

Projet Web

Formulaire et interface de suivi de candidatures de groupes musicaux dans un festival de musique.

Résumé

Un festival de musique associatif lance chaque année un appel à candidatures pour permettre à des groupes de jouer sur scène au début de l'été, sur une période de trois jours et devant un public de plusieurs milliers de personnes. L'association s'engage à mettre tout en œuvre pour garantir à l'ensemble des groupes des conditions d'accueil technique professionnelles (prestations techniques, catering, prise en charge par les équipes de bénévoles). Le festival représente pour certains un tremplin vers des scènes d'envergure nationale.

L'objectif du projet est de proposer un site de candidature fonctionnel, permettant :

- à l'association de suivre l'avancée des candidatures
- aux candidats de procéder à une inscription en ligne, de déposer des maquettes et photos, de saisir les informations les concernant.

Exemple d'interface, vu de l'association :

Liste des candidatures

6	Details	DOLOREST	Pas-de-Calais	62600 BERCK SUR MER	ampli	ROCK Français	2003	DOLOREST C'est romantique, rugueux, poétique,...	Festiv'ovalie Campus (Beau festival à Czela...
7	Details	For The Hackers	Autre	76370 Neuville Les Dieppe	ampli	Pop Rock Electro	2010	MINI BIO "Le jeune groupe normand, pétillant e...	Quelques dates importante Mars 2019 : Sm...
8	Details	SIXTAPE	Somme	80330 LONGUEAU	ampli	Rock	Septembre 2016	Depuis début 2018, avec un nouveau batteur et l'i...	Rockmarin's Saint Valéry s (2018 et 2019)...

Détail d'une candidature



!AYYA!

AUDIO [Bonne-heure.mp3](#) -4:06

AUDIO [AYYA---TRAPPE---02-KAYAK-2.mp3](#) -4:33

AUDIO [AYYA---TRAPPE---04-Jabot-2.mp3](#) -6:12

DEPT Autre

VILLE HEILLECOURT

CP 54180

STYLE Trance-Rock

SND <https://soundcloud.com/ayyalegroupe>

WEB <https://www.facebook.com/AYYAlegroupe/>

VIDEO <https://www.youtube.com/watch?v=e3zaTNUbRgY&t=29s>

EXPERIENCES

Le groupe fera son 100 ème concert au mois de juin prochain en 3 ans d'existence, il tourne à travers la France et pays limitrophes allant des gros festivals aux petits bars. Ils ont partagé la scène entre autre (pour les noms les plus connus) avec No One Is Innocent, Elmer Foot Beat, Guérilla Poubelle...

PRESENTATION

!AYYA! duo Nançoisien est devenu le groupe incontournable de la scène indépendante en Lorraine, c'est d'un côté une guitare, de l'autre une batterie et un synthé. Energie rock, rythmiques alambiquées, ambiance électro et envolées épiques vous plongent dans une frénésie incontrôlable. Ecouter, danser et se sentir simplement vivant.

DOSSIER

SETLIST/SACEM [AYYA-setlist2.pdf](#)

FICHE TECHNIQUE [FT--AYYA--6.pdf](#)

Détail

Attendus du projet

Le site doit permettre :

- aux groupes de créer un compte, soumettre les données et les consulter
 - aux responsables de voir la liste des candidatures (résumé), et le détail de chaque candidature
-

Données

Afin de stocker les candidatures, plusieurs tables sont nécessaires, parmi lesquelles :

Table **département**

Le festival favorise en priorité les groupes des Hauts-de-France. Les candidats sont donc invités à saisir leur département d'origine dans le formulaire, parmi la liste des départements français, ou « autre » s'ils viennent d'un pays limitrophe.

Table **scene**

Le festival divise en 3 scènes les candidatures : Tribute, Acoustique/Folk, Amplifié/Rock

Table **utilisateur**

Deux **types** d'utilisateurs sont gérés par le site

- les responsables du festival, qui ont accès à toute la liste des candidatures
- les candidats.

Table **candidature**

Le formulaire en ligne doit permettre aux candidats de fournir et stocker les informations suivantes :

- Nom du groupe
- Département d'origine (cf. table departement)
- Type de scène (cf. table scene)
- Représentant du groupe (contact principal, lié à l'utilisateur connecté, il peut s'agir du manager)
 - Nom, Prénom, Adresse, Code Postal, email, téléphone
- Style musical (ex : *Rock, Punk, Indie Rock, etc*)
- Année de création
- Présentation du texte (quelques lignes, < 500 caractères)
- Expériences scéniques (quelques lignes, < 500 caractères)
- Site web ou page facebook (url)
- Adresse page soundcloud (url, facultatif)
- Adresse page youtube (url, facultatif)
- Membres du groupe (1 à 8)
 - pour chaque membre : nom, prénom, instruments
- Statut associatif (oui/non)
- Inscrit à la SACEM (oui/non)
- Producteur (oui/non)
- 3 fichiers au format MP3
- un dossier de presse PDF (facultatif)
- 2 photos de groupe avec résolution > 300 dpi
- une fiche technique PDF
- un document SACEM PDF (ou la liste des noms/compositeurs/durée des morceaux de la setlist)

Réalisation du projet

Le projet est à réaliser par groupes de 3 personnes maximum.

Travail et fonctionnalités obligatoires :

1. Analyser les données et créer les tables correspondantes.
2. Créer dans la table **utilisateur** un utilisateur administrateur
3. Etablir un schéma de l'ensemble des routes proposées dans le site, en partant de la route /, en précisant si elles sont de type GET ou POST, et si elles sont accessibles publiquement ou en étant authentifié
4. Créer le formulaire d'inscription puis celui de connexion
5. Créer le formulaire de candidature en veillant, au moins côté serveur :
 - à ce que les champs obligatoires soient saisis
 - à ce que les champs respectent les formats de données (adresse mail, site web, code postal, téléphone, etc), en indiquant à l'utilisateur les messages d'erreur adéquats
6. Créer la liste des candidatures (uniquement accessible aux administrateurs)
7. Créer la vue détaillée d'une candidature (accessible au groupe donné et aux responsables)

Fonctionnalités bonus :

- Liste des utilisateurs (accessible aux responsables)
- Possibilité de modifier sa candidature (accessible aux groupes et aux responsables)
- Possibilité de supprimer une candidature (accessible aux responsables)
- Le site est accessible en ligne.
- API proposant les routes suivantes avec résultat JSON
 - /stats/nombre-candidatures
 - /stats/candidatures-par-departement
 - /stats/candidatures-par-departement/80 (passage d'un argument)
 - /stats/candidatures-hauts-de-france
 - /stats/candidatures-hors-hauts-de-france
 - /stats/candidatures-par-scene
 - /stats/candidatures-par-scene/folk (passage d'un argument)

Evaluation

- Le travail doit être retranscrit dans un rapport au format PDF qui donnera le détail
 - des tables de la base de données,
 - des routes utilisées dans le projet,
 - de la répartition des tâches au sein du groupe.
- Le code source commenté et le document PDF seront déposés sur Moodle, avec les noms des membre du groupe
- La « moulinette » anti-plagiat sera appliquée après dépôt.

Une note globale sera attribuée au bi/tri/nôme, à répartir vous-mêmes entre étudiants (après validation et justification auprès de l'enseignant)

Exemple : Trinôme, note de 13/20 => 39 points à répartir sur les 3 étudiants, de manière égale (13,13,13) ou en tenant compte du travail de chacun (16, 11.5, 11.5)

Contraintes techniques

- PHP 7 : utilisation **obligatoire** d'un framework : Flight recommandé, Symfony/Laravel/Cake... pour ceux qui ont déjà la maîtrise de ces environnements.
- Base de données : MySQL/MariaDB
- Interface PDO

Liste des éléments à fournir

- Rapport PDF
- Code source
- Site fonctionnel, éventuellement en ligne

Critères de notation

Critère	Objectif/ Subjectif	Détail	Prise en compte dans la note
Document PDF	O/S	analyse, clarté des explications, répartition des tâches	30 %
Fonctionnalités			
Tables	O	les tables sont correctement construites (taille des champs, attributs)	5 %
Inscription	O	le formulaire d'inscription permet de créer un compte, les données sont inscrites dans la base, le mot de passe est chiffré	10 %
Connexion	O	le formulaire permet de se connecter au site, et d'avoir accès aux pages spécifiques à un rôle (candidature, fiche du groupe, liste des candidatures...)	10 %
Formulaire Candidature	O	le formulaire permet la saisie de tous les champs spécifiés dans le sujet. les formats et les champs obligatoires sont vérifiés côté serveur les champs sont pré-remplis après essais multiples	20 %
Liste	O	la liste affiche les candidatures (champs pertinents) et un lien vers la fiche du groupe	5 %
Détail d'une candidature	O	la fiche montre le détail de la candidature : photo, tous les champs, les liens vers les fichiers les fichiers ne sont accessibles que si l'utilisateur est authentifié	10 %
Ergonomie	S	les champs de saisie sont choisis de manière à faciliter le remplissage des informations les éventuelles erreurs de saisies apparaissent clairement à l'utilisateur	5 %
Apparence	S	un soin est apporté aux pages (utilisation de CSS)	5 %
Fonctionnalités bonus	O		bonus

Annexe

Pistes pour la mise en œuvre de certaines fonctionnalités

• Transmission de fichiers

Côté navigateur, les formulaires proposent un type de champ spécifique pour l'envoi de fichiers (textes, images, documents, etc.). Le formulaire doit nécessairement utiliser la méthode POST, et préciser un type d'encodage particulier pour permettre de distinguer les données du/des fichiers envoyés et les autres paramètres.

```
<form action="envoi" method="post" enctype="multipart/form-data">
    <input name="photogroupe" type="file">
```

Côté PHP (sans framework), la réception des fichiers est gérée à l'aide du tableau `_FILES`. Le contenu du tableau permet d'avoir accès aux informations concernant le(s) fichier(s) transmi(s). Dans Flight, ces données sont récupérées avec `Flight::request()->files` cf <https://flightphp.com/learn#requests>

extrait de la doc PHP :

```
$_FILES['photogroupe']['name']
```

Le nom original du fichier, tel que sur la machine du client web.

```
$_FILES['photogroupe']['type']
```

Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, cela pourra être "image/gif". Ce type mime n'est cependant pas vérifié du côté de PHP.

```
$_FILES['photogroupe']['size']
```

La taille, en octets, du fichier téléchargé.

```
$_FILES['photogroupe']['tmp_name']
```

Le nom temporaire du fichier qui sera chargé sur la machine serveur.

```
$_FILES['photogroupe']['error']
```

Le code d'erreur associé au téléchargement de fichier.

Si aucune erreur n'est survenue (code 0), on peut déplacer le fichier temporaire (tmp_name) vers son lieu de destination (par exemple : /data/uploads) à l'aide de la fonction `move_uploaded_files`

```
move_uploaded_file($_FILES['photogroupe']['tmp_name'], $nouveauNomFichier)
```

ATTENTION

- L'envoi de fichier est une source importante d'attaque sur le web. De manière générale, il faut faire très attention à la gestion des fichiers (type, contenu, nom) ainsi qu'au répertoire de destination pour éviter qu'un attaquant parvienne à introduire un script.
- On stocke dans la base le nom du fichier, pas le fichier lui-même
- Il est conseillé de renommer les fichiers déposés sur le serveur :

```
function nettoyer_nom_fichier($nom){
    $transliterator = Transliterator::createFromRules(
        ':: NFD; :: [:Nonspacing Mark:] Remove; :: NFC;', Transliterator::FORWARD
    );
    $normalized = $transliterator->transliterate($nom);
    return preg_replace("/^[^a-zA-Z0-9\.\_]/", "-", $normalized);
}
```

• Vérification de la validité des saisies

Une première validation peut être effectuée par le navigateur en choisissant les champs adaptés : url, date, email, number, etc.

cf <https://developer.mozilla.org/fr/docs/Web/HTML/Element/Input>

La vérification côté PHP se fait en testant l'existence de paramètres transmis par POST, récupérés avec `Flight::request()->data`

Exemple :

```
Flight::route('POST /formulaire', function(){
    //tester si un champ test a été transmis
    if(isset(Flight::request()->data->test))
    {...}

    //tester si le champ annee est vide
    if(empty(Flight::request()->data->annee))
    {...}

    //tester si le champ titre a une longueur < 20
    if(strlen(Flight::request()->data->titre)<20)
    {...}
}
```

PHP propose quelques filtres de validation sur lesquels vous pouvez vous appuyer
cf. <https://www.php.net/manual/fr/filter.filters.validate.php>



ATTENTION

La vérification côté client ne dispense pas d'une vérification côté serveur, **obligatoire**.

En effet, les données transmises au script peuvent provenir d'un script, d'une application, etc., dans lesquels aucun contrôle n'est effectué.

Cette vérification assure la cohérence des données dans la base, notamment.

• Pré-remplissage des données de formulaire et messages d'erreurs

Lors de la réception des données du formulaire, si après vérification il reste des erreurs, il faut donc revenir à nouveau sur le formulaire, pré-rempli avec les valeurs déjà saisies, afin d'éviter que l'utilisateur n'ait à recommencer toute la saisie (et finir par abandonner).

Côté serveur, il suffit de demander le rendu du template et de passer les valeurs déjà saisies :

```
Flight::view()->assign("erreurs",true); //passe une variable erreurs au template  
Flight::render("form.tpl",$_POST); //transmets les valeurs de $_POST
```

Côté template, on peut utiliser les valeurs transmises par PHP dans l'attribut **value**

```
<input type='text' name='nom' placeholder='Nom' value='{ $nom|escape|default:'' }'>
```

Explication **{ \$nom|escape|default:'' }**

Utiliser la variable \$nom si elle existe, échapper les caractères (apostrophes, etc), sinon utiliser la chaîne vide.

• Apparence et ergonomie

Pour créer des formulaires efficaces et réduire les erreurs de saisie, il est préférable de guider les saisies (date, département, etc.) à travers des champs adaptés, comme des listes déroulantes, des cases à cocher, des boutons radios, des champs de date, etc.

- Les listes déroulantes, voire les checkbox et radios sont générées à partir des données de la base.
- Un framework CSS peut être utilisé pour enjoliver le formulaire : PureCSS , Bootstrap , ...