3.2.3 Contact administrateur	
3.3 La page d'administration	
3.4 La page de profil	
3.4.1 Edition du profil	
3.4.2 Les publications	
3.4.3 Barre de recherche	
3.4.4 Abonnements et abonnés	
4. Problèmes non résolus et conclusion.....	9

1. Présentation de l'application

Vous souhaitez partager votre passion ? **ArtIstE** est fait pour vous. **ArtIstE** est un nouveau réseau social qui reste très similaire à Instagram mais qui a la particularité de s'adresser exclusivement aux artistes quelque soit leur domaine : architecture, sculpture, peinture, musique, cinéma etc...

ArtIstE vous donne la possibilité de créer votre propre compte afin de partager votre contenu multimédia, qu'il s'agisse de photos ou de vidéos. Ce contenu sera ensuite visible par vos amis.

Afin d'éviter toute intrusion, votre compte devra être confirmé par l'administrateur du site. Une fois confirmé, vous aurez accès à votre propre compte personnel.

A tout moment, si l'administrateur se rend compte que votre activité sur l'application est suspecte ou que votre profil a été piraté, il peut à tout moment bloquer votre compte et lui seul sera en mesure que le débloquent ou de le supprimer définitivement.

2. Cahier des charges et répartition des tâches

Contraintes générales :

- Une authentification
- Un compte administrateur donnant les droits à certaines fonctionnalités (au choix)
- Un profil utilisateur éditable
- Une base de données relationnelle :
 - au moins 3 tables
 - au moins une table de jointure (n...n)
 - au moins une jointure dans une requête
 - des INSERT, DELETE, UPDATE, SELECT
- Un CRUD (Create Read Update Delete)

contraintes imposées par nous même :

- Un système de publication de contenu multimédia
- Un système d'abonnement
- Un système de Like/Dislike

Répartition des tâches :

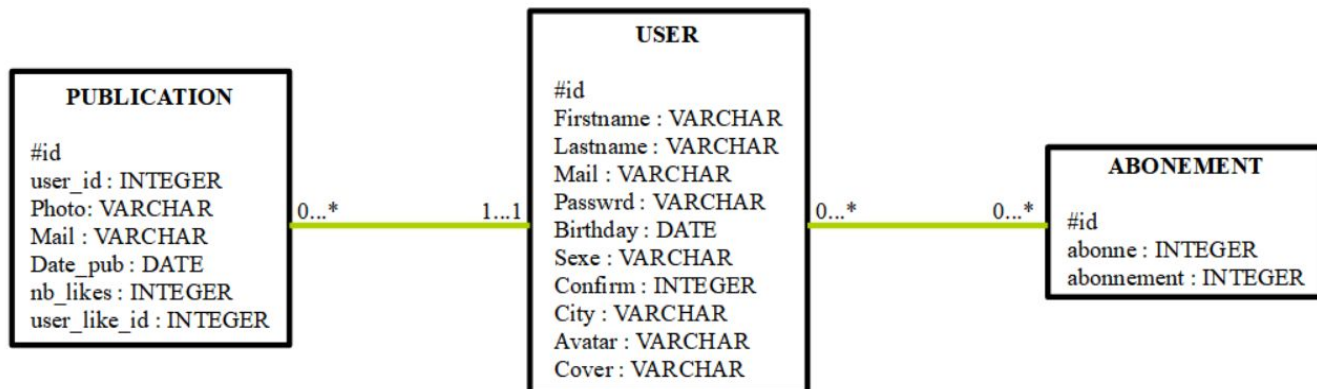
Lucas Herfort	Antony Bacilio	Imane EL MOUL
<ul style="list-style-type: none">• Page d'accueil avec formulaire et connexion• Page administrateur avec ses fonctionnalités	<ul style="list-style-type: none">• Visuel de toutes les pages du site (CSS)• Barre de recherche	<ul style="list-style-type: none">• Page profil• Edition du profil• Système de publication

3 Développement de l'application

Tout au long du développement de l'application, nous avons utilisé le modèle MVC avec d'un côté la partie "view" et de l'autre la partie "controller". Pour démarrer l'application, nous avons écrit index.php qui va exécuter ArtistModel.php et ArtistView.php.

3.1 Mise en place de la base de données

Pour cette application, nous allons utiliser 3 tables dans nos BDD : user, publication et abonnement. Voici le schéma relationnel entre ces 3 tables :

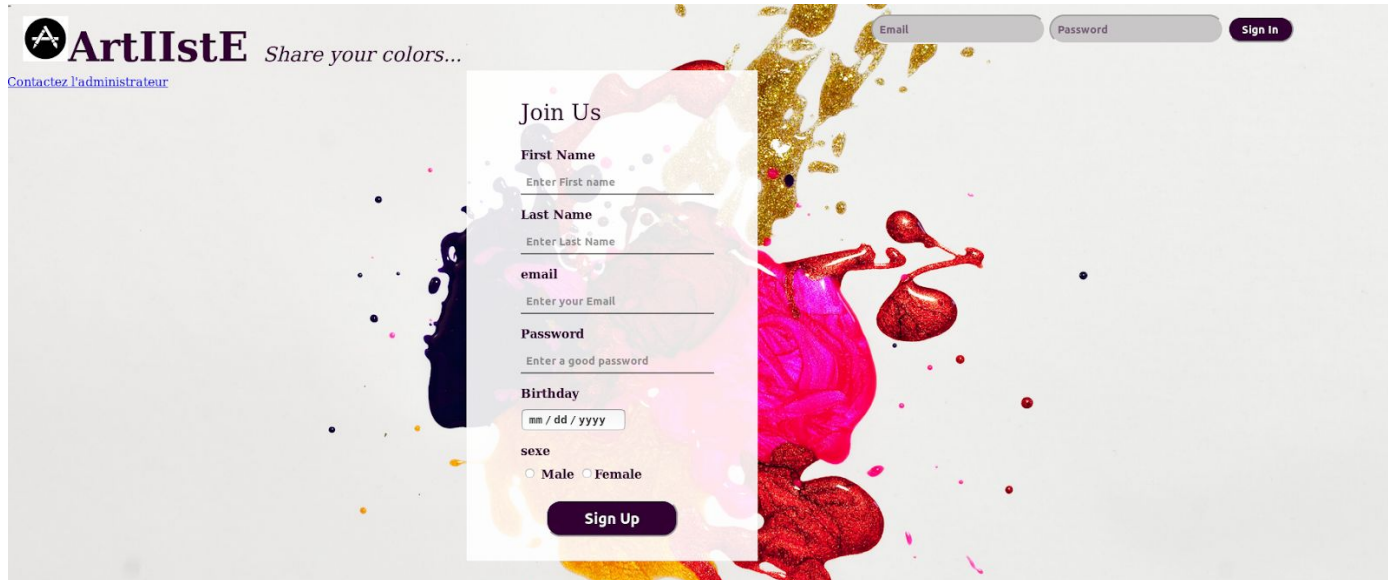


```
CREATE TABLE "user" (id SERIAL PRIMARY KEY ,firstname VARCHAR NOT NULL ,
lastname VARCHAR NOT NULL , mail VARCHAR NOT NULL, passwd VARCHAR
NOT NULL, birthday date, sexe Varchar, confirm INTEGER, city Varchar, description
Varchar, avatar Varchar, cover Varchar);
```

```
CREATE TABLE "Publication" (id SERIAL PRIMARY KEY,user_id INTEGER,
description VARCHAR, photo Varchar, date_pub date, nb_likes INTEGER DEFAULT 0,
user_like_id INTEGER);
```

CREATE TABLE "Abonnement" (id SERIAL PRIMARY KEY, abonne INTEGER, abonnement INTEGER);

3.2 La page d'accueil

The image shows a web browser window with the ArtIstE website. The header features the ArtIstE logo with the tagline 'Share your colors...' and a link to 'Contactez l'administrateur'. On the right, there are input fields for 'Email' and 'Password', and a 'Sign In' button. The main content area displays a 'Join Us' registration form. The form includes fields for 'First Name' (with placeholder 'Enter First name'), 'Last Name' (with placeholder 'Enter Last Name'), 'email' (with placeholder 'Enter your Email'), 'Password' (with placeholder 'Enter a good password'), and 'Birthday' (with a date picker showing 'mm / dd / yyyy'). Below these is a 'sexe' section with radio buttons for 'Male' and 'Female'. A 'Sign Up' button is at the bottom of the form. The background of the website is a colorful abstract painting with splashes of red, yellow, and blue.

3.2.1 Inscription

Pour que l'utilisateur puisse créer son profil, nous avons créé un script artistView.php un formulaire simple dans lequel l'utilisateur devra saisir ; *nom, prénom, adresse mail, mot de passe, date de naissance* et *sexe*.

Une fois ces informations enregistrées(en cliquant sur le bouton sign up), c'est le ArtistModel.php qui va gérer les messages d'erreurs si les informations sont incorrects. Afin de minimiser les risques liés aux attaques de type injections html, nous avons utilisé la fonction php htmlspecialchars().

Avec ArtistModel.php, nous avons effectués tous les tests nécessaire :

- vérification que tous les champs ne sont pas vides
- $\text{nom} \leq 30$ caractères
- $\text{prénom} \leq 30$ caractères
- vérification dans la BDD que l'adresse mail n'est pas déjà utilisée.

Si tout est correct, alors on ajoute ces informations à la BDD avec un "INSERT INTO", sinon on affiche des messages d'erreurs avec la variable \$error.

3.2.2 Connexion

Une fois que l'utilisateur a créé son compte, il devra attendre la validation du compte par l'administrateur (voir 3.3.2). Pour qu'il puisse se connecter à sa page de profil, nous avons créé un formulaire dans ArtistModel.php ou il sera demandé à l'utilisateur de rentrer son *adresse mail* ainsi que son *mot de passe*. Ces informations seront également traitées dans ArtistModel.php avec tous les tests nécessaires :

- vérification que tous les champs ne sont pas vides
- vérification dans la BDD que l'adresse mail et le mot de passe sont corrects
- vérification du statut du compte (voir 3.3.2)

Si tout est correct, alors on fait appel à la fonction header() afin de diriger l'utilisateur vers page de profil (voir 3.4).

3.2.3 Contact administrateur

Dans la page d'accueil, nous avons ajouté un lien de type `Contactez l'administrateur` permettant ainsi en cliquant dessus à un utilisateur d'être redirigé sur sa boîte mail avec l'adresse mail de l'administrateur en destinataire, ce qui lui sera très utile si son compte n'a pas été confirmé ou a été bloqué.

3.3 La page d'administration

#	Firstname	Lastname	Email	Password	Age	sexe	Status	Confirmer	Bloquer	Supprimer
3	Lucas	Herfort	lucas.herfort@gmail.com	123	1977-05-12	Male	1		Bloquer	Supprimer
7	Manuel	Holloway	manuel.holloway@gmail.com	123	1979-05-25	Male	1		Bloquer	Supprimer
9	Otis	Roberson	otis.roberson@gmail.com	123	1995-11-01	Male	1		Bloquer	Supprimer
12	Silvia	Mcguire	silviamcguire@gmail.com	123	1978-04-02	Fem	1		Bloquer	Supprimer
11	Vicky	Pearson	vicky.pearson@gmail.com	123	1978-10-10	Fem	1		Bloquer	Supprimer
5	Yvette	Angel	yvette.angel@gmail.com	123	1972-01-24	Fem	1		Bloquer	Supprimer

Pour la partie administration, nous avons ajouté dans la table user de db.sql un utilisateur avec comme adresse mail artiste.admin@gmail.com et mot de passe : admin . Ce dernier correspondra à notre administrateur.

Lorsque l'on se connecte avec ces identifiants, alors on arrive directement sur la page d'administration ProfilAdminView.php.

Une fois sur la page d'administration, on peut accéder à la liste des utilisateurs en cliquant sur "Utilisateurs". A partir de là, l'administrateur a accès à toutes les informations y compris aux fonctionnalités suivantes : confirmer, bloquer ou supprimer un compte.

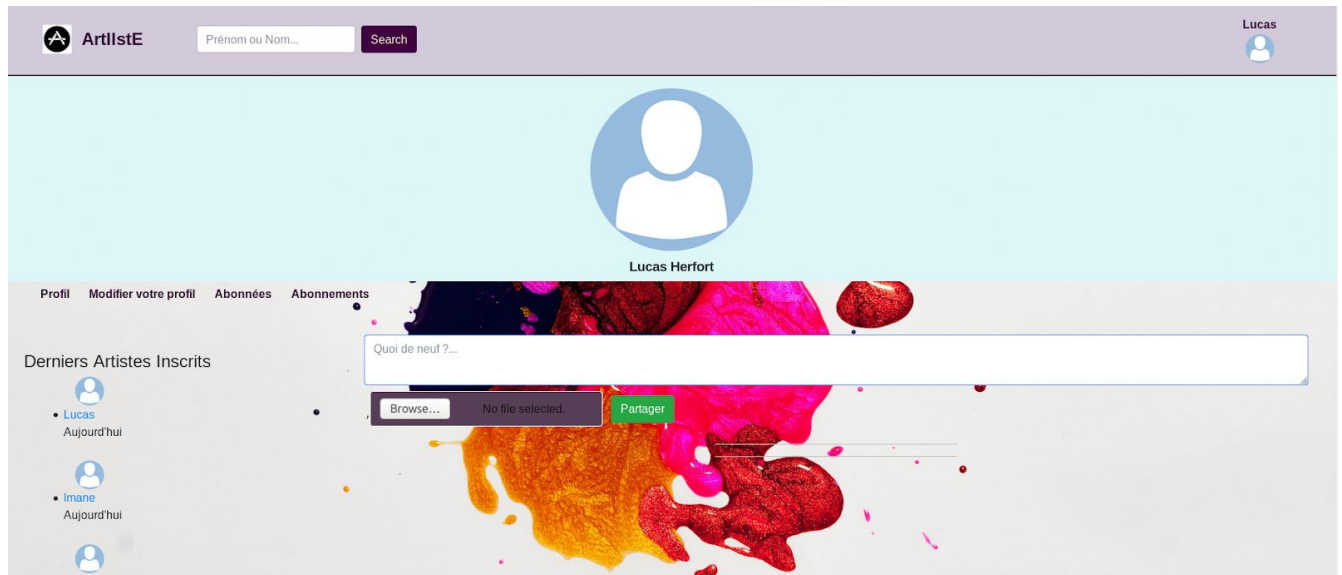
Pour réaliser ce système, nous avons créé un nouvel attribut "confirm" dans la table "user". Cet attribut peut prendre 4 valeurs possibles :

- 0 : compte pas encore validé
- 1 : compte validé
- 2 : compte administrateur

→ 3 : compte bloqué

Cet attribut est ensuite utilisé dans ArtistModel.php afin d'afficher un message d'erreur à l'utilisateur pour l'informer de l'état de son compte si jamais il n'arrive pas à accéder à sa page de profil.

3.4 La page de profil



Une fois connecté, on se retrouve sur cette page de profil (profilView.php). De cette page, il est possible de :

- Modifier son profil
- Consulter sa liste d'abonnés et d'abonnements
- Rechercher des personnes
- Publier du contenu

3.4.1 Edition du profil

Pour que l'utilisateur puisse éditer son profil, nous avons créé un le fichier editProfilView.php avec un formulaire assez similaire à celui de l'inscription mais qui permet : d'apporter des informations en plus comme sa ville d'origine et une petite description de soi ou bien de personnaliser son profil en ajoutant une image de profil ainsi qu'une image de couverture. Le format d'une image doit être de type : 'png','jpg','gif','jpeg' et ne doivent pas dépasser la taille de 2097152 octets.

Les modifications apportées sont également traitées par le fichier ArtistModel.php puis les nouvelles informations (si elles sont valides) sont ajoutées dans la BDD avec des requêtes de la forme :
UPDATE ... SET WHERE ...

3.4.2 Les publications



Quoi de neuf ?...

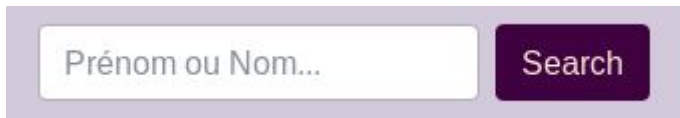
Browse... No file selected. Partager

Pour faire ce système un publications, nous avons utilisé la table PUBLICATION de notre BDD et créé un formulaire afin d'y renseigner une image à publier ainsi qu'un petit texte de description. Une fois les champs saisis, les informations sont transmises au `publicationModel.php` afin d'être traitées puis ajouté dans la BDD si ces informations sont valides.

Une fois l'ajout dans la BDD, la publication sera visible par l'utilisateur ainsi que par tous ses followers. Il sera inquiéter le nom de la personne ainsi que l'heure de la publication :



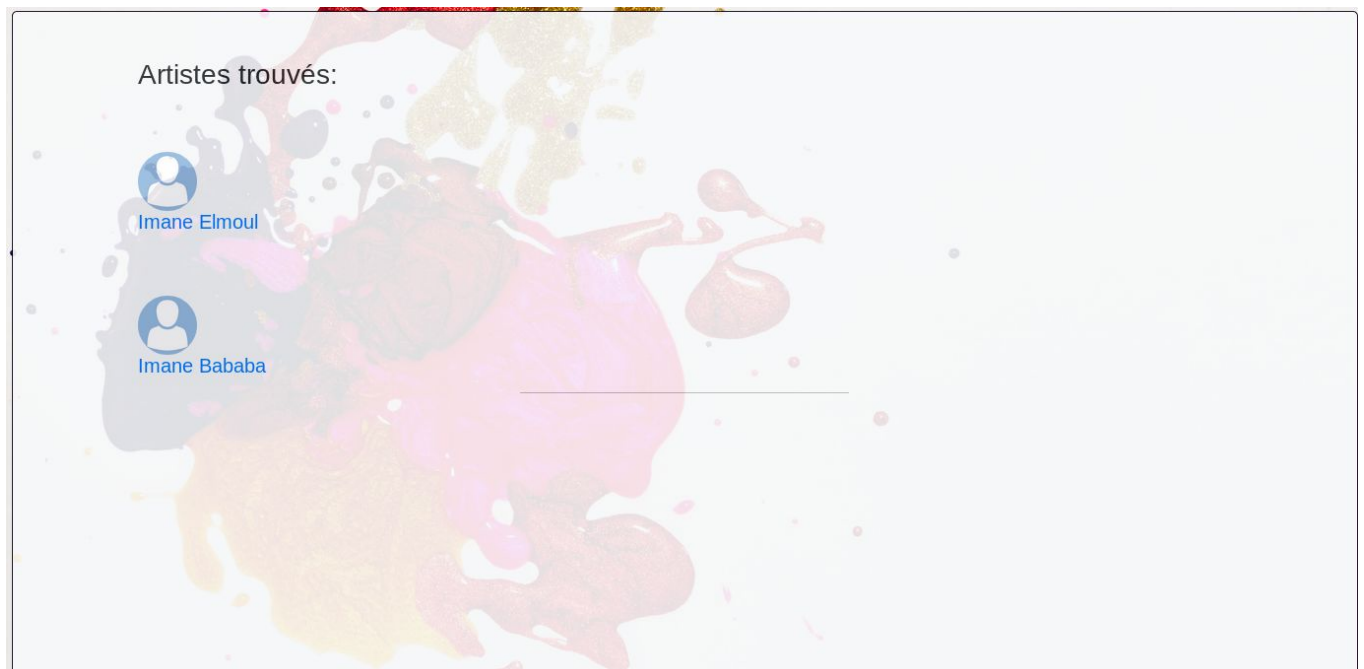
3.4.3 Barre de recherche :

A search bar with a light purple background. It contains a white text input field with the placeholder text "Prénom ou Nom..." and a dark purple button with the text "Search" in white.

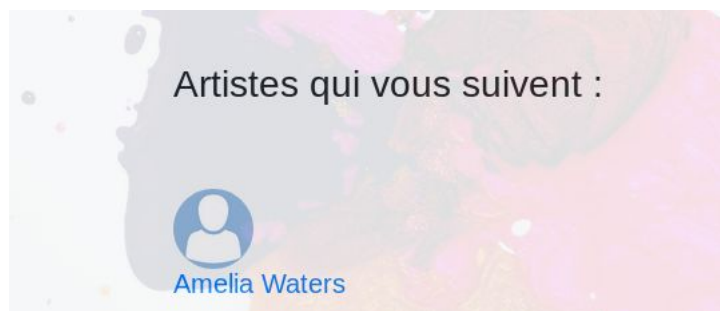
Pour réaliser ce système, nous avons défini un champ texte dans la page de profil puis utilisé le script `recherche.php` qui va exécuter une requête SQL afin de filtrer les résultats selon les contraintes sur le nom et le prénom.

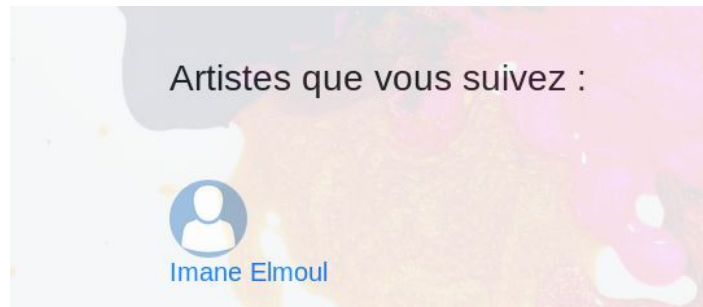
S'il y a au moins un tuple de retour de la requête SQL, alors on affiche les informations de ce tuple comme ça l'utilisateur pour chercher ses amis et ainsi accéder à leur page de profil.

Résultat obtenu si l'on cherche une personne qui s'appelle Imane :



3.4.4 Abonnements et abonnés :





Maintenant qu'un utilisateur peut trouver d'autres personnes avec la barre de recherche, il va désormais avoir la possibilité de suivre des personnes ou d'être suivi par d'autres personnes.

Pour cela, nous avons utilisé la table Abonnement de notre BDD. Le principe est simple, quand une personne A s'abonne à la personne B, alors A est abonné à B tandis que B possède comme abonnement A d'où les deux attributs *abonne* et *abonnement*.

Nous avons commencé par mettre en place pour chaque profil utilisateur deux boutons, l'un permettant de s'abonner et l'autre de se désabonner. Les boutons Abonnés et abonnements servent aussi, en faisant appel à FollowersListView.php et FollowedListView.php d'afficher la liste de ses abonnées et de ses abonnements.

4. Problèmes non résolus et conclusion

Nous n'avons malheureusement pas eu le temps d'implanter un système de like/dislike pour les publications. Nous avons également rencontré beaucoup de problèmes liés à l'utilisation des BDD car certaines requêtes ne fonctionnaient pas alors qu'il n'y avait aucun message d'erreur. La prise en main de Docker a été très difficile au début et a légèrement contribué à notre retard sur certaines choses.

Néanmoins, ce premier projet en développement web a été une belle réussite avec des fonctionnalités plus ou moins complexes qui fonctionnent. C'est ainsi une très bonne manière de nous préparer pour nos futurs stages ainsi qu'à notre futur métier d'ingénieur.