



Art et Cr éation en Partage

ART ET CREATION EN PARTAGE

Dossier de Projet

Projet réalisé dans le cadre d'un stage pour la présentation au Titre Professionnel de

Développeur Web et Web Mobile

Présenté par

Céline CHAIGNE

Organisme de formation :

ENI, Ecole informatique

Promotion avril 2023

Stage réalisé auprès de l'Association Art et Cr éation en Partage du
25/09/2023 au 10/11/2023



Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à la réussite de mon stage en développement Web et Web mobile, une étape cruciale de ma formation et ma reconversion.

Je remercie Françoise Gerlache, qui a eu la gentillesse d'accepter d'être ma maître de stage au pied levé et a toujours porté un regard bienveillant sur mon travail.

Je souhaite remercier vivement Cédric, qui, d'une part, s'est démené pour m'ouvrir les portes de ce stage en urgence mais surtout et avant tout qui m'a aidée, conseillée, débuggé tout au long de la formation et du stage. Son expérience et ses précieux conseils m'ont énormément fait progresser.

Un grand merci également à Xavier et Antoine pour leurs sourires, leur écoute et leurs conseils.

Je n'oublie pas de remercier mes enfants et Héli, pour leur compréhension et leur patience pendant cette période exigeante. Leur soutien inconditionnel a été un moteur précieux pour persévérer dans cette aventure.

Table des matières

Introduction.....	4
1 Présentation de l'entreprise.....	5
2 Contexte et organisation.....	6
2.1 Contexte	6
2.2 Organisation	6
3 Liste des compétences du référentiel qui sont couvertes par le projet	7
4 Résumé du projet	8
5 Cahier des charges.....	9
5.1 Expression des besoins.....	9
5.2 Description fonctionnelle	12
5.3 Maquettes	13
6 Spécifications techniques du projet	15
6.1 Matériel utilisé	15
7 Compétences du référentiel et réalisations.....	16
7.1 CP 1 Maquetter une application	16
7.2 CP 2 Réaliser une interface web statique et adaptable	18
7.3 CP 3 Développer une interface utilisateur web dynamique	19
7.4 CP 5 Créer une base de données.....	20
7.5 CP 6 Développer les composants d'accès aux données	23
7.6 CP 7 Développer la partie back-end d'une application web ou web mobile	24
8 Veille technique et sécurité.....	27
9 Autres réalisations.....	29
9.1 Filtre des artistes	29
9.2 Téléchargement d'images	32
9.3 Lightbox	32
9.4 Fixtures	32
10 Description d'une situation de travail ayant nécessité une recherche	33
11 Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment	35
12 Jeu d'essai.....	36
Bilan.....	42
Conclusion	43
Webographie.....	44
Annexes	45

Introduction

Le projet présenté dans ce rapport a été réalisé lors d'un stage de fin de formation en Développeur Web et Web Mobile, dans le cadre de ma reconversion professionnelle. Après plus de 18 ans au sein de la Police Scientifique (Police Nationale – Ministère de l'Intérieur), j'ai entrepris une reconversion totale, n'ayant aucune expérience en programmation ni en informatique.

La formation de Développeur Web et Web Mobile dispensée par l'ENI, Ecole Informatique, s'est déroulée en ligne sur une durée d'environ 5 mois, suivie d'une période de stage en entreprise de 2 mois.

La réalisation de ce projet lors du stage a pris 7 semaines au lieu de 8. En effet, j'ai débuté ma période de stage dans une autre entreprise sur un autre projet, mais nous avons mis fin à ce stage d'un commun accord avec mon précédent maître de stage.

Le projet actuel a été réalisé pour le compte de l'Association Art et Création en Partage. Son objectif est de présenter l'association, de promouvoir les événements artistiques gérés par celle-ci, ainsi que les artistes participants.

Pendant ce stage, je n'ai pas bénéficié d'un tuteur technique. Mme. Gerlache, ma maître de stage, n'a aucune compétence en informatique, et son rôle relevait davantage de celui de client. J'ai réalisé l'intégralité du projet de manière autonome, en travaillant à distance.

Vous pourrez trouver en [Annexe 9](#) des vues de différentes pages de l'application depuis le navigateur de site web Firefox en fin de stage.

1 Présentation de l'entreprise

L'association Art et Création en Partage est à l'origine le regroupement de deux artistes soucieux de sortir des sentiers battus des expositions classiques et conventionnelles. Ils ont choisi de s'organiser en association, conformément à la loi de 1901. Françoise Gerlache, ma maître de stage, est la présidente de l'association Art et Création en Partage. Le but de cette association est de créer des événements favorisant la promotion d'artistes de toutes disciplines, qu'ils soient confirmés ou débutants, en offrant une visibilité à leur travail.

L'association privilégie des lieux atypiques plutôt que des salles d'exposition traditionnelles. Cela permet aux artistes de mettre en valeur leurs œuvres de manière différente et au public de découvrir des espaces originaux, tout en prenant le temps d'apprécier les œuvres et d'en apprendre davantage sur les artistes.

Le principal événement organisé à ce jour s'intitule « Barjac – Hors des sentiers battus » et a lieu le temps d'un week-end chaque année depuis deux ans à Barjac, une petite ville du Gard. Lors de la première édition, les œuvres et les artistes étaient dispersés dans les rues de la ville, offrant ainsi aux visiteurs une expérience artistique inédite. La dernière édition s'est déroulée en septembre 2023 et s'est tenue dans la propriété privée de Françoise Gerlache.

Pour chaque événement, Art et Création en Partage sélectionne une vingtaine d'artistes en fonction de ses coups de cœur, de la qualité de leurs créations, de l'histoire de leurs œuvres, de leur maîtrise technique, de leur vision unique et du message qu'ils apportent. Les membres de l'association mettent l'accent sur la découverte de nouveaux talents. À cette fin, un appel à candidatures est lancé pour les artistes souhaitant participer. Chaque candidature est soigneusement étudiée avant l'intégration de l'artiste à l'exposition.

Actuellement, seul un compte Facebook dédié aux événements « Barjac – Hors des sentiers Battus » existe et l'association Art et Création en Partage ne dispose pas d'un site web dédié. Cependant, l'association a pour projet la création d'un site internet pour donner davantage de visibilité aux événements et aux artistes dans le but d'attirer plus de visiteurs et d'artistes candidats et afin d'enrichir la diversité et la qualité des sélections artistiques.

2 Contexte et organisation

2.1 Contexte

Ce projet s'est déroulé sur une période de 7 semaines au lieu des 8 semaines initialement prévues. En effet, j'ai entamé ma période de stage dans une autre entreprise sur un projet différent mais nous avons convenu de mettre fin à ce stage d'un commun accord avec mon précédent maître de stage.

Ce changement de stage inattendu s'est déroulé sans problème, bien que de manière précipitée, afin de ne pas perdre trop de temps. Un de mes collègues de formation connaissait Mme Gerlache de l'association Art et Création en Partage et il m'a mis en relation avec elle pour vérifier si le projet de création d'un site internet pour l'association était toujours d'actualité.

Mme Gerlache a manifesté un vif intérêt pour la réalisation de ce projet et a gentiment accepté d'être ma maître de stage. N'ayant aucune connaissance en développement web ou en informatique, elle a pris le rôle de client pour ce projet.

J'ai réalisé ce projet en autonomie, faute de tuteur technique. J'ai sollicité des conseils auprès de collègues de formation plus expérimentés et j'ai effectué de nombreuses recherches personnelles en consultant divers forums, tutoriels et documentations.

Le stage s'est entièrement déroulé en distanciel, étant donné que Mme Gerlache et moi étions géographiquement éloignés. Notre communication s'est principalement faite par téléphone et par courrier électronique, avec quelques réunions en visioconférence pour présenter l'avancement du projet et discuter des ajustements nécessaires.

2.2 Organisation

Pour m'organiser dans l'avancement du projet, j'ai pris des notes dans un agenda quotidien répertoriant les principales actions de la journée ainsi que les principaux liens vers les ressources et tutoriels que j'ai trouvés.

La réalisation s'est déroulée plus ou moins selon une méthode agile. En effet, Mme Gerlache ne disposait pas d'un cahier des charges précis et sa vision du projet n'était pas totalement. Par conséquent, le projet s'est construit et a évolué au fil des discussions et de l'avancement.

3 Liste des compétences du référentiel qui sont couvertes par le projet

Nom et prénom du candidat : CHAIGNE Céline.....

Document complété d'un commun accord entre le stagiaire et le responsable du stage en entreprise à joindre au rapport d'activité.

Compétences Voir le détail dans le référentiel d'emploi, d'activités et de compétences	Cocher les compétences mises en œuvre lors du projet en entreprise (en totalité ou partiellement)
C1 - Maquetter une application	x
C2 - Réaliser une interface utilisateur web statique et adaptable	x
C3 - Développer une interface utilisateur web dynamique	x
C4 - Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce	
C5 - Créer une base de données	x
C6 - Développer les composants d'accès aux données	x
C7 - Développer la partie back-end d'une application web ou web mobile	x
C8 - Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce	

Observations éventuelles :

Du stagiaire : /

De l'entreprise : /

Nom et Prénom du stagiaire :
Chaigné Céline
Signature

Entreprise :
Association Art et Partage en Création
Nom du responsable de stage :
Françoise Gerlache
Signature :

4 Résumé du projet

Le projet présenté vise à développer depuis zéro une application web pour l'association Art et Création en Partage. L'objectif est de présenter l'association, visualiser les événements et les artistes participants, tout en offrant à ceux-ci la possibilité de créer et modifier leurs fiches de présentation ainsi que de gérer leur galerie photo. L'application inclut également un back-office pour les administrateurs simplifiant la gestion des utilisateurs, des événements, des artistes et des catégories.

L'environnement technique repose sur le framework Symfony et utilise des outils modernes tels que Figma et TailWindCSS. Dans le cadre du stage, les fonctionnalités attendues ont été mises en place ainsi que certaines fonctionnalités supplémentaires telles que le filtrage des artistes et l'affichage d'images en plein écran pour une meilleure expérience utilisateur.

L'application intègre une série de mesures de sécurité dont la gestion des accès selon les utilisateurs et la validation des données, garantissant ainsi un environnement robuste et fiable.

Bien que des éléments restent à perfectionner, le projet a atteint ses objectifs fondamentaux.

Le stage a été marqué par la nécessité de travailler sans un tuteur technique, ce qui a exigé des recherches approfondies, une lecture de documentations et une veille technologique constante. Cette autonomie a renforcé les compétences d'apprentissage, de résilience et d'initiative. Globalement, le projet a été gratifiant, ouvrant la voie à une reconversion réussie dans le développement web et stimulant le désir d'apprendre et de créer au quotidien.

5 Cahier des charges

5.1 Expression des besoins

5.1.1 User Stories (V1)

J'ai recueilli les besoins de la cliente, Mme Gerlache, au début du projet ainsi qu'en cours de celui-ci. En effet, elle n'avait pas une idée précise des fonctionnalités ou du rendu visuel attendus, et bien sûr, il n'y avait pas de cahier des charges établi.

Elle souhaitait avant tout permettre une bonne visibilité des événements et des artistes qui y participent, tout en offrant une méthode simple et rapide de gestion des nouveaux événements et artistes.

Les éléments de départ recueillis étaient les suivants :

- Permettre aux visiteurs de :
 - Visualiser les renseignements et la description de l'association
 - Visualiser les événements, leurs photos et leur description
 - Visualiser les fiches des artistes avec leurs photos et leur description
- Permettre aux administrateurs de :
 - Créer les événements avec leurs photos et leur description
 - Créer les fiches des artistes avec leurs photos et leur description
 - Permettre aux artistes souhaitant participer à un futur événement de remplir un formulaire, lequel serait ensuite transmis à Mme. Gerlache pour approbation. Une fois l'artiste approuvé, il pourrait être intégré parmi les artistes de l'événement.

5.1.2 Evolution

Au fur et à mesure de l'avancement du projet et des discussions, les fonctionnalités demandées ont évolué.

La partie relative à la gestion des candidatures et des inscriptions d'artistes a été sujette à plusieurs discussions et modifications.

L'une des premières solutions envisagées consistait à permettre aux utilisateurs de s'inscrire et de créer leur propre fiche artiste. Ces fiches, lorsqu'elles étaient "en attente", n'étaient pas visibles par les visiteurs. L'administrateur avait la possibilité de récupérer les fiches en attente de validation dans une section réservée. Si la fiche était validée, elle devenait visible pour les visiteurs, sinon elle était supprimée. J'avais également évoqué la possibilité d'envoyer automatiquement un e-mail à l'administrateur en cas de création d'une nouvelle fiche, ainsi qu'à l'utilisateur postulant en cas de réponse positive ou négative.

Cependant, cette solution n'a pas été retenue, car elle était non seulement compliquée à gérer pour l'administrateur (la validation des artistes se faisait après une réunion annuelle de la commission de

l'association) mais elle pouvait également s'avérer très complexe et frustrante pour les candidats qui étaient souvent des personnes âgées ou peu familiarisées avec l'utilisation d'Internet. De plus, pour les candidats refusés, l'appréhension à l'idée de laisser leurs coordonnées et leurs photos de création sans savoir si celles-ci seraient utilisées sans leur accord a également été soulevée.

J'ai proposé d'autres solutions intermédiaires, telles que l'inscription avec un code d'accès.

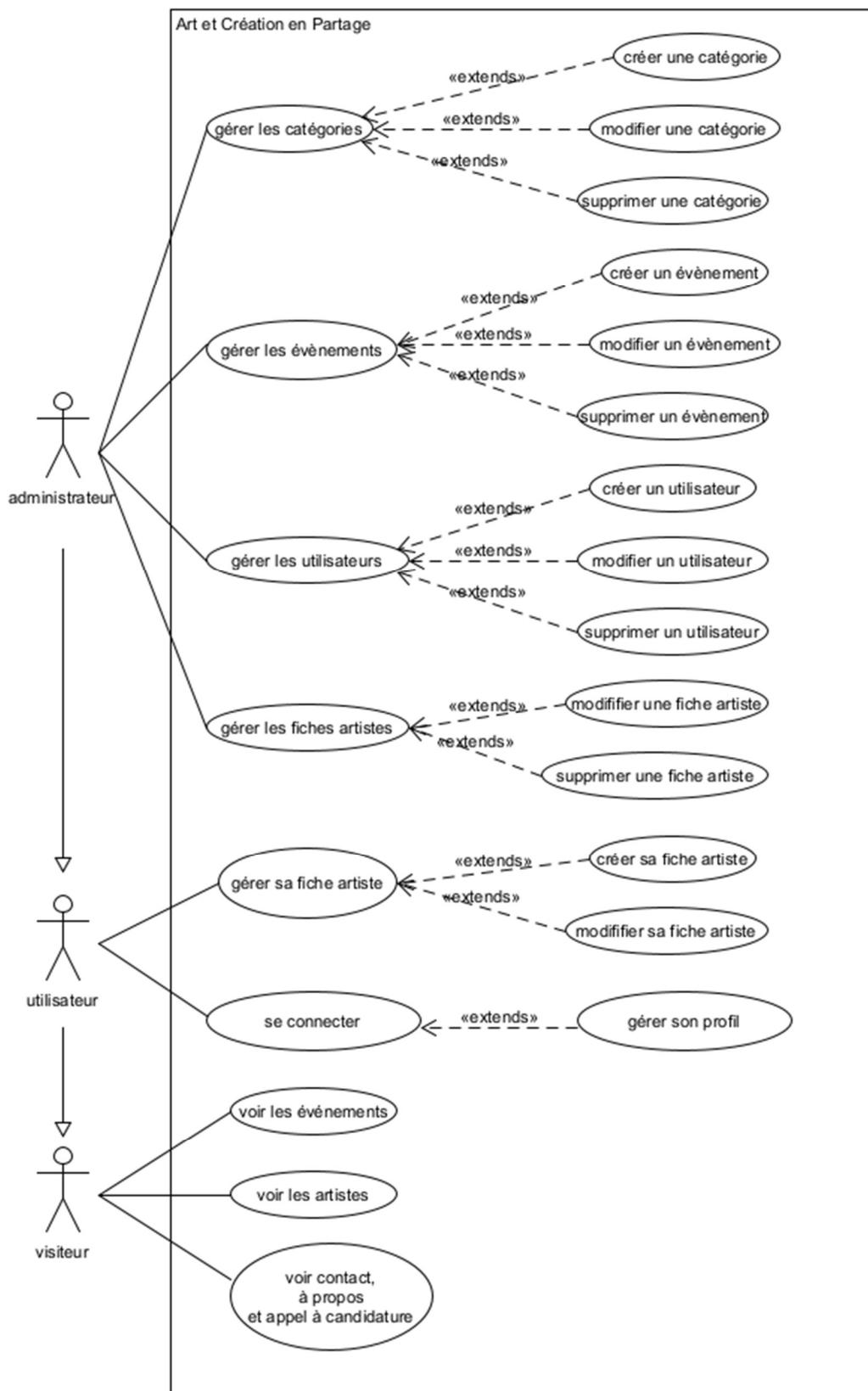
Finalement, une solution beaucoup plus simple et conforme aux pratiques actuelles de l'association a été retenue. Elle consiste en l'envoi direct d'un e-mail de la part des candidats à l'association, comprenant une présentation de l'artiste et quelques photos. Après validation par la commission, les artistes retenus sont informés par e-mail, avec indication de leur mot de passe de connexion au site. L'administrateur est responsable de l'inscription des utilisateurs qui, eux, ont la responsabilité de créer leur propre fiche artiste. Si un artiste postulant n'est pas en mesure de créer sa fiche, l'administrateur peut facilement le faire à sa place.

5.1.3 User Stories (V2)

Les fonctionnalités finales attendues sont :

- Permettre aux visiteurs de :
 - o Visualiser les renseignements et la description de l'association
 - o Visualiser les événements, leurs photos et leur description
 - o Visualiser les fiches des artistes avec leurs photos et leur description
 - o Accéder aux moyens de contact de l'association et aux instructions pour postuler en tant qu'artiste
- Permettre aux utilisateurs connectés de :
 - o Gérer son profil
 - o Créer ou modifier leur fiche artiste avec des photos, une description et des liens vers leurs réseaux sociaux
- Permettre aux administrateurs de :
 - o Créer, modifier et supprimer des utilisateurs (artistes postulants acceptés)
 - o Créer, modifier et supprimer des événements
 - o Modifier et supprimer des fiches d'artistes
 - o Créer, modifier et supprimer des catégories
 - o Gérer le lien entre l'événement et les artistes participants (ajout/suppression)

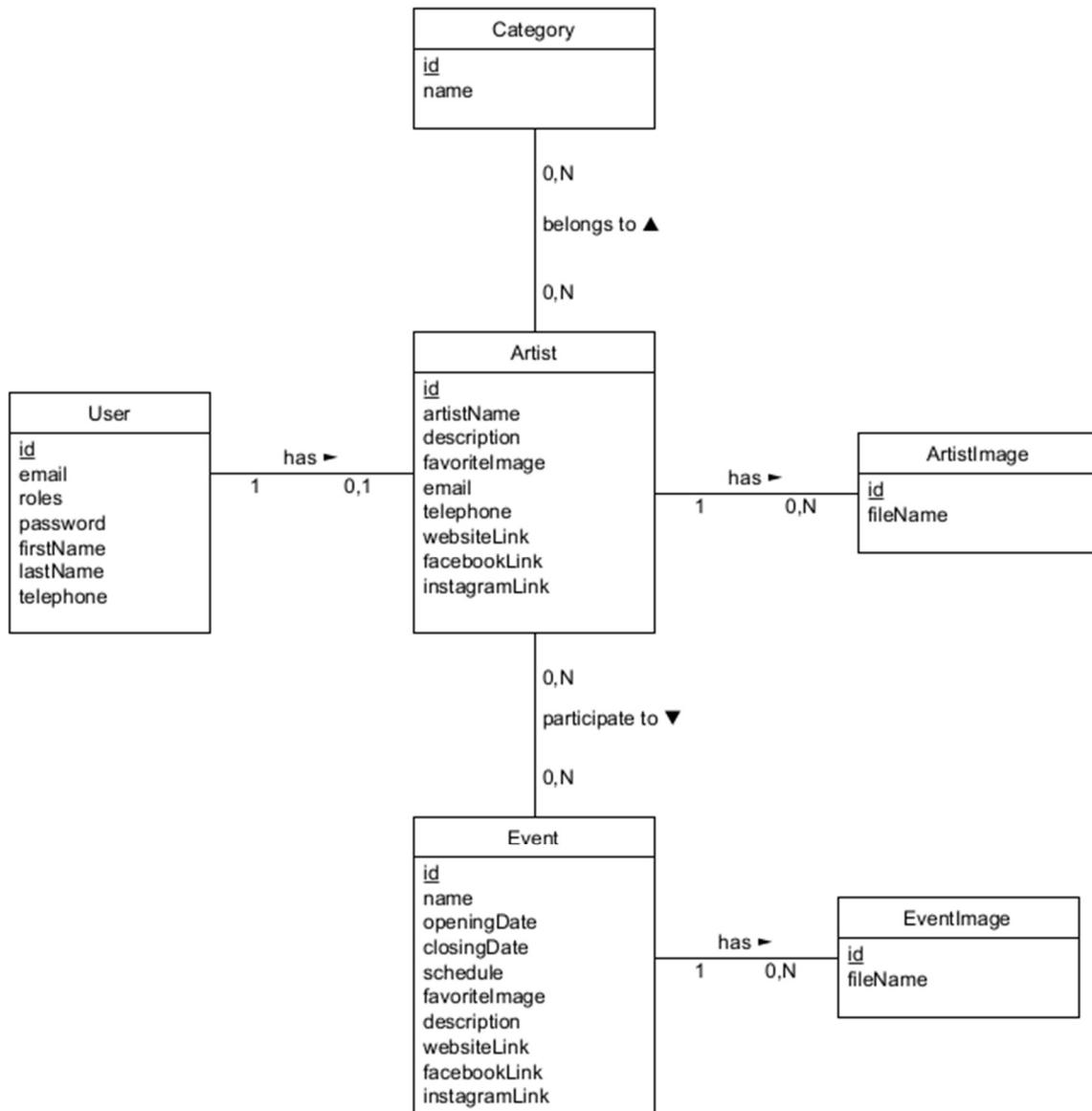
Ci-dessous, vous trouverez le diagramme des cas d'utilisation (use case) que j'ai réalisé avec UMLet :



5.2 Description fonctionnelle

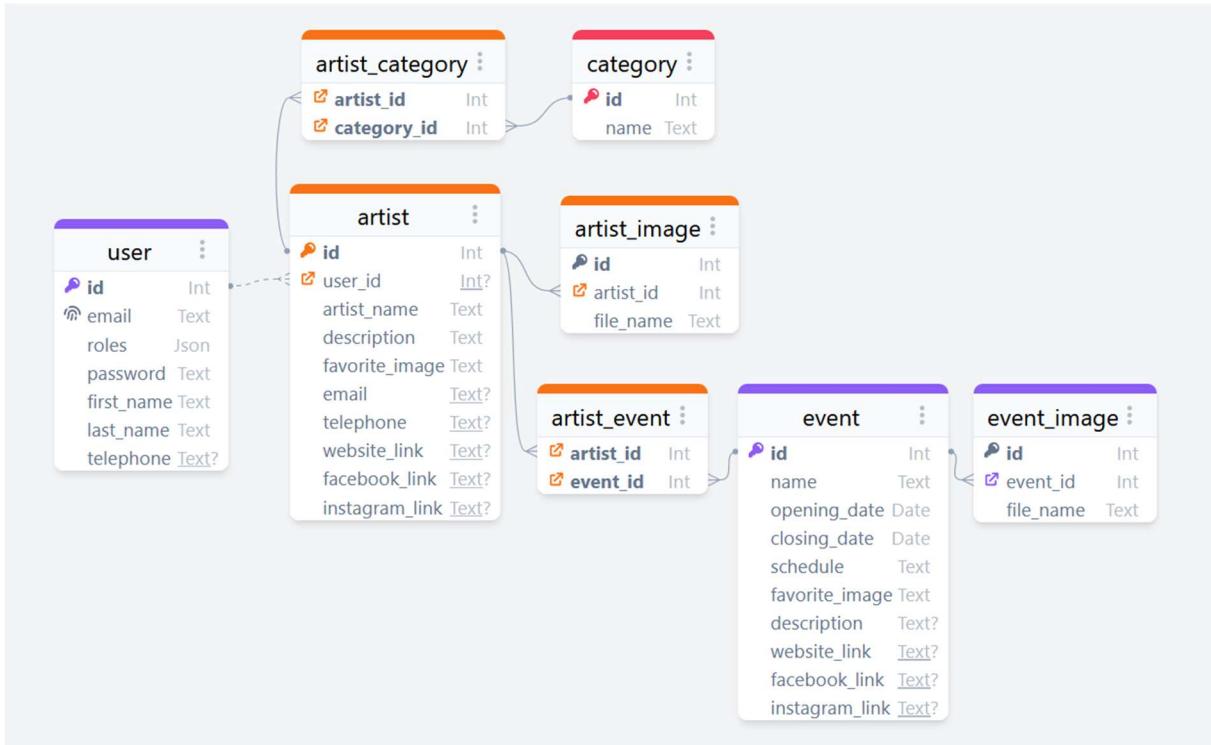
5.2.1 MCD (Modèle Conceptuel de Données)

J'ai réalisé le MCD (Modèle conceptuel de Données) grâce au logiciel UMLet.



5.2.2 MPD (Modèle Physique de Données)

Le MPD (Modèle Physique de Données) a été généré par l'outil en ligne Azimutt à partir du script de création de base de données avec Doctrine.



5.3 Maquettes

Ci-après la page d'accueil destinée aux visiteurs que j'ai conçue avec les versions desktop et mobile :

The image shows a side-by-side comparison of a website's desktop and mobile versions. The desktop view on the left displays a complex layout with multiple sections: 'Art et création en Partage' (header), 'Nos éducateurs' (with three cards), 'Une photo de la dernière expo' (with a large image and caption), 'Les artistes' (with four cards), and 'Qui sommes-nous?' (with three cards). The mobile view on the right shows a simplified layout where the content is reduced to a single column of cards, prioritizing the 'Qui sommes-nous?' section at the top.

6 Spécifications techniques du projet

Étant donné que ce projet est la création d'un site internet à partir de zéro ("from scratch"), sans environnement technique prédefini, j'ai eu la liberté de choisir les langages, les environnements de développement et les logiciels.

6.1 Matériel utilisé

Le projet a été réalisé en distanciel, depuis mon domicile.

Le matériel utilisé est composé de :

- Un PC portable équipé du système d'exploitation Windows 11 Famille
- Un écran secondaire, un clavier et une souris

6.1.1 Langages utilisés

La partie Front-end est réalisée en **HTML5, CSS et Javascript**.

La partie Back-end est réalisée avec **PHP** version 8.1.1.

La base de données est conçue en **SQL** avec **MySQL** comme système de gestion de bases de données.

6.1.2 Environnements de développement

L'environnement de développement intégré choisi est **Visual Studio Code**, une solution open source adaptée à la programmation PHP. Il s'agit d'un IDE polyvalent et extensible offrant un large éventail de fonctionnalités puissantes pour améliorer la productivité. Parmi ses atouts, on trouve la reconnaissance automatique des langages de programmation, tels que PHP, HTML, JavaScript et CSS, avec des fonctionnalités d'autocomplétion facilitant la rédaction du code.

Pour la gestion des différentes versions de développement du site, j'ai utilisé Git.

Le serveur de développement web est **Laragon**, qui regroupe nativement un ensemble d'outils tels que : **Apache** (serveur http), PHP, MySQL et **HeidiSQL**. J'ai utilisé HeidiSQL comme outil d'administration de base de données.

6.1.3 Frameworks utilisés

J'ai opté pour **Symfony** version 6.3, un framework PHP orienté objet (Open Source) qui est actuellement l'un des frameworks PHP les plus populaires, reconnu pour sa documentation complète et ses fonctionnalités avancées. Symfony repose sur le modèle Modèle – Vue – Contrôleur (MVC), utilisant le moteur de templates Twig et l'ORM(Object-Relational Mapping) Doctrine. L'utilisation de Symfony CLI (Command Line Interface), un ensemble d'outils en ligne de commande, a simplifié le développement et la gestion des projets Symfony, par exemple avec des commandes telles que `symfony server:start` ou `symfony doctrine:migration:migrate`.

Pour la partie CSS, j'ai découvert et mis en pratique **Tailwind CSS**, un framework CSS qui se distingue de Bootstrap par son approche axée sur la méthodologie de "conception utilitaire". Chaque classe CSS a une tâche spécifique, par exemple "text-center", "bg-blue-500", ou "p-4". De plus, Tailwind CSS génère un CSS léger et optimisé pour les performances, éliminant les classes inutilisées pour réduire la taille du fichier final.

6.1.4 Logiciels utilisés

Pour la modélisation UML (Unified Modeling Language), j'ai fait usage d'**UMLet**. Cet outil a été choisi pour sa simplicité, son intuitivité dans la création de diagrammes tels que les MCD et les diagrammes de cas d'utilisation.

Pour la conception visuelle des interfaces, j'ai choisi d'utiliser Figma, une application de conception et de prototypage d'interfaces utilisateur modulaire et adaptable aux évolutions en cours de projet.

7 Compétences du référentiel et réalisations

7.1 CP 1 Maquetter une application

Conformément aux spécifications du chapitre du cahier des charges, je présente ici la page d'accueil destinée aux visiteurs de l'application, maquettée à l'aide de Figma. J'ai créé des versions pour les écrans de bureau (Desktop) et les appareils mobiles (cf 5.3 Maquettes).

J'ai opté pour Figma, une plateforme de conception d'interface utilisateur en ligne. Figma est modulaire et adaptable aux évolutions en cours de projet, ce qui m'a permis d'obtenir un aperçu plus précis du rendu visuel souhaité. Bien que mon utilisation de Figma ait été initialement maladroite en raison de ma découverte de cet outil, il m'a tout de même permis de fournir un aperçu visuel du site à Mme Gerlache dès le début du projet.

La maquette que j'ai réalisée inclut un en-tête (header) avec une image de fond et une barre de navigation, qui varie en fonction des droits des utilisateurs. Ensuite, le corps de la page comprend différentes sections, notamment les événements, les artistes et la section "Qui sommes-nous". Pour une meilleure distinction visuelle des différentes sections, j'ai inséré des images entre elles.

Enfin, j'ai créé un pied de page (footer) qui reprend une partie de la navigation, un lien vers les mentions légales (portant sur la propriété intellectuelle et les droits d'auteur) ainsi que le copyright.

Voici la maquette de la page d'accueil pour un visiteur en version desktop :

Desktop- Accueil

7.2 CP 2 Réaliser une interface web statique et adaptable

7.2.1 Tailwind CSS

Dans le cadre de mon projet, j'ai choisi d'utiliser Tailwind CSS du côté Front-end. Tailwind CSS est un framework CSS "utility-first" qui simplifie la conception des éléments visuels du site web. Plutôt que de créer des styles personnalisés à partir de zéro, Tailwind CSS fournit un ensemble complet de classes CSS préconçues que l'on peut appliquer directement dans le code, simplifiant ainsi la gestion des styles pour chaque composant ou page, tout en prenant en charge une mise en page responsive.

N'ayant jamais utilisé Tailwind CSS, j'ai entrepris des recherches pour comprendre son installation et son fonctionnement. Sa documentation offre des instructions spécifiques pour son intégration avec Symfony. Pour installer Tailwind CSS, il a été nécessaire d'utiliser le bundle Webpack Encore, un outil développé par Symfony pour simplifier la gestion des assets (fichiers CSS, JavaScript, images, etc.) dans les applications Symfony. Cette installation est facilitée par la commande en terminal suivante : `composer require symfony/webpack-encore-bundle`. Ensuite, pour installer Tailwind CSS, les étapes consistent à exécuter les commandes suivantes : `npm install -D tailwindcss postcss postcss-loader autoprefixer` puis `npx tailwindcss init -p`. Enfin, pour visualiser les modifications en direct pendant la phase de développement, il suffit de lancer `npm run watch`.

Dans le fichier `base.html.twig`, les fonctions `encore_entry_link_tags('app')` et `encore_entry_script_tags('app')` génèrent les balises `link` et `script` appropriées pour inclure les feuilles de style CSS et les scripts JavaScript.

L'utilisation de Tailwind CSS permet de définir directement les mises en forme en fonction des media queries dans les balises. Des classes préconçues sont disponibles pour différentes tailles d'écran (sm, md, lg, etc.), et l'absence de classe préconçue indique par défaut une classe applicable aux écrans d'une largeur inférieure à 640px (format mobile).

Pour illustrer ce concept, prenons l'exemple de la page d'accueil (cf Annexe 1). Les cartes (cards) des artistes sont affichées en une seule colonne sur les écrans mobiles (inférieurs à 640px), en deux colonnes à partir de 640px, en trois colonnes à partir de 1024px et en quatre colonnes à partir de 1280px :

```
<div class="grid grid-cols-1 gap-y-10 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 xl:gap-x-4">
```

Les cartes des artistes et événements ont été élaborées à partir de composants gratuits disponibles sur le site de TailwindUI, qui propose une variété de composants directement compatibles avec TailwindCSS.

7.2.2 Formulaires

La génération de la partie Front des formulaires repose sur les fichiers de création de formulaires (« Type ») et leurs fonctions `buildForm` associées. Ces fichiers définissent les champs de formulaire en fonction du code fourni, notamment le nom du champ, le libellé, le type, le placeholder, etc.

La conception de formulaires de cette manière garantit une maintenabilité accrue. Toute modification apportée à un champ spécifique est automatiquement répercutée dans tous les fichiers concernés, simplifiant ainsi le processus de gestion.

En outre, cette approche facilite l'application de contraintes telles que required, regex, ou constraints (par exemple mimeTypes). Elle permet également de définir des mises en page spécifiques, comme l'option expanded pour un style de case à cocher ou multiple pour une sélection multiple.

Pour illustrer ceci, prenons l'exemple du champ « catégorie » dans le fichier ArtistType.php :

```
->add('categories', EntityType::class,[  
    'class' => Category::class,  
    'choice_label' => 'name',  
    'label' => 'Catégories : ',  
    'placeholder' => 'Sélectionnez une catégorie',  
    'required' => false,  
    'expanded' => true, //checkbox style  
    'multiple' => true, // Enable multiple selections  
])
```

La mise en forme des formulaires pour les fiches artiste et les événements a nécessité le plugin '@tailwindcss/forms'. Ce plugin permet une stylisation cohérente et esthétique des formulaires tout en conservant la flexibilité offerte par Tailwind CSS.

7.3 CP 3 Développer une interface utilisateur web dynamique

Navigation :

L'affichage de la navigation du header évolue dynamiquement en fonction de l'utilisateur grâce à des conditions. La partie gauche des onglets est commune à tous les utilisateurs, tandis que celle de droite varie en fonction des critères suivants dans les templates Twig :

- {%- if app.user%}
-> condition permettant de vérifier si l'utilisateur est connecté
- {%- if app.user.artist %}
-> condition permettant de vérifier si l'utilisateur connecté possède déjà une fiche artiste
- {%- if is_granted("ROLE_ADMIN") %}
-> condition permettant de vérifier si l'utilisateur dispose des droits d'administrateur

L'utilisation de ces conditions permet une personnalisation dynamique de l'interface en fonction des caractéristiques de l'utilisateur, offrant ainsi une expérience utilisateur adaptée à ses droits.

Les différentes configurations permettent à l'utilisateur d'accéder rapidement et facilement aux pages personnelles ou de gestion.

L'identifiant de l'utilisateur connecté est rappelé à droite de la déconnexion.

Ci-dessous, les différentes configurations de la barre de navigation en fonction de l'utilisateur :





Vous pourrez trouver une partie du code de la navigation en [Annexe 2](#)

Boutons de gestion :

Les boutons de gestion du contenu, tels que la création, la suppression ou la modification d'événements et d'artistes, bénéficient également de cet affichage conditionnel en fonction des droits de l'utilisateur. La gestion de ces conditions est réalisée dans les templates Twig. Cette approche garantit la conformité aux fonctionnalités définies dans l'expression des besoins.

Affichage aléatoire des artistes en page d'accueil :

Sur la page d'accueil, une fonctionnalité aléatoire est implémentée, affichant quatre artistes différents à chaque recharge de la page. Cette approche assure une présentation équitable et variée des artistes dès l'ouverture de la page.

Filtrage des artistes :

Dans la page regroupant tous les artistes, un filtre de recherche par nom (entier ou partiel) et de sélection par catégorie permet une mise à jour dynamique de l'affichage en fonction des critères sélectionnés. (cf 9.1 [Filtre des artistes](#))

Affichage des images en plein écran :

L'utilisation de Javascript est intégrée pour gérer la fonction « LightBox » des images associées aux artistes et aux événements. Cette lightBox se présente comme une fenêtre contextuelle qui s'affiche au-dessus du contenu existant de la page Web. Elle obscurcit le contenu de la page en arrière-plan pour focaliser l'attention de l'utilisateur sur le contenu de la Lightbox. Elle peut être fermée à l'aide d'un bouton dédié. La Lightbox offre la possibilité d'afficher une image d'artiste ou d'événement en plein écran tout en permettant, une fois fermée, de visualiser l'ensemble des images associées en miniatures. (cf Annexe 3)

7.4 CP 5 Créer une base de données

Durant la phase de développement Backend, j'ai créé de la base de données de l'application. Cette étape a impliqué la conception du modèle conceptuel de données (MCD).

Cela a permis de définir avec précision la structure de la base de données, y compris la gestion des tables et des collections ainsi que les relations complexes entre elles.

Le projet étant réalisé avec Symfony et son ORM Doctrine, j'ai défini les entités et leurs propriétés dont j'aurais besoin et qui formeront les tables dans la base de données:

- User (`id`, email, roles, password, firstName, lastName, telephone) : représentant une personne inscrite sur le site
- Artist (`id`, artistName, description, favoriteImage, email, telephone, websiteLink, facebookLink, instagramLink) : représentant une fiche d'artiste
- Category (`id`, name) : représentant une catégorie d'art (par ex : peinture, sculpture...)
- Event (`id`, name, openingDate, closingDate, schedule, favoriteImage, description, websiteLink, facebookLink, instagramLink) : représentant un événement/une exposition

- ArtistImage (id, fileName) : représentant une image ou une photo d'une œuvre de l'artiste
- EventImage (id, fileName) : représentant une photo prise lors de l'événement

Le choix d'avoir créé deux classes séparées, ArtistImage et EventImage, au lieu d'une seule peut sembler incohérent. En effet, toutes les deux comportent actuellement les mêmes propriétés (id et fileName). Néanmoins, ce choix s'explique par le fait que ce projet est une première version et qu'il est amené à évoluer peu. La cliente n'a pas une vision précise de son évolution, ce qui ne m'a pas permis de réaliser un modèle conceptuel idéal et définitif. Dans l'idée principale de l'évolution, il y aurait d'un côté, une image de l'œuvre de l'artiste regroupant des informations spécifiques comme le poids de l'œuvre et son prix et, de l'autre côté, une image liée à un événement permettant au public d'ajouter des commentaires. J'ai donc préféré séparer les deux entités en vue de leur évolution incertaine plutôt que de les regrouper au sein d'une seule classe Image qui aurait été moins malléable selon les futurs changements.

Ensuite j'ai établi les relations, les cardinalités, les quantités minimum et maximum d'occurrences entre chaque table et ce dans les deux sens :

- Relation User-Artist :

Un user peut avoir au minimum 0 artist et au maximum 1 artist => 0,1

Un artist peut appartenir au minimum et au maximum à 1 user=>1

C'est une relation One-To-One.

- Relation Artist-Category :

Un artist appartient au minimum à 0 category et au maximum plusieurs category => 0,N

Une category peut être appliquée au minimum à 0 artist et au maximum à plusieurs artists=> 0,N

C'est une relation Many-To-Many.

- Relation Artist-Event :

Un artist peut participer au minimum à 0 event et au maximum plusieurs event => 0,N

Un event peut avoir au minimum à 0 artist et au maximum à plusieurs artists=>0,N

C'est une relation Many-To-Many.

- Relation Artist-ArtistImage :

Un artist peut avoir au minimum 0 artistImage et au maximum plusieurs artistImage => 0,N

Une artistImage appartient au minimum et au maximum à 1 artist =>1

C'est une relation One-To-Many.

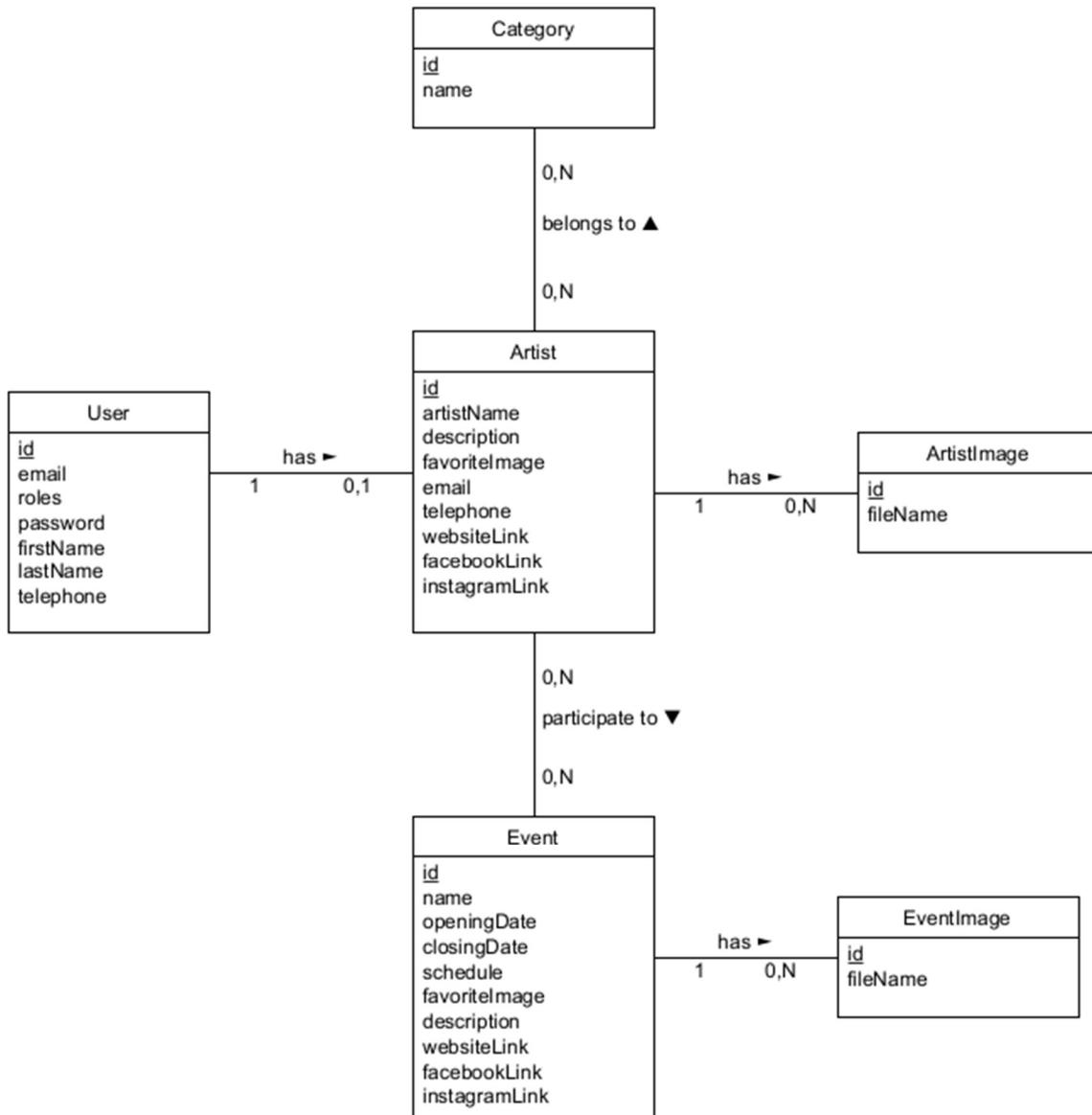
- Relation Event-EventImage :

Un event peut avoir au minimum 0 eventImage et au maximum plusieurs eventImage => 0,N

Une eventImage appartient au minimum et au maximum à 1 event =>1

C'est une relation One-To-Many.

Une fois cet exercice terminé, je me suis servi de UMLet afin de réaliser le MCD :



Vous trouverez dans la description fonctionnelle le modèle physique de données généré partir du script de création de base de donnée avec Doctrine (cf 5.2.2 MPD (Modèle Physique de Données)).

Les relations Many-to-Many sont gérées par Doctrine grâce à la création de tables de jonction, comme illustré dans la création de la table `artist_category`.

En l'absence d'utilisation de Doctrine, j'aurais écrit le script de création de cette table ainsi :

```

CREATE TABLE artist_category (
    artist_id INT NOT NULL,
    category_id INT NOT NULL,
    PRIMARY KEY(artist_id, category_id),
    CONSTRAINT FK_ArtistCategory_Artist FOREIGN KEY(artist_id) REFERENCES artist(id),
    CONSTRAINT FK_ArtistCategory_Category FOREIGN KEY(category_id) REFERENCES category(id)
);
  
```

Dans ce script :

- artist_id et category_id sont les clés étrangères faisant référence aux entités Artist et Category respectivement.
- La clé primaire (PRIMARY KEY) est composée de ces deux clés étrangères, assurant l'unicité des relations.
- Les contraintes étrangères (CONSTRAINT FK_ArtistCategory_Artist et CONSTRAINT FK_ArtistCategory_Category) assurent l'intégrité référentielle entre les tables.

7.5 CP 6 Développer les composants d'accès aux données

La gestion de la base de données s'est effectuée en toute fluidité grâce à l'utilisation de Doctrine, l'ORM intégré à Symfony. Voici comment j'ai traité l'accès aux données dans le cadre du projet :

1- Configuration de la base de données :

La configuration de la base de données a été effectuée dans le fichier «`.env.local` ». Cela permet de définir les variables d'environnement spécifiques au développement sans perturber l'environnement de production.

```
DATABASE_URL="mysql://root:@127.0.0.1:3306/artcreationpartage?serverVersion=8.0.32&charset=utf8mb4"
```

2- Utilisation de Doctrine :

- Création des entités :

À l'aide de Symfony CLI, j'ai employé la commande `symfony console make:entity` pour créer les entités, leurs propriétés, et spécifier les relations entre elles. Cela génère les classes PHP correspondantes ainsi que leurs repositories.

- Création de la base de données :

J'ai créé la base de données via la commande `symfony console doctrine:database:create` dans le terminal.

- Gestion des contraintes avec les annotations :

Les contraintes, telles que les contraintes d'unicité, de nullité ou de type sont gérées directement dans le fichier de l'entité à l'aide des annotations Doctrine. Ces annotations définissent les règles de validation pour l'insertion, la modification, et la suppression en base de données.

Étant donné l'utilisation de PHP 8.1, les annotations Doctrine et les contraintes de validation Symfony sont définies en tant qu'attributs de classe ou de propriété, directement dans le code source de l'entité. Les « assert » permettent de donner des contraintes de validation pour l'insertion/modification/suppression en base de données ainsi que des messages personnalisés à l'utilisateur.

Vous pouvez trouver une partie du fichier `Artist.php` en Annexe 4

- Génération des migrations :

Pour chaque modification de la base de données, j'ai créé une migration en utilisant la commande `symfony console make:migration`. Cette étape génère un fichier de migration dans le répertoire dédié du projet. Ce fichier contient les instructions pour effectuer les modifications de base de données spécifiques telles que la création de tables, l'ajout de colonnes, etc.

- Exécution des migrations :

Enfin, j'ai migré les données vers la base de données avec la commande `symfony console doctrine:migration:migrate`. Symfony parcourt les fichiers de migration, les exécute dans l'ordre, et applique les modifications à la base de données.

3- Getters et Setters automatiques :

Doctrine génère automatiquement les getters et setters (accesseurs et mutateurs) pour chaque propriété de la classe. Cela simplifie la manipulation des données au sein de l'application.

Cette approche m'a permis de gérer efficacement la structure de la base de données, les relations entre entités et les opérations courantes d'accès aux données

4- Repository

Les repositories jouent un rôle essentiel dans la gestion des données entre l'application et la base de données. Un repository agit comme un intermédiaire entre le contrôleur et la base de données. Il simplifie la récupération et la manipulation des données stockées en utilisant une interface orientée objet au lieu de requêtes SQL directes.

Le repository est généralement créé en héritant de la classe `EntityRepository` fournie par Symfony qui offre diverses méthodes couramment utilisées, telles que `find()`, `findAll()`, `findOneBy()`, `findBy()`, etc.

L'utilisation de repositories contribue à rendre le code plus modulaire et à encapsuler la logique d'accès aux données, favorisant ainsi une meilleure organisation et maintenance du code.

7.6 CP 7 Développer la partie back-end d'une application web ou web mobile

7.6.1 CRUD (Create, Read, Update, Delete)

Pendant la phase de développement de la partie Back-end, j'ai mis en place un back office complet offrant toutes les fonctionnalités de CRUD pour gérer les données liées aux entités Artist, Category, Event, et User.

J'ai utilisé la commande Symfony CLI `symfony console make:crud` pour générer automatiquement le CRUD pour chaque entité.

Pour chaque CRUD d'entité, les éléments suivants sont automatiquement générés :

- Contrôleur (ex: `CategoryController`) avec des fonctions CRUD spécifiques et les routes associées
- Constructeur de formulaire (ex: `CategoryType`) pour la gestion des opérations Create et Update.
- Vues (templates) regroupées dans un dossier dédié à chaque entité.

Par exemple, en utilisant la commande `symfony console make:crud category`, j'ai créé le CRUD pour l'entité Category. Cela a généré le `CategoryController` avec les routes, le `CategoryType` (cf [Annexe 5](#)) et les vues associées.

Cette approche automatisée a considérablement accéléré le processus de création du back office, tout en assurant une cohérence et une structure uniforme pour chaque entité.

Une fois les éléments de CRUD créés, je peux les modifier en fonction de mon application.

Dans un premier temps, pour chaque entité, j'ai modifié le constructeur de formulaire afin de donner aux champs les paramètres attendus, tels que le type, le label, les attributs, contraintes... (cf `CategoryType` en [Annexe 5](#))

Puis j'ai personnalisé le contrôleur qui regroupe les différentes fonctions du CRUD.

Pour illustrer la création d'une catégorie, examinons la fonction de création d'une nouvelle catégorie. Dans cette fonction, je crée une nouvelle instance de la catégorie. Ensuite, la méthode `createForm` génère un formulaire avec les champs appropriés en utilisant `CategoryType`, tandis que "handleRequest" gère la récupération de la requête. Si le formulaire est soumis et valide, je récupère les données du nom de la catégorie, le convertis en majuscules (uppercase) en utilisant l'encodage UTF-8 (pour prendre en compte les accents) avec la fonction `mb_strtoupper()`, et enfin, je réassigne ce nouveau nom via `setName()`. Avec `persist`, j'enregistre cet objet de catégorie et, avec `flush`, j'exécute l'enregistrement en insérant cette catégorie dans la base de données.

```

25     #[Route('/creation', name: 'app_category_new', methods: ['GET', 'POST'])]
26     public function new(Request $request, EntityManagerInterface $entityManager): Response
27     {
28         $category = new Category();
29         $form = $this->createForm(CategoryType::class, $category);
30         $form->handleRequest($request);
31
32         if ($form->isSubmitted() && $form->isValid()) {
33             $name = $form->get('name')->getData();
34             $upperName = mb_strtoupper($name, 'UTF-8');
35             $category->setName($upperName);
36             $entityManager->persist($category);
37             $entityManager->flush();
38
39             return $this->redirectToRoute('app_category_index', [], Response::HTTP_SEE_OTHER);
40         }
41
42         return $this->render('category/new.html.twig', [
43             'category' => $category,
44             'form' => $form,
45         ]);
46     }

```

Fonction de création d'une catégorie (partie du `CategoryController`)

Enfin, j'ai géré les templates générés pour chaque entité.

Exemple de la liste des templates générés pour la catégorie :



7.6.2 Inscription et connexion

En ce qui concerne l'inscription et la connexion des utilisateurs, j'ai débuté en créant l'entité utilisateur à l'aide de la commande en terminal `symfony make:user`. Cela déclenche une configuration automatique dans le fichier `security.yaml`, définissant, entre autres, la propriété unique utilisée pour l'identification de l'utilisateur (dans mon projet, j'ai opté pour l'email) ainsi qu'un système de hachage du mot de passe. J'ai maintenu le hachage 'auto' dans le paramètre `password_hashers`, car il choisit automatiquement l'algorithme le plus sécurisé disponible.

Pour la gestion de l'authentification (login), j'ai utilisé *symfony console make:auth* pour générer le contrôleur (SecurityController) avec ses méthodes de login et logout, ainsi que le fichier Twig dédié à la connexion.

Enfin, pour le processus d'inscription (registration), j'ai fait appel à la commande *symfony console make:registration-form*. Cette commande génère le contrôleur (avec le hachage de mot de passe), le formulaire et le fichier Twig associés, permettant ainsi la création de nouvelles instances d'utilisateurs.

7.6.3 Rôles et hiérarchie

Afin de réguler l'accès à certaines pages, j'ai utilisé des rôles pour définir les permissions des différents utilisateurs de l'application :

- ROLE_USER pour les utilisateurs connectés
- ROLE_ADMIN pour les administrateurs

Les simples visiteurs (ou les utilisateurs non connectés) du site n'ont pas de rôle car ils n'ont aucun droit de gestion.

Dans le fichier security.yaml, j'ai configuré une hiérarchie des rôles (role_hierarchy) permettant aux administrateurs d'avoir également le ROLE_USER.

```
access_control:
    - { path: ^/admin, roles: ROLE_ADMIN }

role_hierarchy:
    ROLE_USER: [ROLE_USER]
    ROLE_ADMIN: [ROLE_ADMIN, ROLE_USER]
```

J'y ai également configuré l'« access-control » afin de sécuriser l'ensemble des pages d'un contrôleur réservées aux administrateurs. Ainsi, les chemins commençant par « admin » dans leur URL ne sont accessibles que pour les utilisateurs ayant le ROLE_ADMIN. Cette technique a été utilisée pour sécuriser le back-office des catégories avec #[Route('/admin/categorie')] déclaré au-dessus de la classe CategoryController.

J'ai géré la restriction d'accès à certaines pages grâce à des annotations dans les contrôleurs :

- #[IsGranted("ROLE_USER")] pour la création/modification de fiche artiste et l'affichage/modification de son profil
- #[IsGranted("ROLE_ADMIN")] pour la création/modification d'événements, l'inscription (register)/affichage/modification d'utilisateurs

7.6.4 Autres contrôleurs et Twig

J'ai créé d'autres contrôleurs ainsi que leur templates Twig associés afin de gérer les pages d'accueil, contact, « à propos » et « appel à candidature ».

Le contrôleur de la page d'accueil permet la récupération et l'affichage des événements et des artistes.

8 Veille technique et sécurité

La documentation de Symfony propose toute une section consacrée à la sécurité sur sa page d'accueil. Symfony est un framework qui intègre de nombreuses fonctionnalités de sécurité. Vous trouverez des exemples et des explications tout au long de ce dossier.

Parmi les fonctionnalités de sécurité, on peut citer :

- Gestion des utilisateurs et des rôles (cf [Rôles et hiérarchie](#))
-)
- Authentification et hachage des mots de passe (cf [Inscription et connexion](#))
- Protection contre les failles CSRF (Cross-Site Request Forgery). Pour toutes les méthodes delete (artiste, événement, utilisateur et catégorie) et celles concernant le profil, login et register, j'ai utilisé la sécurisation par csrf-token. Je l'ai également utilisée pour la suppression des images des galeries (artiste et événements).

Ci-dessous le formulaire de suppression d'une image de la galerie d'un artiste :

```
templates > artist > _delete_image_form.html.twig
You, maintenant | 1 author (You)
1  <form method="post" action="{{path('app_artist_delete_image', {'id':image.id}) }}"
2  onsubmit="return confirm('Voulez-vous supprimer cette image ?');">
3  |   <input type="hidden" name="_token" value="{{ csrf_token('delete' ~ image.id) }}"/>
4  |   <button class="text-sm text-red-800 underline hover:text-red-600">Supprimer</button>
5  </form>
artist/_delete_image_form.html.twig
```

- Protection contre les injections SQL : Symfony utilise Doctrine, un ORM qui aide à éviter les injections SQL en paramétrant les requêtes de manière sécurisée en utilisant des paramètres plutôt que d'incorporer directement des valeurs dans les requêtes SQL (cf [9.1.3 ArtistRepository.php](#))
- Protection contre les injections de scripts (XSS) : Symfony fournit des fonctions d'échappement pour éviter les attaques XSS en nettoyant les données avant de les afficher
Ex : dans <p>{{message}}</p>,
si « message » contenait <script>alert('XSS!');</script>,
il serait converti en <p><script>alert('XSS!');</script></p>
- Validation des données : Symfony propose des outils de validation pour garantir que les données soumises par les utilisateurs sont correctes et sécurisées tels que les Assert des entités (cf [7.5 CP 6 Développer les composants d'accès aux données](#)) ou l'utilisation de contraintes directement dans les formulaires dont voici un exemple dans le builder de formulaire d'événement :

```
$builder
    ->add('name', TextType::class, [
        'label'=> 'Nom de l\'événement :',
        'required' => true,
        'constraints' => [
            new Assert\NotBlank(['message' => 'Le nom est obligatoire.']),
        ],
    ])
) You, le mois dernier * Modification des entités, Type et views
Champ du nom dans EventType.php
```

En plus de ces sécurisations, j'ai mis en place une fonction de validation des données dans les contrôleurs des artistes, des événements et des utilisateurs, permettant de détecter les problèmes après la soumission du formulaire mais avant l'enregistrement en base de données.

Exemple de la fonction checkEvent dans EventController :

```
public function checkEvent(Event $event): array
{
    $errors = [];
    //get data from event
    $name = $event->getName();
    $openingDate = $event->getOpeningDate();
    $closingDate = $event->getClosingDate();
    $schedule = $event->getSchedule();

    //Check constraints
    if (!$name) {
        $errors[] = 'Le nom est obligatoire.';
    }
    if ($closingDate < $openingDate) {
        $errors[] = 'La date de clôture doit être postérieure à la date d\'ouverture.';
    }
    if (!$schedule) {
        $errors[] = 'Les horaires sont obligatoires.';
    }

    return $errors;
}
```

J'ai également introduit des conditions ponctuelles, telle que la condition qui empêche un utilisateur de créer plus d'une fiche artiste.

Exemple dans ArtistController.php :

```

38     #[IsGranted("ROLE_USER")]
39     #[Route('/creation', name: 'app_artist_new', methods: ['GET', 'POST'])]
40     public function new(Request $request, EntityManagerInterface $entityManager): Response
41     {
42         $user = $this->getUser();
43
44         if ($user->getArtist() !== null) {
45             throw new AccessDeniedException('Vous avez déjà créé une fiche artiste.
46             | Vous ne pouvez pas en posséder plusieurs.');
47         }
48         $artist = new Artist();
49         $form = $this->createForm(ArtistType::class, $artist);
50         $form->handleRequest($request);
51

```

Enfin, des messages Flash de succès ou d'erreur ont été ajoutés pour informer l'utilisateur sur la validation de son formulaire et l'ajout d'images.

Exemple dans ArtistController.php :

```

$entityManager->persist($artist);
$entityManager->flush();
$this->addFlash('success', 'L\'artiste a été ajouté avec succès.');

```

9 Autres réalisations

Dans le cadre de ce projet, j'ai développé quatre fonctionnalités spécifiques contribuant à l'enrichissement de l'expérience utilisateur et à la facilitation du développement et du test de l'application :

- Filtre des artistes :
Filtre permettant d'afficher les artistes en fonction d'un terme de recherche et/ou d'une catégorie sélectionnée.
- Téléchargement d'images :
Possibilité d'uploader des images pour l'image de présentation de l'artiste et de l'événement, ainsi que le téléchargement d'images pour la galerie d'un artiste et celle d'un événement.
- Affichage en plein écran :
Mécanisme d'affichage d'images en plein écran de type Lightbox.
- Fichier de Fixtures :
Fichier avec données d'exemples permettant de faciliter le processus de développement et de test.

9.1 Filtre des artistes

Dans la page d'affichage des artistes, j'ai implémenté un filtre permettant de restreindre l'affichage des artistes en fonction d'un terme de recherche et/ou d'une catégorie sélectionnée. Pour la création de ce filtre, j'ai consulté plusieurs tutoriels.

Le système de filtrage repose sur l'utilisation d'un formulaire.

9.1.1 SearchData.php

Suite à la création des entités Artist et Category, j'ai établi un dossier Filter dans le répertoire src, contenant un fichier SearchData.php. Ce fichier renferme un "DataModel" représentant les données associées à ce formulaire, c'est-à-dire les informations liées à la recherche.

```
src > Filter > SearchData.php > ...
You, il y a 4 semaines | 1 author (You)
1 <?php You, il y a 4 semaines * Filtre de
2
3 namespace App\Filter;
4
5 use App\Entity\Category;
6
7 class searchData
8 {
9     public ?string $q = null;
10    public ?Category $category = null;
11 }
```

SearchData.php

La propriété \$q symbolise le mot-clé utilisé pour la recherche. Cette propriété est de type string, car elle sera saisie par l'utilisateur sous forme d'une chaîne de caractères. La propriété \$category est de type Category et sera choisie par l'utilisateur parmi la liste des catégories disponibles.

9.1.2 ArtistFilterForm.php

Suite à cela, j'ai créé le constructeur de formulaire dans le fichier ArtistFilterForm.php. (cf Annexe 6)

La fonction buildForm me permet de structurer le formulaire en incluant :

- Le champ de recherche 'q', présenté sous la forme d'un champ texte facultatif.
- Le champ 'category', représenté par une liste déroulante des catégories basées sur leurs noms, également facultatif.

La fonction configureOptions sert à définir certaines options :

- SearchData est défini comme classe de données.
- La méthode par défaut est Get (si l'utilisateur souhaite partager sa recherche via l'URL).
- La protection csrf est désactivée, étant donné qu'il s'agit d'un formulaire de recherche.

9.1.3 ArtistRepository.php

Puis dans le fichier ArtistRepository.php, j'ai créé la fonction findSearch. Cette fonction permet de récupérer les artistes en fonction du mot-clé et/ou de la catégorie.

J'ai initié un objet de requête en utilisant la méthode createQueryBuilder('a'). En premier lieu, j'ai sélectionné les tables des artistes ('a') et des catégories ('c') avec une jointure LEFT JOIN, ce qui signifie que je récupère tous les artistes ainsi que les catégories correspondantes s'il y en a. Si un artiste n'a pas de catégorie, il sera affiché sans catégorie (catégorie nulle).

La requête SQL est construite progressivement en fonction des données 'q' et 'category'. J'ai vérifié la présence de données envoyées via le formulaire pour le mot-clé. En cas de présence, j'ai ajouté une condition à la requête en utilisant `andWhere` pour rechercher les entités dont le champ `artistName` est similaire à la valeur de "q" (une recherche partielle, contenant le mot-clé). Cette condition est sécurisée en utilisant des paramètres pour éviter les injections SQL.

De même, j'ai vérifié la présence de données pour la catégorie. Si c'est le cas, une autre condition est ajoutée à la requête, vérifiant si l'entité associée "category" est membre de la collection de catégories associée à l'artiste. Cela permet de filtrer les entités en fonction de la catégorie sélectionnée.

Une fois que toutes les conditions ont été ajoutées à la requête, la fonction appelle `getQuery()` pour obtenir un objet `QueryDoctrine`. Enfin, elle utilise `getResult()` pour exécuter la requête et renvoyer les résultats.

```
public function findSearch(SearchData $search)
{
    $query = $this
        ->createQueryBuilder('a')
        ->select('a', 'c')
        ->leftjoin('a.categories', 'c');

    if (!empty($search->q)) {
        $query = $query
            ->andWhere('a.artistName LIKE :q')
            ->setParameter('q', "%{$search->q}%");
    }

    if (!empty($search->category)) {
        $query = $query
            ->andWhere(':category MEMBER OF a.categories')
            ->setParameter('category', $search->category);
    }

    return $query->getQuery()->getResult();
}
```

Fonction `findSearch()` du `ArtistRepository.php`

9.1.4 ArtistController.php

Dans la fonction `index` pour l'affichage des artistes, j'ai créé un nouvel objet `SearchData` et un formulaire de filtre de la classe `ArtistFilterForm`.

Les données soumises par l'utilisateur dans le formulaire sont vérifiées et traitées (`handleRequest`), puis stockées dans l'objet de type `SearchData`. Ensuite, la recherche des artistes concernés est effectuée grâce à la méthode `findSearch` sur l'`ArtistRepository`.

Enfin, les résultats sont stockés dans la variable `$artists` qui est passée à la vue `artist/index.html.twig` avec le formulaire de filtre (affiché grâce à `createView`).

```

#[Route('/', name: 'app_artist_index', methods: ['GET'])]
public function index(ArtistRepository $artistRepository, Request $request): Response
{
    $data = new SearchData();
    $form = $this->createForm(ArtistFilterForm::class, $data);
    $form->handleRequest($request);
    $artists = $artistRepository->findSearch($data);
    return $this->render('artist/index.html.twig', [
        'artists' => $artists,
        'form' => $form->createView()
    ]);
}

```

Affichage des artistes en fonction du filtre dans ArtistController.php

9.1.5 Artist/index.html.twig

Le fichier artist/index.html.twig gère la partie vue de l'affichage de l'ensemble des artistes. Le filtre est inclus depuis artist/_filter.html.twig

Les artistes sont listés sous forme de cartes (depuis artist/card.html.twig) présentant l'image représentative de l'artiste (propriété favoriteImage de Artist), son nom d'artiste et ses catégories. (cf [Annexe 7](#))

9.2 Téléchargement d'images

Pour permettre aux utilisateurs connectés ou aux administrateurs de télécharger leurs images (fiche artiste/événement ou galeries), j'ai mis en place un système de téléchargement de fichiers.

Vous trouverez les détails de cette fonctionnalité dans le chapitre [10 - Description d'une situation de travail ayant nécessité une recherche à partir de site anglophone](#).

9.3 Lightbox

J'ai mis en place une fonctionnalité d'affichage d'images en plein écran (souvent nommée Lightbox) afin de permettre aux utilisateurs de mieux visualiser les images des galeries photo des artistes et des événements sans impacter une vision globale de ladite galerie.

Cette lightBox se présente comme une fenêtre contextuelle qui s'affiche au-dessus du contenu existant de la page Web. Elle obscurcit le contenu de la page en arrière-plan pour focaliser l'attention de l'utilisateur sur le contenu de la Lightbox et peut être fermée à l'aide d'un bouton dédié.

Cette fonctionnalité est créée grâce à l'association de Javascript, CSS et des fichiers Twig.

(cf Annexe 3 et la partie consacrée à la **Erreur ! Source du renvoi introuvable.** dans le chapitre 7.3 [CP 3 Développer une interface utilisateur web dynamique](#)

9.4 Fixtures

Afin de donner un aperçu réaliste du rendu de l'application à la cliente, j'ai élaboré un fichier de fixtures AppFixtures.php contenant les données et informations obtenues auprès de celle-ci. Ces informations étaient de formats variés, provenant de discussions téléphoniques, d'e-mails, d'envois de photos et de fichiers PDF via WebTranfert. Lorsqu'il manquait des informations, j'ai créé des données fictives.

Pour créer cette fixture, j'ai utilisé le bundle DoctrineFixturesBundle, disponible dans la documentation des bundles sur le site de Symfony.

Dans le terminal de VSCode, j'ai utilisé la commande `composer require --dev orm-fixtures`.

Dans la fonction `load` du fichier `AppFixtures.php`, j'ai ensuite créé les différents événements, catégories, utilisateurs et artistes, en veillant aux relations entre eux.

J'ai également veillé à créer des administrateurs pour pouvoir tester l'application.

Une fois le fichier créé, j'ai pu charger mes données avec la commande `symfony console doctrine:fixtures:load`. Cette commande interagit avec la base de données pour insérer les données spécifiées dans les fichiers de fixtures, utilisant Doctrine pour effectuer ces opérations.

Vous pouvez voir deux parties de ce fichier de fixtures dans [l'Annexe 8](#).

10 Description d'une situation de travail ayant nécessité une recherche à partir de site anglophone

Tout au long de mon projet, j'ai effectué de nombreuses recherches sur des sujets que je ne connaissais pas ou maîtrisais peu. Parmi eux, ceux qui ont demandé le plus de temps sont :

- TailwindCSS
- La Lightbox
- Le téléchargement d'images

Je vais vous présenter ici la partie concernant le téléchargement d'images.

Afin de mettre en place le système de téléchargement d'images (upload), j'ai consulté plusieurs tutoriels sur YouTube, notamment :

- Upload et gestion d'images multiples (Symfony 6) par Nouvelle Techno (<https://www.youtube.com/watch?v=axbLC9PqzfE&t=3362s>)
- Symfony6 Techwall #45 Les formulaires upload d'image de Tech Wall (https://www.youtube.com/watch?v=Yd_qvZpD-L4&t=221s)

Au final, très peu de vidéos n'utilisaient pas de bundle tels que VichUploaderBundle. J'ai également consulté la documentation de Symfony, notamment les pages « FileType Field » (<https://symfony.com/doc/current/reference/forms/types/file.html>) et « How to Upload Files » (https://symfony.com/doc/current/controller/upload_file.html). Cette dernière était très claire, c'est pourquoi je me suis fortement appuyée dessus.

Pour le choix des MIME (Multipurpose Internet Mail Extensions) pour les formats d'images, je me suis référée à la page :

https://developer.mozilla.org/fr/docs/Web/HTTP/Basics_of_HTTP/MIME_Types

Voici l'exemple pour télécharger l'image de présentation de l'artiste (propriété `favoriteImage` de `Artist`) lors de la création d'une fiche artiste.

J'ai ajouté un champ de type `FileType` dans le builder du `ArtistType` en spécifiant les contraintes attendues :

- la taille maximum ('`maxSize`')
- les formats autorisés (jpeg, png et webp)
- le message d'erreur en cas de format non adapté ('`mimeTypesMessage`')

```
->add('favoriteImage', FileType::class, [
    'label' => 'Image du profil de l\'artiste: Télécharger une image
(jpeg, png, webp - taille max: 1024K) puis cliquer sur "Valider" ',
    'mapped' => false,
    'required' => false,
    'constraints' => [
        new File([
            'maxSize' => '1024k',
            'mimeType' => [
                'image/jpeg',
                'image/png',
                'image/webp',
            ],
            'mimeTypeMessage' => 'Merci de télécharger une image au format valide',
        ])
    ],
])
```

Puis, dans le contrôleur ArtistController (ci-dessous en partie), j'ai récupéré les données concernant le fichier téléchargé depuis le formulaire avec `getData`.

Si une image est soumise (la soumission et la validation du formulaire sont traitées plus haut dans le code), un nom de fichier unique est créé pour le fichier avec `md5(uniqid())` suivi de l'extension récupérée grâce à `guessExtension`. Ce nom est stocké dans la variable `$file`.

L'image est alors copiée dans le répertoire spécifié grâce à `getParameter('artist_images_directory')` et porte le nouveau nom du fichier `$file`.

Une fois l'image copiée avec succès, le nouveau nom du fichier est stocké dans la base de données avec `setFavoriteImage($file)`, la propriété `favoriteImage` étant de type `string`.

Si aucune image n'est soumise, une image par défaut est associée pour permettre un affichage correct de la card de l'artiste.

L'instance de l'artiste est ensuite persistée et stockée en base (flush).

```
//Favorite Image
//Get images uploaded
$favoriteImage = $form->get('favoriteImage')->getData();
//return a string: temporary filename in the file Temp of the App
if ($favoriteImage) {
    //Generate a new unique filename
    $file = md5(uniqid()) . '.' . $favoriteImage->guessExtension();

    //copy the file in uploads directory
    $favoriteImage->move(
        $this->getParameter('artist_images_directory'),
        $file
    );
    //Stock image in database
    $artist->setFavoriteImage($file);
} else {
    // default image
    $defaultImage = 'default.png';
    $artist->setFavoriteImage($defaultImage);
}
```

Dans le cas de l'ajout d'images pour la galerie, je crée d'abord une nouvelle instance d'ArtistImage nommée \$artistImage. Ensuite, j'attribue le nouveau nom à la propriété fileName, et j'ajoute cet objet à la collection d'ArtistImage de l'artiste.

```
//Stock image in database
$artistImage = new ArtistImage();
$artistImage->setFileName($file);
$artist->addArtistImage($artistImage);
```

Des processus similaires sont établis pour la modification de l'artiste ainsi que pour la création et la modification de l'événement.

Les répertoires où sont stockées les images téléchargées via l'application sont spécifiées dans le fichier services.yaml.

```
parameters:
  artist_images_directory: '%kernel.project_dir%/public/assets/uploads/artist_img'
  event_images_directory: '%kernel.project_dir%/public/assets/uploads/event_img'
```

11 Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment

Voici un extrait de la page « How to Upload Files » issu de la documentation de Symfony (https://symfony.com/doc/current/controller/upload_file.html) suivi de sa traduction en français : (les parties comportant du code ont été supprimées pour plus de clarté)

How to Upload Files

 Edit this page



Instead of handling file uploading yourself, you may consider using the [VichUploaderBundle](#) community bundle. This bundle provides all the common operations (such as file renaming, saving and deleting) and it's tightly integrated with Doctrine ORM, MongoDB ODM, PHPCR ODM and Propel.

Imagine that you have a `Product` entity in your application and you want to add a PDF brochure for each product. To do so, add a new property called `brochureFilename` in the `Product` entity:

.../...

Note that the type of the `brochureFilename` column is `string` instead of `binary` or `blob` because it only stores the PDF file name instead of the file contents.

The next step is to add a new field to the form that manages the `Product` entity. This must be a `FileType` field so the browsers can display the file upload widget. The trick to make it work is to add the form field as "unmapped", so Symfony doesn't try to get/set its value from the related entity:

.../...

Now, update the template that renders the form to display the new `brochure` field (the exact template code to add depends on the method used by your application to [customize form rendering](#)):

.../...

Finally, you need to update the code of the controller that handles the form:

.../...

Now, create the `brochures_directory` parameter that was used in the controller to specify the directory in which the brochures should be stored:

Comment télécharger des fichiers

Au lieu de gérer vous-même le téléchargement de fichiers, vous pouvez envisager d'utiliser le bundle communautaire VichUploaderBundle. Ce bundle fournit toutes les opérations communes (comme renommer, sauvegarder et supprimer un fichier) et il est étroitement intégré avec l'ORM Doctrine, MongoDB ODM, PHPCR ODM et Propel.

Imaginez que vous ayez une entité Produit dans votre application et que vous vouliez ajouter une brochure PDF pour chaque produit. Pour ce faire, ajoutez une nouvelle propriété appelée brochureFilename dans l'entité Produit :

Notez que le type de la colonne brochureFilename est une chaîne de caractères (string) au lieu de binary ou blob, car on stocke uniquement le nom du fichier PDF et non le contenu du fichier.

La prochaine étape consiste à ajouter un nouveau champ au formulaire qui gère l'entité Produit. Ce doit être un champ de type FileType pour que les navigateurs puissent afficher le widget de téléchargement de fichiers. L'astuce pour que ça fonctionne est d'ajouter le champ du formulaire en « unmapped », ainsi Symfony n'essaie pas de récupérer/modifier sa valeur à partir de l'entité liée :

Maintenant, mettez à jour le modèle de rendu du formulaire pour afficher le nouveau champ de brochure (le code exact du modèle à ajouter dépend de la méthode utilisée par votre application pour personnaliser le rendu du formulaire) :

Enfin, vous devez mettre à jour le code du contrôleur qui gère le formulaire :

Maintenant, créez le paramètre brochures_directory qui a été utilisé dans le contrôleur pour spécifier le répertoire dans lequel les brochures doivent être stockées :

12 Jeu d'essai

Pour réaliser ce jeu d'essai, je vais vous présenter la création et la modification d'une fiche artiste depuis un utilisateur connecté.

Les artistes déjà présents sont issues des fixtures. Les images utilisées pour la création de l'artiste fictif sont des images libres de droits et gratuites à télécharger issues de Pixabay.com.

D'autres vues de l'application sont disponibles en Annexe 9.

- ♦ J'ai au préalable créé un nouvel utilisateur nommé 'fictif' depuis le compte administrateur.
- ♦ Connectée sur le site en tant que l'utilisateur 'fictif' (identifiant par mail fictif@fictif.com), je clique sur « Crée une fiche artiste » dans la barre de navigation.

- ♦ Dans le formulaire, je crée exprès des erreurs sur les champs « téléphone » et « lien de votre site internet », les erreurs sont bien rapportées.

Créer une fiche Artiste

Nom d'artiste :

Artiste Fictif

Description :

La description de l'Artiste fictif.

Email :

artistefictif@fictif.com

Téléphone :

Le numéro de téléphone doit être au format français valide, par exemple, 01-23-45-67-89.

01

lien de votre site internet :

Veuillez saisir une URL valide commençant par http:// ou https://

artistefictif

lien de votre site Facebook :

https://www.facebook.com/exemple/

♦ Je complète ensuite le formulaire sans erreur :

Créer une fiche Artiste

Nom d'artiste :
Artiste Fictif

Description :
La description de l'Artiste fictif.

Email :
artistefictif@fictif.com

Téléphone :
01 02 03 04 05

lien de votre site internet :
http://www.exemple.com/

lien de votre site Facebook :
http://www.facebookexemple.com/

lien de votre site Instagram :
http://www.instagramexemple.com/

Catégories :
 PEINTURE
 SCULPTURE
 PLASTICIEN.NE
 POESIE
 PHOTOGRAPHIE
 MUSIQUE
 AUTRE

Image du profil de l'artiste: Télécharger une image (jpeg, png, webp - taille max: 1024K) puis cliquer sur "Valider"
[Parcourir...](#) tiger-2923186_1280.jpg

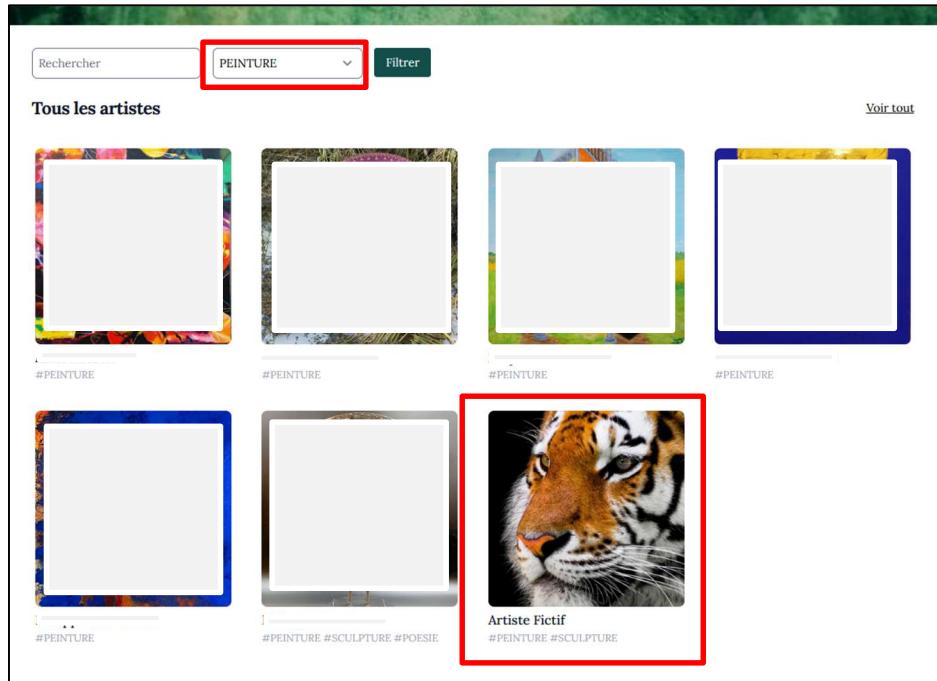
Télécharger une image (jpeg, png, webp - taille max: 1024K) puis cliquer sur "Valider"
[Parcourir...](#) cat-1045782_1280.jpg

Valider

♦ Le message Flash de succès de création de l'artiste apparaît sous le header, je suis bien renvoyée sur la page précédente et l'onglet de ma barre de navigation a changé de « Crée une fiche artiste » en « Ma fiche artiste »

The screenshot shows a website interface. At the top, there's a navigation bar with links: Accueil, A propos, Evénements, **Artistes**, Contact, Ma fiche artiste (which is highlighted with a red border), Mon profil, Déconnexion, and an email address fictif@fictif.com. Below the navigation, there's a banner with the text "Art et création en partage" and "Promotion d'artistes en tout domaine". A red box highlights the success message "L'artiste a été ajouté avec succès." in a light blue box. Below the message, there's a search bar, a dropdown menu for categories, and a "Filtrer" button. Underneath, there's a section titled "Tous les artistes" with four small preview boxes.

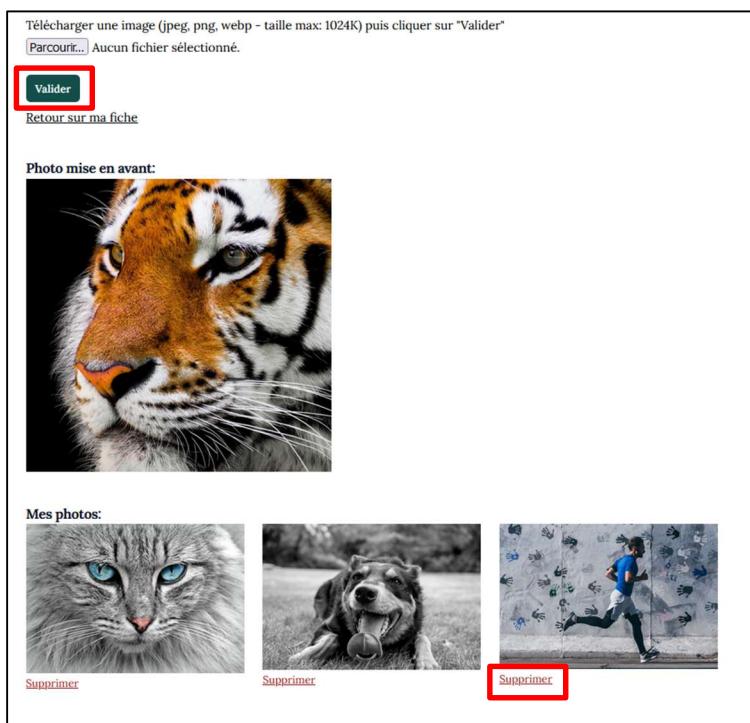
- ♦ Ma carte de fiche artiste ‘Artiste Fictif’ est bien présente dans la page des artistes et lors d’une sélection depuis le filtre de catégorie (ou de mot clé).



- ♦ Je clique sur la carte de l’artiste et j’arrive sur la fiche de l’artiste fictif avec toutes ses informations et ses images.

- ♦ Je clique sur « Modifier la fiche artiste » et arrive sur le formulaire de modification de l’artiste avec ses informations préremplies.

♦ Je rajoute des images (un chien et un coureur) dans la galerie de l'artiste via le formulaire et modifie la description de l'artiste (« La nouvelle description de l'Artiste Fictif »). Je clique sur Valider. Les messages Flash de succès apparaissent bien et l'ensemble des images de la galerie de l'artiste sont bien affichées en bas du formulaire de modification.



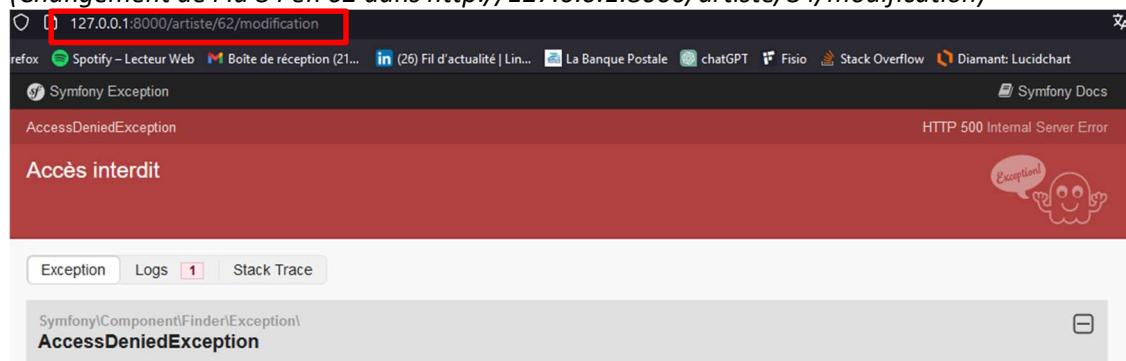
♦ Je clique sur « supprimer » sous l'image du coureur et une modale de confirmation apparait.



Après confirmation, l'image du coureur a bien disparu. Le message flash de confirmation de suppression apparaît.

♦ Si je tente, depuis l'URL, de modifier une fiche artiste qui n'est pas la mienne en tant qu' 'utilisateur fictif', l'accès à la page m'est refusé.

(*Changement de l'id 84 en 62 dans http://127.0.0.1:8000/artiste/84/modification*)



♦ Je me déconnecte de tout compte et vais sur l'application en simple visiteur. Sur la fiche de l'Artiste Fictif, l'image du chat et du chien sont bien présentes et celle du coureur absente. De même, le bouton de modification de la fiche artiste est effectivement absent.

The screenshot shows a profile page for 'Artiste Fictif'. At the top, it says '#PEINTURE #SCULPTURE'. Below that is a placeholder text 'La nouvelle description'. Under 'Contacts', there is a list of social media links: Email (mailto:artisteefictif@fictif.com), Telephone (01 02 03 04 05), Site internet (<http://www.example.com/>), Facebook (<http://www.facebookexample.com/>), and Instagram (<http://www.instagramexample.com/>). To the right of the text area is a large image of a tiger's face. Below the text area are two smaller images: one of a fluffy grey cat with blue eyes, and another of a brown and black dog lying down.

♦ Dans ma base de données :

Dans la table artist, les données de l'artiste fictif sont conformes à la création et aux modifications apportées (description) :

id	artist_name	description	email	telephone	website_link	facebook_link	instagram_link	user_id	favorite_image
84	Artiste Fictif	La nouvelle descr...	artisteefictif@ficti...	01 02 03 ...	http://www.example...	http://www.facebookexe...	http://www.instagrameme...	74	2538c195f56efc2ef7...

id	artist_name	description	email	telephone	website_link	facebook_link
84	Artiste Fictif	La nouvelle descr...	artisteefictif@ficti...	01 02 03 ...	http://www.example...	http://www.facebookexe...
<hr/>						
facebook_link	instagram_link	user_id	favorite_image			
http://www.facebookexe...	http://www.instagrameme...	74	2538c195f56efc2ef7...			

Dans la table user, l'utilisateur associé à cette fiche artiste est bien fictif@fictif.com. (notez le ROLE_USER et le mot de passe haché).

id	email	roles	password	first_name	last_name	telephone
74	fictif@fictif.com	["ROLE_USER"]	\$2y\$13\$IhcZ3yoUMfauaPAKS...	fictif	fictif	(NULL)

Dans la table artist_image, j'ai bien les deux images associées à la fiche artiste et dont le nom de fichier a été modifiée de façon unique.

id	artist_id	file_name
64	84	ac1cb36f86dd471f8339e4899ac058d4.jpg
65	84	50ff0a4eadba9bc96d5d300735312612.jpg

L'ensemble des données au niveau du front-end et de la base de données sont conformes aux attentes.

Bilan

Le projet présenté dans ce rapport est une application web réalisée pour l'association Art et Création en Partage qui permet de visualiser les événements organisés par l'association et les artistes y participant ainsi que la présentation de celle-ci.

Les artistes peuvent créer et modifier leur fiche de présentation et gérer la galerie photo des leurs œuvres.

Le back-office permet aux administrateurs de gérer les utilisateurs, les événements, les artistes et leurs catégories.

A ce jour, l'ensemble des fonctionnalités attendues dans l'expression des besoins sont présentes et fonctionnelles. Des fonctionnalités supplémentaires ont été ajoutées afin d'améliorer l'expérience utilisateur (filtre des artistes, affichage des images en plein écran...). La gestion du back-office par l'administrateur est simple mais intuitive pour un utilisateur sans connaissance informatique.

L'absence de tuteur technique fut la principale difficulté pour ce projet. En effet, Mme Gerlache, n'ayant aucune connaissance en développement, a pris le rôle de client dans le projet. Les problèmes rencontrés au cours du développement ont nécessité que je fasse de nombreuses recherches de résolution, lecture de documentations et de veille technologique. Certaines fonctionnalités m'ont donné du fil à retordre telles que le téléchargement des images et la lightbox que je n'avais jamais réalisé auparavant ou encore le filtre des artistes.

L'aspect positif fut que cela m'a donné une grande liberté dans la conception de l'application et m'a encouragé à une importante autonomie générale. Ce projet m'a permis également de découvrir de nouveaux outils comme Figma ou TailWind CSS que je voulais essayer car ce sont des outils modernes et largement utilisés par les développeurs.

Je n'ai pas pu terminer complètement le projet dans le temps imparti par la durée du stage. Parmi les principaux éléments à développer se trouvent : le téléchargement d'images en lot (et l'utilisation de VichUploaderBundle), l'amélioration du visuel des messages d'erreurs, la personnalisation des pages d'erreur, l'ajout de slug et régler les problèmes d'affichages en version mobile du header (ajout d'un menu burger) et du footer.

J'aurai aimé également effectuer le déploiement de l'application, une partie que je n'ai encore jamais eu l'occasion de mettre en œuvre.

Je vais tout faire pour terminer l'application pendant mon temps personnel car je dois, dès la fin du stage, continuer mon emploi à la Police Scientifique en attendant d'intégrer une alternance pour le diplôme de Concepteur d'Application. Je souhaite livrer à Mme Gerlache une application totalement fonctionnelle, esthétique et responsive, bref une application prête à l'emploi.

Conclusion

J'ai pris beaucoup de plaisir à travailler sur ce projet. La création d'une application à partir de zéro a représenté un défi que je suis ravie d'avoir relevé.

Cela a renforcé ma confiance en mes capacités d'apprentissage, d'autonomie, et de résilience. Cette expérience m'a permis de mieux comprendre le fonctionnement de Symfony, PHP et du développement en général. J'étais ravie d'apprendre de nouveaux logiciels, frameworks, outils et fonctionnalités.

Mon choix de reconversion dans le développement web s'est avéré très gratifiant et je suis plus motivée que jamais à continuer d'apprendre, de comprendre et de créer de nouvelles choses chaque jour.

Webographie

Les sites internet indiqués dans cette webographie sont les principaux sites utilisés pour ce projet mais ne constituent en rien une liste exhaustive.

Contacts de l'association Art et Création en Partage :

artetcreationenpartage@gmail.com

www.facebook.com/Barjachorsdessentiersbattus/

UMLet

www.umlet.com

Figma

www.figma.com

Tailwind CSS

www.tailwindcss.com

Procédure d'installation spécifique à Symfony : <https://tailwindcss.com/docs/guides/symfony>

Tutoriel de Nouvelle techno : <https://www.youtube.com/watch?v=I93ibDUMy0k>

Tutoriel de Grafikart : <https://www.youtube.com/watch?v=D6-g6JgiUIs>

Messages flash : <https://v1.tailwindcss.com/components/alerts>

Plugin pour formulaire : <https://github.com/tailwindlabs/tailwindcss-forms>

Composants de TailwindCSS : (<https://tailwindui.com/components>)

Symfony

Page d'accueil de la documentation : <https://symfony.com/doc/current/index.html>

Bundle de fixtures : <https://symfony.com/bundles/DoctrineFixturesBundle/current/index.html>

Filtre des artistes :

Tutoriel de Grafikart : <https://grafikart.fr/tutoriels/filtre-produit-symfony-1211>.

Téléchargement d'images :

<https://www.youtube.com/watch?v=axbLC9PqzfE&t=3362s>

https://www.youtube.com/watch?v=Yd_qvZpD-L4&t=221s

<https://symfony.com/doc/current/reference/forms/types/file.html>

https://symfony.com/doc/current/controller/upload_file.html

https://developer.mozilla.org/fr/docs/Web/HTTP/Basics_of_HTTP/MIME_Types

<https://pixabay.com/fr/>

Annexes

Table des matières :

Annexe 1 : Page d'accueil.....	46
Annexe 2 : Navigation	47
Annexe 3 : Lightbox	48
Annexe 4 : Partie de fichier PHP de la classe Artist.....	50
Annexe 5 : exemple de fichier de formulaire (« Type »).....	51
Annexe 6 : Constructeur de formulaire pour le filtre des artistes	52
Annexe 7 : Partie vue de l'affichage des artistes selon le filtre	53
Annexe 8 : Parties du fichier de fixtures concernant les utilisateurs et artistes.....	55
Annexe 9 : Pages de l'application depuis le navigateur de site web Firefox.....	57

Annexe 1 : Page d'accueil

`main/home.html.twig`

```

templates > main > home.html.twig
You, il y a 27 secondes | 1 author (You)
1  {% extends 'base.html.twig' %}
2
3  {% block title %}Art et Cr ation en partage
4  {% endblock %}
5
6  {% block body %}
7
8      <div
9          class="mx-auto px-6 py-16 sm:px-6 sm:py-8 lg:max-w-7xl lg:px-8">
10         <!--Nos  v nements-->
11         <section>...
12             </section>
13
14         <!--Picture 1-->
15         <section class="bg-cover bg-center h-80" style="background-image: url('/assets/images/barjac_header.jpg')">...
16             </section>
17
18         <!--Les artistes-->
19         <section>
20             <div class="py-20"> You, il y a 2 semaines * css page accueil
21                 <div class="flex items-center justify-between pb-10">
22                     <h2 class="text-2xl font-bold tracking-tight text-gray-900 ">
23                         Les artistes</h2>
24                     <a href="{{ path('app_artist_index') }}" class="underline hover:text-teal-800" title='Voir tous les artistes'>Voir tout</a>
25                 </div>
26
27                 <div class="grid grid-cols-1 gap-y-10 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 xl:gap-x-4">
28                     {% for artist in randomArtists %}
29                         {% include 'artist/_card.html.twig' %}
30                     {% endfor %}
31                 </div>
32             </div>
33         </section>
34
35         <!--Picture 2-->
36         <section class="bg-cover bg-center h-80" style="background-image: url('/assets/images/main_photo4.jpg')">
37             <div class="flex bg-black bg-opacity-30 bg-cover bg-center h-80 relative">
38                 <div class="text-white text-4xl font-bold absolute bottom-10 left-0 p-4">
39                     <p class="text-sm lg:text-xl pl-20">Barjac - Hors des sentiers battus 2023</p>
40                 </div>
41             </div>
42         </section>
43
44         <!--Qui sommes nous-->
45         <section>...
46             </section>
47         </div>
48     {% endblock %}
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116

```

Annexe 2 : Navigation

Navigation dans le header (partie droite des onglets)

inc/nav.html.twig

```

templates > inc > nav.html.twig
You, il y a 4 secondes | 1 author (You)

1   <nav>
2     <div class="flex flex-col lg:flex-row items-center justify-between text-gray-100 text-lg font-bold px-10">
3       <div class="flex lg:flex-row sm:flex-col flex-col">...
4       </div>
5
6       <div class="flex items-center lg:flex-row flex-col">
7
8         {% if app.user %}
9           <div class="flex flex-col lg:flex-row">
10             {% if is_granted("ROLE_ADMIN") %}
11               <div>
12                 <a href="{{ path('app_category_index') }}" class="hover:bg-teal-200 hover:text-black rounded-md px-4 py-2">Catégories</a>
13               </div>
14               <div>
15                 <a href="{{ path('app_register') }}" class="hover:bg-teal-200 hover:text-black rounded-md px-3 py-2">Inscription</a>
16               </div>
17               {% endif %}
18             </div>
19           <div>
20             {% if app.user.artist %}
21               <a href="{{ path('app_artist_show', {'id': app.user.artist.id} ) }}" class="hover:bg-teal-200 hover:text-black rounded-md px-4 py-2">Ma fiche artiste</a>
22             {% else %}
23               <a href="{{ path('app_artist_new') }}" class="hover:bg-teal-200 hover:text-black rounded-md px-4 py-2">Créer une fiche artiste</a>
24               {% endif %}
25             </div>
26           <div>
27             | <a href="{{ path('my_profile') }}" class="hover:bg-teal-200 hover:text-black rounded-md px-4 py-2">Mon profil</a>
28           <div>
29             | <a href="{{ path('app_logout') }}" class="hover:bg-teal-200 hover:text-black rounded-md px-4 py-2">Déconnexion</a>
30           <div class="text-gray-300 px-4 py-2 text-sm font-medium">
31             {{ app.user.userIdentifier }}
32           </div>
33         {% else %}
34           <div class="flex items-center lg:flex-row sm:flex-col flex-col">
35
36             <div>
37               <a href="{{ path('app_login') }}" class="hover:bg-teal-200 hover:text-black rounded-md px-3 py-2">Se connecter</a>
38             </div>
39           <div>
40             {% endif %}
41         </div>
42       </div>
43     </div>
44   </nav>

```

Annexe 3 : Lightbox

Partie Javascript pour la gestion de la Lightbox :

app.js

```
assets > JS app.js > ...
You, il y a 1 seconde | 1 author (You)
1
2 // any CSS you import will output into a single css file (app.css in this case)
3 import './styles/app.css';
4
5 /**Lightbox */
6 var images = document.querySelectorAll('.image-container img');
7 var modal = document.querySelector('.popup-image');
8 var closeButton = document.querySelector('.popup-image span');
9
10 images.forEach(function(image) {
11   image.addEventListener('click', function() {
12     modal.style.display = 'block';
13
14     var imageSrc = image.getAttribute('src');
15     var modalImage = modal.querySelector('img');
16     modalImage.setAttribute('src', imageSrc);
17
18   });
19 });
20
21 closeButton.addEventListener('click', function() {
22   modal.style.display = 'none';
23 });
24
```

Partie CSS :

app.css

```
17 ****lightbox*****
18 .image-container{
19   display:flex;
20   flex-wrap:wrap;
21   gap:30px;
22   justify-content:center;
23   padding:80px;
24
25 }
26
27 .image-container .image{
28   display:flex;
29   align-items:center;
30   justify-content:center;
31   height:250px;
32   width:350px;
33   /* border:5px solid #ffff; */
34   border-radius:10px;
35   box-shadow: 0 5px 10px #rgba(0,0,0,.2);
36   overflow : hidden;
37   cursor:pointer;
38 }
39
40 .image-container .image img{
41   max-height:100%;
42   width:100%;
43   object-fit: cover;
44   transition: .2s linear;
45 }
46
47
48 .image-container .image:hover img{
49   transform: scale(1.1);
50 }
51
52 .image-container .popup-image{
53   position:fixed;
54   top:0; left:0;
55   background:#rgba(0,0,0,.9);
56   height:100%;
57   width:100%;
58   z-index:100;
59   display:none;
60 }
61 .image-container .popup-image span{
62   position:absolute;
63   top:0; right:10px;
64   font-size:60px;
65   font-weight:bolder;
66   color: #ffff;
67   cursor:pointer;
68   z-index:100;
69 }
70 .image-container .popup-image img{
71   position:absolute;
72   top:50%; left:50%;
73   transform: translate(-50%, -50%);
74   border:5px solid #ffff;
75   border-radius: 5px;
76   height:100%;
77   width:auto;
78   object-fit: cover;
79   display: block;
80   margin: 0 auto;
81 }
82
83 @media (max-width:768px) {
84   .image-container .popup-image img{
85     width:95%;
86   }
87 }
```

Partie template pour l'affichage de l'artiste
artist/show.html.twig

```
templates > artist > show.html.twig
You, il y a 1 seconde | 1 author (You)
1  {% extends 'base.html.twig' %}

2
3  {% block title %}
4  |   {{artist.artistName}}
5  {% endblock %}

6
7
8  {% block body %}
9    <div class="pt-6 px-6">
10   >   <div class="flex justify-end pr-10">...
11
12
13
14    <!-- Artist info -->
15   <div class="mx-auto max-w-2xl px-4 pb-10 pt-10 sm:px-6 lg:grid lg:max-w-7xl lg:grid-cols-3
16   | lg:grid-rows-[auto,auto,1fr] lg:gap-x-8 lg:px-8 lg:pb-5 lg:pt-16">...
17
18
19    <!--buttons-->
20   <div class="flex justify-center space-x-4">...
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589 You, il y a 1 seconde * Uncommitted changes
```

Annexe 4 : Partie de fichier PHP de la classe Artist

Artist.php

```

13 #[ORM\Entity(repositoryClass: ArtistRepository::class)]
14 #[UniqueEntity(fields: ['artistName'], message: 'Ce nom d\'artiste est déjà pris.')]
15 class Artist
16 {
17     #[ORM\Id]
18     #[ORM\GeneratedValue]
19     #[ORM\Column]
20     private ?int $id = null;
21
22     #[ORM\Column(length: 255)]
23     #[Assert\NotBlank(message: 'Le nom d\'artiste est obligatoire')]
24     #[Assert\Length(min: 1, max: 255,
25         minMessage: 'Le nom d\'artiste doit faire au moins {{ limit }} caractères',
26         maxMessage: 'Le nom d\'artiste doit faire au plus {{ limit }} caractères')]
27     private ?string $artistName = null;
28
29     #[ORM\Column(type: Types::TEXT)]
30     #[Assert\NotBlank(message: 'Une description est obligatoire')]
31     private ?string $description = null;
32
33     #[ORM\Column(length: 255, nullable: true)]
34     #[Assert\Email(
35         message: 'L\'email n\'est pas valide',
36     )]
37     private ?string $email = null;
38
39     #[ORM\Column(length: 255, nullable: true)]
40     #[Assert\Regex(pattern: '/^0[1-9]([-. ]?[0-9]{2}){4}$/',
41         message: 'Le numéro de téléphone doit être au format français valide, par exemple, 01-23-45-67-89.')]
42     private ?string $telephone = null;
43
44     #[ORM\Column(length: 255, nullable: true)]
45     #[Assert\Url(message : "Veuillez saisir une URL valide commençant par http:// ou https://" )]
46     private ?string $websiteLink = null;
47
48     #[ORM\Column(length: 255, nullable: true)]
49     #[Assert\Url(message : "Veuillez saisir une URL valide commençant par http:// ou https://" )]
50     private ?string $facebookLink = null;
51
52     #[ORM\Column(length: 255, nullable: true)]
53     #[Assert\Url(message : "Veuillez saisir une URL valide commençant par http:// ou https://" )]
54     private ?string $instagramLink = null;
55
56     #[ORM\OneToOne(inversedBy: 'artist', cascade: ['persist'])] You, il y a 1 seconde * Uncommitted change
57     private ?User $user = null;
58
59     #[ORM\ManyToMany(targetEntity: Category::class, inversedBy: 'artists')]
60     #[Assert\NotBlank(message: 'Veuillez sélectionner au moins une catégorie')]
61     private Collection $categories;
62
63     #[ORM\ManyToMany(targetEntity: Event::class, inversedBy: 'artists',)]
64     private Collection $events;
65
66     #[ORM\OneToMany(mappedBy: 'artist', targetEntity: ArtistImage::class, cascade: ['persist', 'remove'])]
67     private Collection $artistImages;
--
```

Annexe 5 : exemple de fichier de constructeur de formulaire (« Type »)

categoryType.php

```
1 <?php      You, le mois dernier • Modification des entités, Type et views
2
3 namespace App\Form;
4
5 use App\Entity\Artist;
6 use App\Entity\Category;
7 use Symfony\Component\Form\AbstractType;
8 use Symfony\Component\Form\Extension\Core\Type\TextType;
9 use Symfony\Component\Form\FormBuilderInterface;
10 use Symfony\Component\OptionsResolver\OptionsResolver;
11
12 You, il y a 4 semaines | 1 author (You)
13 class CategoryType extends AbstractType
14 {
15     public function buildForm(FormBuilderInterface $builder, array $options): void
16     {
17         $builder
18             ->add('name', TextType::class, [
19                 'label'=> 'Nom de la catégorie : ',
20                 'required' => true,
21             ]);
22     }
23
24     public function configureOptions(OptionsResolver $resolver): void
25     {
26         $resolver->setDefaults([
27             'data_class' => Category::class,
28         ]);
29     }
30 }
```

Annexe 6 : Constructeur de formulaire pour le filtre des artistes

artistFilterForm.php

```
src > Form >  ArtistFilterForm.php >  ArtistFilterForm >  buildform
    You, il y a 2 semaines | 1 author (You)
1   <?php
2
3   namespace App\Form;
4
5   use App\Entity\Category;
6   use App\Filter\SearchData;
7   use Symfony\Bridge\Doctrine\Form\Type\EntityType;
8   use Symfony\Component\Form\AbstractType;
9   use Symfony\Component\Form\Extension\Core\Type\TextType;
10  use Symfony\Component\Form\FormBuilderInterface;
11  use Symfony\Component\OptionsResolver\OptionsResolver;
12
13  You, il y a 2 semaines | 1 author (You)
14  class ArtistFilterForm extends AbstractType
15  {
16      public function buildform(FormBuilderInterface $builder, array $options)
17      {
18          $builder
19              ->add('q', TextType::class,[
20                  'label'=>false,
21                  'required'=>false,
22                  'attr'=> [
23                      'placeholder'=> 'Rechercher'
24                  ]
25              ])
26              ->add('category', EntityType::class,[ You, il y a 4 semaines *
27                  'class' => Category::class,
28                  'choice_label' => 'name',
29                  'label' => 'Catégorie : ',
30                  'placeholder' => 'Toutes les catégories',
31                  'required' => false,
32              ]);
33      }
34
35      public function configureOptions(OptionsResolver $resolver)
36      {
37          $resolver->setDefaults([
38              'data_class' => SearchData::class,
39              'method' => 'GET',
40              'csrf_protection'=> false
41          ]);
42      }
43
44      public function getBlockPrefix()
45      {
46          return ''; //to clean the URL
47      }
48  }
```

Annexe 7 : Partie vue de l'affichage des artistes selon le filtre

artist/index.html.twig

```

templates > artist > index.html.twig
You, il y a 1 seconde | 1 author (You)
1  {% extends 'base.html.twig' %}

2
3  {% block title %}Liste des artistes
4  {% endblock %}

5
6  {% block body %}
7    <section>
8      <!--Les artistes-->
9      <div class="mx-auto max-w-2xl px-6 py-16 sm:px-6 sm:py-8 lg:max-w-7xl lg:px-8">
10        <div class="mb-6">
11          {% include 'artist/_filter.html.twig' %}
12        </div>
13        <div class="flex items-center justify-between pb-10">
14          <h2 class="text-2xl font-bold tracking-tight text-gray-900 ">Tous les artistes</h2>
15          <a href="{{ path ('app_artist_index')}}" class="underline hover:text-teal-800"
16            title='Voir tous les artistes'>Voir tout</a>
17        </div>

18        <div class="grid grid-cols-1 gap-x-6 gap-y-10 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 xl:gap-x-8">
19
20          {% for artist in artists %}
21            {% include 'artist/_card.html.twig'%}
22          {% else %}
23            <p>Aucun artiste trouvé</p>
24
25          {% endfor %}
26        </div>
27      </div>
28    </section>
29  {% endblock %}
30
31

```

artist/_filter.html.twig

```

templates > artist > _filter.html.twig
You, il y a 1 seconde | 1 author (You)
1  {{ form_start(form) }}
2  <div class="flex justify-start ">
3    <div >
4      {{ form_widget(form.q, {'attr':
5        |{'class': 'border p-2 rounded-md w-full focus:ring-2 focus:ring-teal-500 focus:outline-none'}) }}}
6    </div>
7    <div >
8      {{ form_widget(form.category, {'attr':
9        |{'class': 'border p-2 rounded-md pr-10 focus:ring-2 focus:ring-teal-500 focus:outline-none'}) }}}
10     </div>
11     <div >
12       <button type="submit" class="bg-teal-900 text-white px-4 py-2 rounded hover:bg-teal-500 ">Filtrer</button>
13     </div>
14   </div>
15   {{ form_end(form) }}
16

```

```
artist/_card.html.twig
templates > artist > _card.html.twig
You, maintenant | 1 author (You)
1 <a href="{{path ('app_artist_show', {'id':artist.id})}}" title='{{artist.artistName}}'
2 class="group flex justify-center">
3 <div>
4     <div class="h-[270px] w-[270px] overflow-hidden rounded-lg bg-gray-200">
5         
8     </div>
9     <h3 class="mt-1 text-lg font-medium text-gray-900">{{ artist.artistName }}</h3>
10    <div class="text-sm text-gray-400">
11        {% for category in artist.categories %}
12            #{{ category.name }}
13        {% endfor %}
14    </div>
15    </div>
16 </a> You, il y a 4 semaines • Début de mise en forme tailwind
```

Annexe 8 : Parties du fichier de fixtures concernant les utilisateurs et artistes

Partie de données pour un utilisateur avec fiche artiste dans AppFixtures.php

```
[  
    'firstName' => 'Ghislaine',  
    'lastName' => 'DEROUGE',  
    'email' => 'ghislaine_derouge@exemple.com',  
    'telephone' => '',  
    'artist' =>  
        [  
            'artistName' => 'Ghislaine DEROUGE',  
            'description' => "Installation de grands bustes en grès chamotté, avec  
                [REDACTED]  
                d'autres temps",  
            'email' => 'ghislaine_derouge@exemple.com',  
            'telephone' => '',  
            'websiteLink' => 'ghislainederouge.fr',  
            'facebookLink' => '',  
            'instagramLink' => '',  
            'categories' => ['SCULPTURE'],  
            'favoriteImage' => 'ghislaine_derouge_DSCN5123.JPG',  
        ],  
],
```

Partie du traitement des données dans AppFixtures.php

```
foreach ($userDatas as $userData) {
    $user = new User();
    $user->setFirstName($userData['firstName']);
    $user->setLastName($userData['lastName']);
    $user->setEmail($userData['email']);
    $user->setPassword($this->passwordHasher->hashPassword($user, '123456'));
    $user->setRoles(['ROLE_USER']);
    $user->setTelephone($userData['telephone']);

    // Create artist and associate to user
    if (isset($userData['artist'])) {
        $artistData = $userData['artist'];

        $artist = new Artist();
        $artist->setArtistName($artistData['artistName']);
        $artist->setDescription($artistData['description']);
        $artist->setEmail($artistData['email']);
        $artist->setTelephone($artistData['telephone']);
        $artist->setWebsiteLink($artistData['websiteLink']);
        $artist->setFacebookLink($artistData['facebookLink']);
        $artist->setInstagramLink($artistData['instagramLink']);
        $artist->setFavoriteImage($artistData['favoriteImage']);
        $artist->addEvent($event1);

        // Associate category to artist
        $categories = $artistData['categories'];
        foreach ($categories as $categoryName) {
            $category = $manager->getRepository(Category::class)->findOneBy(['name' => $categoryName]);
            if ($category) {
                $artist->addCategory($category);
            }
        }
    }

    $artist->setUser($user);
    $user->setArtist($artist);

    $manager->persist($user);
    dump('User créé : ' . $user->getLastName());
    dump('Artist créé : ' . $artist->getArtistName());
}
}
```

Annexe 9 : Pages de l'application depuis le navigateur de site web Firefox

Utilisateur non connecté



Nos événements

[Voir tout](#)

Barjac Hors des sentiers battus

Du 09/09/2023 au 10/09/2023

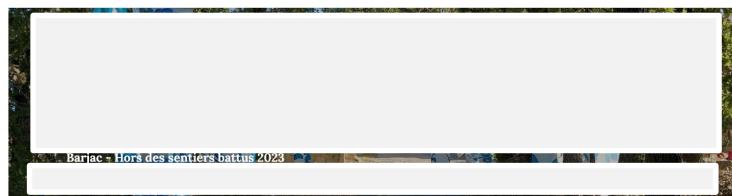
Barjac Hors des sentiers battus - 2023

description de l'événement 2023

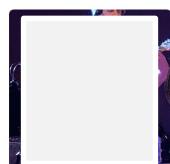
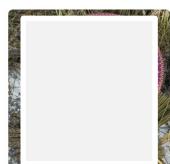
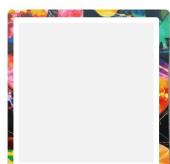
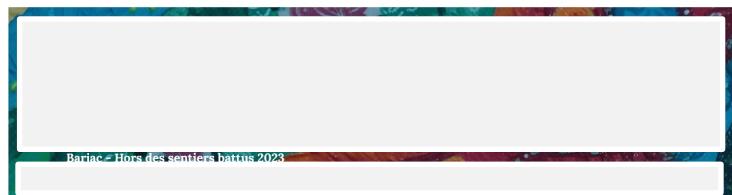
Du 28/10/2023 au 30/10/2023

Barjac 2024

Pour les amateurs de chocolats et autres friandises. Le chocolat [ʃɔkɔlat], terme d'origine mésoaméricaine Note 1, est un aliment plus ou moins sucré produit à partir de la...



Les artistes

[Voir tout](#)Poupl
#MUSIQUEAurélie MARTINEZ
#PEINTUREJean-Marc FRAISSE
#PEINTUREAnne MAURY
#PEINTURE

Qui sommes-nous?



A propos



Appel à candidature



Nous contacter

*Page d'accueil (utilisateur non connecté)*

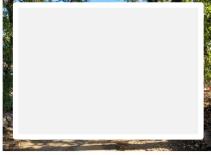
[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Se connecter](#)

Art et création en partage
Promotion d'artistes en tout domaine

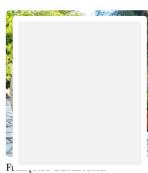
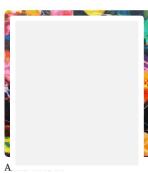
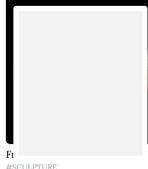
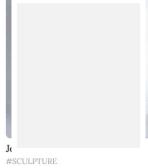
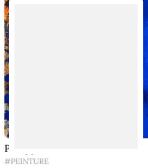
[Voir tous les événements](#)

Barjac Hors des sentiers battus - 2023
Du 09/09/2023 au 10/09/2023
Horaires : de 10h à 19h
description de l'événement 2023

Contacts:
Facebook: <https://www.facebook.com/Barjachorsdessentiersbattus/>



Les artistes

			
A1 #SCULPTURE	A2 #SCULPTURE	A3 #PEINTURE	A4 #SCULPTURE
			
A5 #PEINTURE	A6 #SCULPTURE	A7 #PEINTURE	A8 #SCULPTURE
			
A9 #SCULPTURE	A10 #SCULPTURE	A11 #POESIE	A12 #PEINTURE
			
A13 #SCULPTURE	A14 #PHOTOGRAPHIE	A15 #PHOTOGRAPHIE	A16 #PEINTURE
			
A17 #PEINTURE #SCULPTURE #POESIE	A18 #MUSIQUE	A19 #PLASTICIEN.NE	

[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Mentions légales et politique de confidentialité](#)

2023 | Art et création en partage

Affichage d'un événement

[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Se connecter](#)

Art et création en partage
Promotion d'artistes en tout domaine

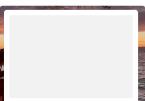
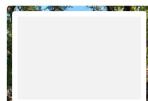
Barjac Hors des sentiers battus



Cela fait 2 ans que nous organisons une balade artistique à Barjac, petite ville du Gard où les artistes se déclinent.

Chaque année nous sélectionnons une vingtaine d'artistes pour cette rencontre conviviale entre artistes et le public qui a lieu durant tout un week-end début septembre.

En 2023, l'événement a pris ses quartiers dans l'espace privé, les artistes ont exposé tout à l'intérieur qu'à l'extérieur pour un week-end festif avec musique, bouffe et petite restauration.



Du 09/09/2023 au 10/09/2023

Barjac Hors des sentiers battus - 2023

description de l'événement 2023

Pour les amateurs de chocolats et autres friandises, Le chocolat [chocolat] tenu d'origine mésaméricaine, il est un aliment plus ou moins cuit par rapport à la forme brûlée. Celle-ci est déclinée sous de nombreuses formes plus ou moins élaborées et variées. La forme la plus connue est celle des tablettes de chocolat. Ces dernières sont également déclinées en plusieurs saveurs, comme la vanille, ou des morceaux de fruits, etc. Comme elles sont généralement vendues sous forme de tablette (boisson grise) au Mexique et en Amérique centrale, le chocolat se démocratise avec la révolution industrielle.

Barjac 2024

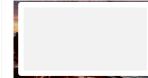
Du 29/09/2023 au 30/09/2023

Horaires : de 9h à 18h

Pour les amateurs de chocolats et autres friandises, Le chocolat [chocolat] tenu d'origine mésaméricaine, il est un aliment plus ou moins cuit par rapport à la forme brûlée. Celle-ci est déclinée sous de nombreuses formes plus ou moins élaborées et variées. La forme la plus connue est celle des tablettes de chocolat. Ces dernières sont également déclinées en plusieurs saveurs, comme la vanille, ou des morceaux de fruits, etc. Comme elles sont généralement vendues sous forme de tablette (boisson grise) au Mexique et en Amérique centrale, le chocolat se démocratise avec la révolution industrielle.

Contact

- Site internet: <http://www.artetcreationpartage.org/se/Chocolat>
- Facebook: <https://www.facebook.com/se/Chocolat>
- Instagram: <https://www.instagram.com/se/Chocolat>

[Voir tous les événements](#)

•

**Les artistes**Artistie Fictif
ARTISTE EN SCULPTURE

[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Mentions légales et politique de confidentialité](#)

Affichage d'un autre événement (fictif)

Mentions légales et politique de confidentialité
2023 | Art et création en partage

Affichage d'un autre événement (ajouté via l'application, hors fixtures)

[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Mentions légales et politique de confidentialité](#)

Les événements (dont celui de droite ajouté hors fixtures)

[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Se connecter](#)

Art et création en partage
Promotion d'artistes en tout domaine

A propos de nous

L'association Art et Crédit en Partage regroupe à la base deux artistes qui, soucieux de sortir des expositions classiques et conventionnelles, ont décidé de se regrouper en association selon la loi 1901. Le but de l'association est la création d'événements qui favorisent la promotion des artistes de toutes disciplines, de toutes origines et de toutes nationalités.

Tous trois, des personnes passionnées d'art et tout autant bâtardeusement au gré de leur envie de créer pour la création d'un événement qui a le plaisir aussi maintenant d'être baptisé "Barjac, hors des sentiers battus".

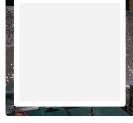
Nos artistes : Chaque année nous sélectionnons des artistes en fonction de nos coups de cœur, de la beauté de leur création, de l'essence de leurs œuvres, de la matière de leur technique, de leur regard original, du message apporté. Je ne nous laissez guider par nos émotions.

Le lieu : nous priviligeons les endroits atypiques ou non des salles d'exposition traditionnelles. Cela permet aux artistes de mettre leurs œuvres en valeur différemment et au public de découvrir des lieux, de prendre le temps de découvrir les œuvres des artistes.

Nous sommes toujours à la recherche de nouveaux partenaires pour nos événements. Si vous avez des idées, n'hésitez pas à nous contacter.

Barjac, hors des sentiers battus : à priori nos quartiers dans l'espace privé, les artistes ont exposé tout à l'intérieur qu'à l'extérieur pour un week-end festif avec musique, bouffe et petite restauration.

[Retour à l'accueil](#)



[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Mentions légales et politique de confidentialité](#)

Page "à propos de nous"

[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Se connecter](#)

Art et création en partage
Promotion d'artistes en tout domaine

Contact

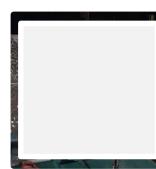
Pour nous contacter :

- Email : artetcreationpartage@gmail.com
- Tél : 0640906055, Françoise Gerleche, présidente de l'association

Nous sommes toujours à la recherche de nouveaux talents.

Si vous voulez faire partie des artistes pour la prochaine exposition, merci de lire l'appel à candidature.

[Retour à l'accueil](#)



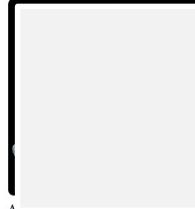
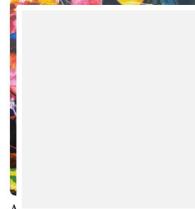
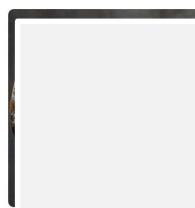
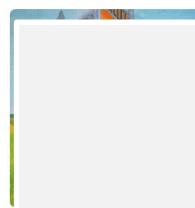
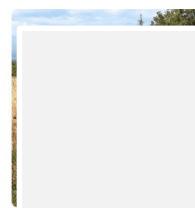
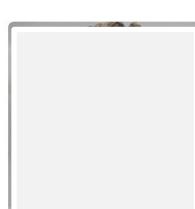
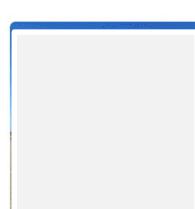
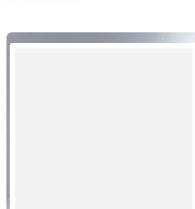
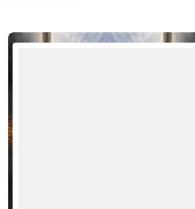
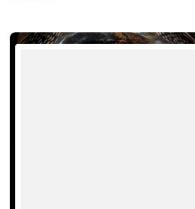
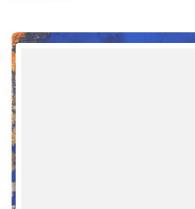
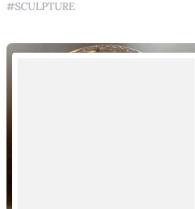
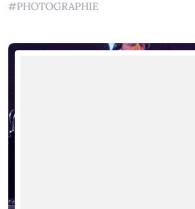
[Accueil](#) [A propos](#) [Événements](#) [Artistes](#) [Contact](#) [Mentions légales et politique de confidentialité](#)

Page de contact

Art et création en partage
Promotion d'artistes en tout domaine

Rechercher Toutes les catégories Se connecter

Tous les artistes [Voir tout](#)

 A #SCULPTURE	 B #SCULPTURE	 C #PEINTURE	 D #SCULPTURE
 E #PEINTURE	 F #SCULPTURE	 G #PEINTURE	 H #SCULPTURE
 I #SCULPTURE	 J #SCULPTURE	 K #POESIE	 L #PEINTURE
 M #SCULPTURE	 N #SCULPTURE	 O #PHOTOGRAPHIE	 P #PEINTURE
 Q #SCULPTURE	 R #MUSIQUE	 S #PHOTOGRAPHIE	 T #PLASTICIEN.NE
 U #PEINTURE #SCULPTURE #POESIE	 V #MUSIQUE	 W #PLASTICIEN.NE	 X Artiste Fictif #PEINTURE #SCULPTURE

Accueil A propos Evénements Artistes Contact Mentions légales et politique de confidentialité

2023 | Art et création en partage

Les artistes

Accueil A propos Evénements Artistes Contact Se connecter

Art et création en partage
Promotion d'artistes en tout domaine

[Tous les artistes](#)[Voir tout](#)

Artiste Fictif
#PEINTURE #SCULPTURE



Accueil A propos Evénements Artistes Contact Mentions légales et politique de confidentialité

2023 | Art et création en partage

Action du filtre (« fic » inscrit dans la barre de recherche)

Accueil A propos Evénements Artistes Contact Se connecter

Art et création en partage
Promotion d'artistes en tout domaine

[Voir tous les artistes](#)

Artiste Fictif

#PEINTURE #SCULPTURE

La nouvelle description

Contacts:

- * Email: artisefictif@fictif.com
- * Téléphone: 01 02 03 04 05
- * Site internet: <http://www.exemple.com/>
- * Facebook: <http://www.facebookexemple.com/>
- * Instagram: <http://www.instagramexemple.com/>

Événements participés :

[Barjac 2024](#)



Accueil A propos Evénements Artistes Contact Mentions légales et politique de confidentialité

2023 | Art et création en partage

Fiche artiste de « Artiste Fictif »



Connexion

Email:

Mot de passe:

Se connecter

Accueil A propos Evénements Artistes Contact Mentions légales et politique de confidentialité 2023 | Art et création en partage

Page de connexion



Mon profil

Pseudo: fictif

Nom: fictif

Téléphone:

E-mail: fictif@fictif.com

Mot de passe:

Confirmer le mot de passe:

Valider

Accueil A propos Evénements Artistes Contact Mentions légales et politique de confidentialité 2023 | Art et création en partage

Page du profil utilisateur (utilisateur connecté)

Utilisateur connecté



Modification de la barre de navigation en tant qu'utilisateur ayant une fiche artiste



Voir tous les artistes

Artiste Fictif

#PEINTURE #SCULPTURE

La nouvelle description

Contacts:

- + Email: artistefictif@fictif.com
- + Téléphone: 01 02 03 04 05
- + Site internet: <http://www.example.com/>
- + Facebook: <http://www.facebookexample.com/>
- + Instagram: <http://www.instagramexample.com/>

Événements participés :

Barjac 2024

Modifier la fiche artiste

Bouton « Modifier la fiche artiste »



Fiche artiste avec l'artiste en tant qu'utilisateur connecté



Affichage de photo en plein écran (Lightbox)



Modifier une fiche artiste

Nom d'artiste :

Artiste Fictif

Description :

La nouvelle description

Email :

artistefictif@fictif.com



Téléphone :

01 02 03 04 05

lien de votre site internet :

http://www.exemple.com/

lien de votre site Facebook :

http://www.facebookexemple.com/

lien de votre site Instagram :

http://www.instagramexemple.com/

Catégories :

- PEINTURE
- SCULPTURE
- PLASTICIEN.NE
- POESIE
- PHOTOGRAPHIE
- MUSIQUE
- AUTRE

Image du profil de l'artiste: Télécharger une image (jpeg, png, webp - taille max: 1024K) puis cliquer sur "Valider"

Parcourir... Aucun fichier sélectionné.

Télécharger une image (jpeg, png, webp - taille max: 1024K) puis cliquer sur "Valider"

Parcourir... Aucun fichier sélectionné.

[Valider](#)

[Retour sur ma fiche](#)

photo mise en avant:



Mes photos:



[Supprimer](#)



[Supprimer](#)

Page de modification d'une fiche artiste (l'artiste concerné en tant qu'utilisateur connecté)

The screenshot shows the top navigation bar with the title "Administrateur connecté" in a large, bold, black font. Below it is a banner with the text "Art et création en partage" and "Promotion d'artistes en tout domaine". The navigation menu includes links for Accueil, A propos, Evénements, Artistes, Contact, Catégories, Inscription, Crée une fiche artiste, Mon profil, Déconnexion, and an email address admin@admin.com.

Modification de la barre de navigation en tant qu'administrateur

The page title is "Liste des catégories". It displays a table of categories with columns "NOM" and "ACTION". The categories listed are PEINTURE, SCULPTURE, PLASTICIEN.NE, POESIE, PHOTOGRAPHIE, MUSIQUE, and AUTRE. Each row has a "Modifier / Supprimer" link under ACTION. At the bottom is a green button labeled "Crée une catégorie".

The footer contains links for Accueil, A propos, Evénements, Artistes, Contact, Mentions légales et politique de confidentialité, and the year 2023 | Art et création en partage.

Liste des catégories (administrateur connecté)

The page title is "Crée une catégorie". It has a form with a "Nom" input field containing "PEINTURE", a "Valider" button, and a "Retour aux catégories" link.

Mise en valeur de l'onglet au survol (hover) (pour tout onglet et tout utilisateur)

The footer contains links for Accueil, A propos, Evénements, Artistes, Contact, Mentions légales et politique de confidentialité, and the year 2023 | Art et création en partage.

Création de catégorie (administrateur connecté)

The page title is "Crée une catégorie". It has a form with a "Nom" input field containing "PEINTURE", a "Valider" button, and a "Retour aux catégories" link.

Modifier la catégorie PEINTURE

Nom
PEINTURE

[Retour aux catégories](#)

The footer contains links for Accueil, A propos, Evénements, Artistes, Contact, Mentions légales et politique de confidentialité, and the year 2023 | Art et création en partage.

Modification de catégorie (administrateur connecté)



Inscription

Prénom

Nom

Téléphone

Email

Mot de passe

Valider

Accueil A propos Evénements Artistes Contact Mentions légales et politique de confidentialité 2023 | Art et création en partage

Page d'inscription (accessible uniquement pour l'administrateur)

Voir tous les événements

Barjac Hors des sentiers battus - 2023

Du 09/09/2023 au 10/09/2023
Horaires : de 10h à 19h
description de l'événement 2023

Contacts:
+ Facebook: <https://www.facebook.com/Barjachorsdessentiersbattus/>

Modifier **Supprimer l'événement**

Page d'un événement- boutons de modification et de suppression accessible uniquement pour l'administrateur

Artiste Fictif
#PEINTURE #SCULPTURE

La nouvelle description

Contacts:
+ Email: artiste fictif@fictif.com
+ Téléphone: 01 02 03 04 05
+ Site internet: <http://www.example.com/>
+ Facebook: <http://www.facebookexample.com/>
+ Instagram: <http://www.instagramexample.com/>

Événements participés:
[Barjac 2024](#)

Modifier la fiche artiste **Supprimer l'artiste**

Page d'un artiste - bouton de suppression accessible uniquement pour l'administrateur



Modifier un événement

[Retour sur l'événement](#)

Nom de l'événement :

Barjac Hors des sentiers battus - 2023

Date d'ouverture de l'événement :

09 / 09 / 2023



Date de clôture de l'événement :

10 / 09 / 2023



Créneau horaire :

de 10h à 19h



Description :

description de l'événement 2023



lien du site internet :

<http://www.example.com/>

lien du site facebook :

<https://www.facebook.com/Barjachorsdessentiersbattus/>

lien du site instagram :

<https://www.instagram.com/exemple/>

Artistes :



Image mise en avant:

[Parcourir...](#) Aucun fichier sélectionné.

Télécharger une image (jpeg, png, webp - taille max: 1024K) puis cliquer sur "Valider".

[Parcourir...](#) Aucun fichier sélectionné.[Valider](#)

Photo mise en avant:



Toutes les photos de l'événement:

[Supprimer](#)[Retour sur l'événement](#)[Accueil](#)[A propos](#)[Événements](#)[Artistes](#)[Contact](#)

Mentions légales et politique de confidentialité

2023 | Art et création en partage



Nos événements

Voir tout



Barjac hors des sentiers battus - 2023

Description de l'événement 2023



Les artistes

Voir tout



Qui sommes-nous?

Mentions légales et politique de confidentialité
2023 | Art et création en partage

Page d'accueil en version mobile (utilisateur non connecté)

Page de modification d'un événement (accessible uniquement pour l'administrateur)