

## **Adoption d'animaux de compagnie**

Vous devez concevoir une application web servant à offrir un animal de compagnie en adoption. L'adoption et la mise en adoption sont gratuites pour tous les utilisateurs.

### **Consultation**

Tous les utilisateurs peuvent consulter la liste des animaux offerts pour une adoption. Chaque animal possède sa propre page affichant ses données et son adresse. La page d'accueil doit afficher 5 animaux disponibles au hasard ainsi que des liens vers leur page respective. Un moteur de recherche simple doit également permettre une recherche parmi les animaux disponibles.

Les résultats de la recherche doivent être affichés sur une nouvelle page. Les résultats doivent être affichés avec leur description et leur photo, ainsi qu'un lien vers la page de l'animal. Une pagination de 5 résultats par page doit être appliquée sur les résultats de la recherche.

Sur la page d'un animal, un formulaire doit permettre de contacter le propriétaire de l'animal par courriel pour lui signifier notre intérêt pour l'adoption. La soumission du courriel doit être fait avec une requête Ajax. Un message de succès ou d'erreur doit indiquer sur le message a bien été envoyé ou si une erreur s'est produite. L'utilisateur doit fournir son adresse courriel pour le retour et le corps de son message. Le sujet du courriel doit être généré par l'application.

### **Mise en adoption**

Sur la page d'accueil, il doit être possible de s'authentifier ou de cliquer sur un lien qui mène à un formulaire de création d'un compte. Le compte doit contenir au minimum l'information suivante : le nom et prénom de la personne, un numéro de téléphone, une adresse et un courriel. Après l'authentification ou la création du compte, l'utilisateur est dirigé vers une page où il peut modifier ses informations personnelles (sauf le courriel).

L'utilisateur peut alors mettre un animal en adoption en remplissant un autre formulaire contenant : le nom de l'animal, le type d'animal, sa race, son âge, une description textuelle et la possibilité de téléverser une photo de l'animal.

La page de coordonnées et celle pour mettre un animal en adoption ne doivent être accessibles qu'à un utilisateur authentifié sur le site. Une page d'authentification doit être affichée à l'utilisateur qui tente d'accéder une de ces pages sans être authentifié. Les autres routes où une authentification est requise doivent retourner un code de statut 401 si on tente d'y accéder sans être authentifié.

Le mot de passe doit être stocké selon la technique du *salt* présentée en classe.

Vous devez mettre en place un mécanisme de réinitialisation de mot de passe en cas d'oubli, comme discuté en classe.

### **API**

Vous devez fournir une interface programmable permettant d'obtenir la liste de tous les animaux actuellement disponibles pour une adoption en format JSON. Vous devez fournir la totalité des données connues sur l'animal (incluant l'adresse du propriétaire) à l'exception de la photo.

## ***La base de données***

Vous devrez créer la base de données pour mettre en place ces fonctionnalités. Vous devez fournir un script SQL contenant les instructions nécessaires à la construction de votre base de données. Un fichier readme, situé à la racine du projet, doit indiquer où se situe le script SQL, la procédure pour créer la base de données et où elle doit être située.

## ***Courriel***

Il est nécessaire d'utiliser une adresse Gmail pour l'envoi des courriels. L'adresse courriel et le mot de passe associé doivent être situés dans un fichier de configuration dans votre projet (le format est à votre discrétion). Indiquez, dans le fichier readme, où se trouve le fichier de configuration. Le correcteur devrait pouvoir utiliser sa propre adresse Gmail lors de la correction en ne modifiant que ce fichier de configuration. Ne pas mettre ces données directement dans le code source.

## ***Messages***

Aucun message ne doit être affiché à l'utilisateur à l'aide des fonctions de popups de Javascript (ex. alert, prompt, etc.). Utilisez une autre technique impliquant du Javascript et la manipulation du DOM.

## ***Technologies***

Dans le front-end, vous **devez** utiliser les technologies suivantes :

- HTML 5
- CSS 3
- Javascript

Toujours dans le front-end, vous **pouvez** utiliser toutes les bibliothèques et tous les frameworks Javascript que vous souhaitez, tant que ça ne nécessite aucune installation lors de la correction (vous fournissez les sources dans votre projet).

Dans le back-end, vous ne devez utiliser que les technologies vues en classe, ce qui inclut :

- Python 3
- Flask 0.12
- SQLite 3
- Jinja2

Le code Python doit respecter PEP8. Tous les fichiers sources doivent être encodés en UTF8.

Il est nécessaire d'utiliser le patron POST-REDIRECT-GET sauf si spécifié autrement.

## ***Remise***

Le travail peut être fait seul ou en équipe de 2 personnes. Le répertoire de travail contenant les fichiers doit être archivé dans un fichier zip et nommé selon les codes permanents des auteurs. L'archive doit être remise par Moodle avant le 27 avril 2018 à 23h55. Aucun retard ne sera accepté et une pénalité sera appliquée pour une archive non conforme sans les codes permanents.

Si vous voulez travailler en équipe, vous devez mettre votre logiciel sous gestion de sources dans un dépôt privé git (ex. github, bitbucket) et m'en donner un accès en lecture. Cette mesure sert à

m'assurer que les deux coéquipiers ont partagé l'effort équitablement. Un coéquipier n'ayant pas offert une participation significative n'aura pas les points pour le travail.

### ***Pondération***

Fonctionnalités : 55%

Respect des exigences et standard du web : 35%

Respect de PEP8 et qualité du code (taille des fonctions, qualité de la nomenclature, etc.) : 10%