



PRÉSENTATION DU DOSSIER PROJET ZOO ARCADIA

Thomas Berbigier
STUDI 2023/2024

QUI SUIS-JE ?



Parcours

- Entrepreneur
 - 2015 – 2020
- Employé commercial
 - 2020 – Aujourd'hui

Pourquoi ?

- Intérêt depuis tout jeune pour les nouvelles technologies
- Métier d'avenir

SOMMAIRE



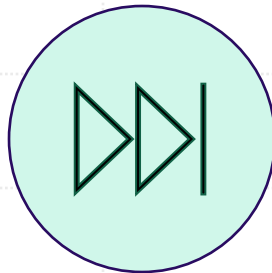
I. Présentation
du projet



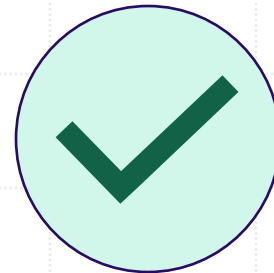
II. Conception



III. Développement



IV. Démonstration
du projet



V. Conclusion

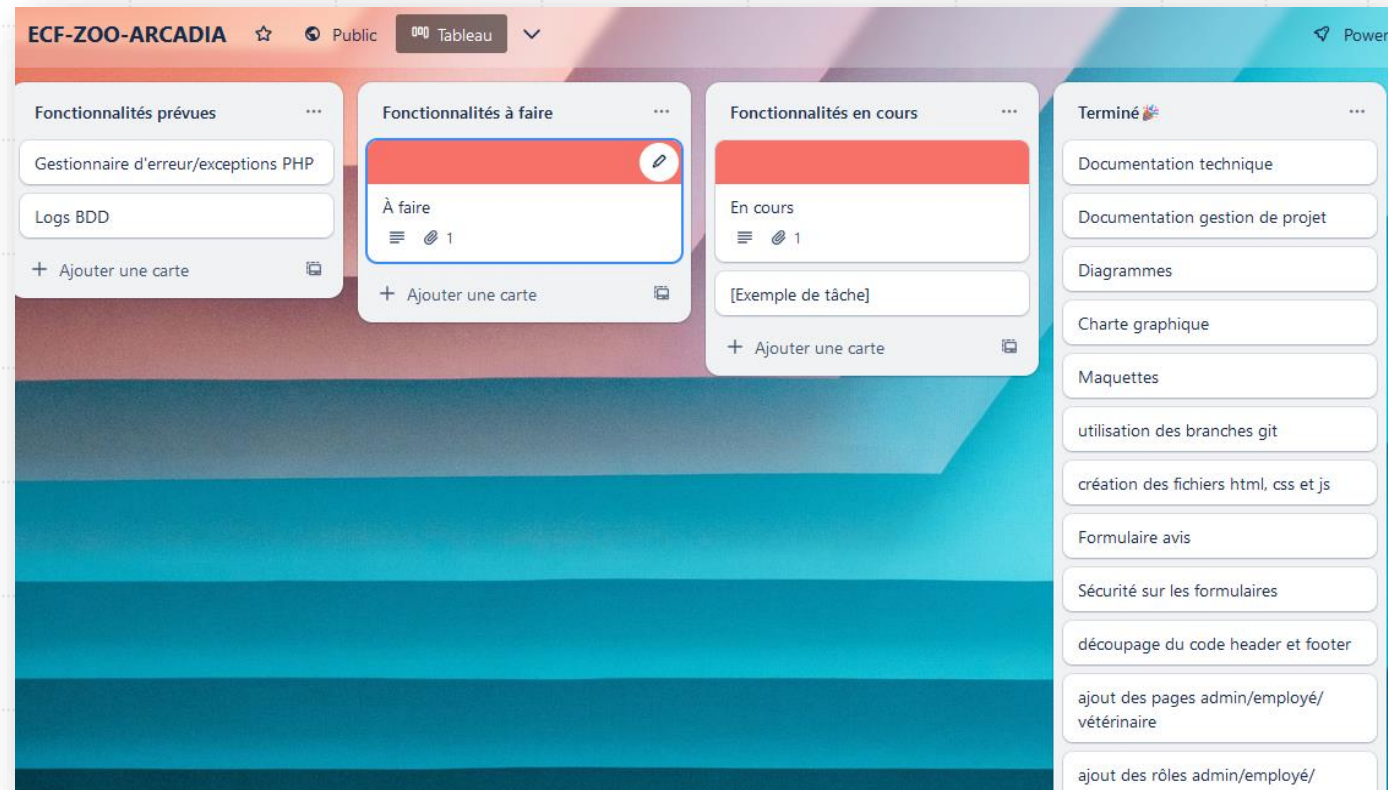
ORGANISATION DU TRAVAIL



7 semaines de travail prévues



Gestion de projet via Trello



ZOO ARCADIA – CAHIER DES CHARGES

CONSTAT

Le zoo Arcadia ne possède pas de site web pour ses potentiels clients ni d'espace fonctionnel pour son personnel.

AUDIENCE

- Visiteurs
- Administrateur
- Employés
- Vétérinaires

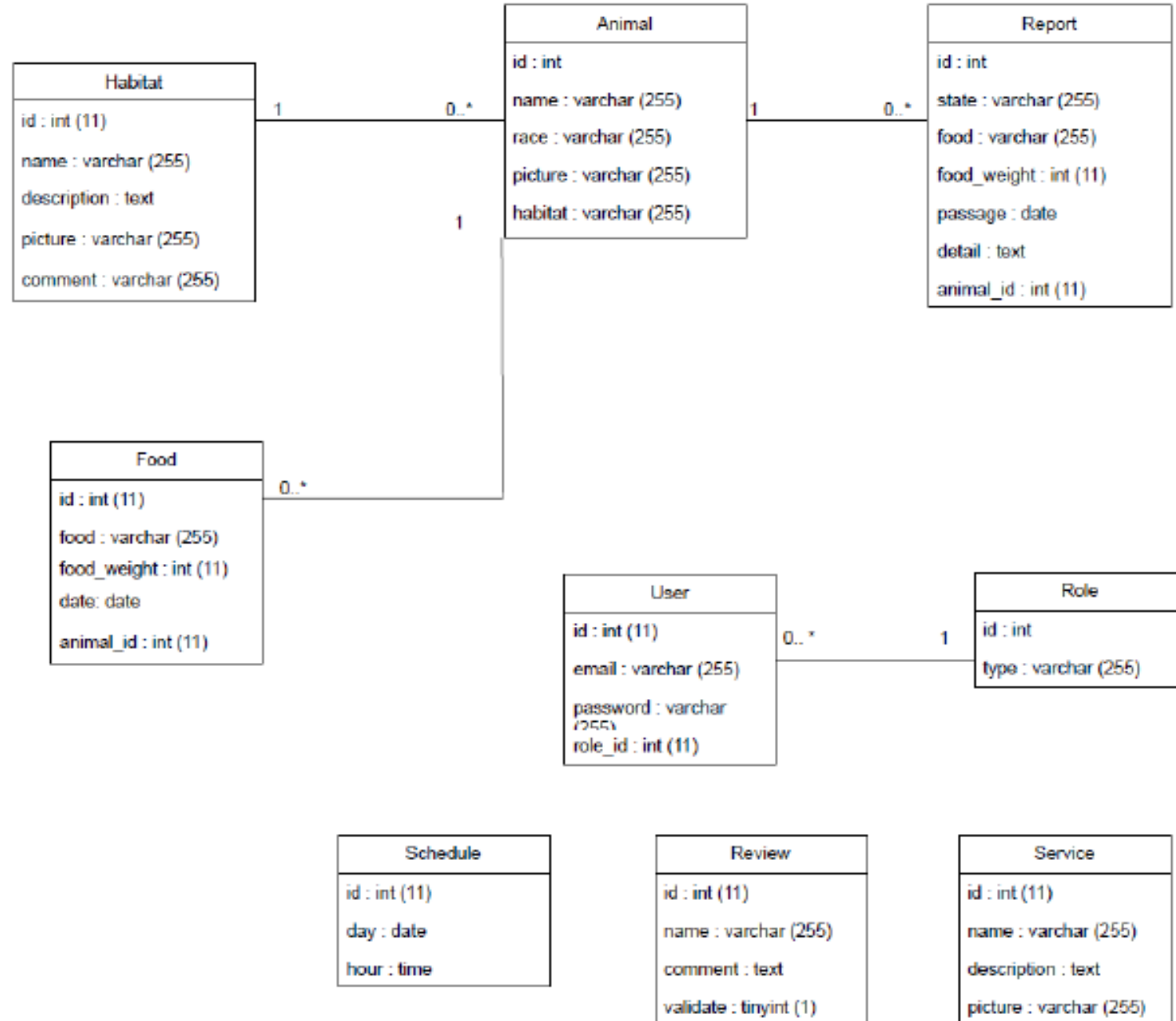


CONCEPTION

USER STORIES & DIAGRAMME DE CAS D'UTILISATION

Acteur	Souhaite	Nécessite
Visiteur	Accéder à la page d'accueil de l'application	
	Rédiger un avis	L'approbation de l'employé pour être publiée
	Accéder aux vues globales de l'application	
	Contacter le zoo via le formulaire	Une réponse de l'employé
Employé	Valider ou invalider les avis	Une connexion à son compte
	Nourrir un animal	
	CRUD Services	
Vétérinaire	Remplir les comptes rendus	Une connexion à son compte
	Mettre une commentaire sur un habitat	
	Accéder à l'historique du nourrissage	
Administrateur	Créer un compte employé et vétérinaire	Une connexion à son compte
	Accès aux comptes rendus vétérinaires	
	Dashboard administrateur	
	CRUD horaires / habitats / animaux	

DIAGRAMME DE CLASSE



WIREFRAMES & MAQUETTES



STACK TECHNIQUE

Front-end	Back-end	Environnement	Contrôle de version
HTML5	PHP	Visual Studio Code	Git & GitHub
CSS3 avec Bootstrap 5	Composer (PHP Mailer, MongoDB)	PHP Intelephense, PHP Debug, Indent-rainbow, Prettier	
JavaScript	MariaDB, MongoDB		



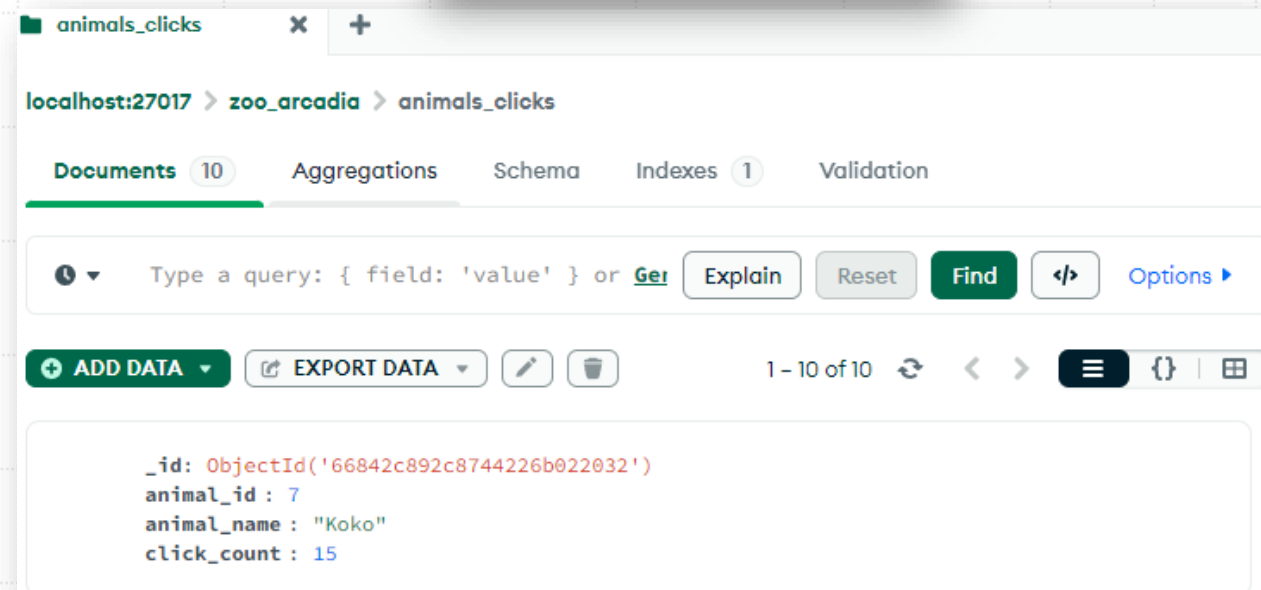
DÉVELOPPEMENT

CRÉATION DES BASES DE DONNÉES

- Création de la base de données locale "zoo_arcadia"
- Configuration d'un compte utilisateur avec un mot de passe conforme aux normes **CNIL**.
- Création des tables et insertion des données via des **requêtes SQL**

- Configuration de la base "zoo_arcadia" via **MongoDB Compass**
- Création d'une **collection** "animals_clicks"

```
CREATE TABLE roles (  
  id int(11) NOT NULL AUTO_INCREMENT,  
  type varchar(255) NOT NULL,  
  PRIMARY KEY (id)  
);  
  
INSERT INTO roles (id, type) VALUES  
(1, 'administrateur'),  
(2, 'employe'),  
(3, 'veterinaire');  
  
CREATE TABLE users (  
  id int(11) NOT NULL AUTO_INCREMENT,  
  email varchar(255) NOT NULL,  
  password varchar(255) NOT NULL,  
  role_id int(11) NOT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (role_id)  
    REFERENCES roles (id)  
);
```



ACCÈS AUX DONNÉES

SQL

- Connexion sécurisée à MariaDB avec **PDO**
- Opérations **CRUD** pour gérer les données
- **Filtrage** avant stockage
- **Filtrage** pour la récupération

```
$pdo = new PDO("mysql:host=$hostname;dbname=$database", $username, $password);
```

```
// Récupère tous les animaux  
$sql = 'SELECT * FROM animals';  
try {  
    $stmtAnimaux = $pdo->query($sql);  
} catch (Exception $e) {  
    $_SESSION['error'] = " Erreur ! " . $e->getMessage();  
}  
$animals = $stmtAnimaux->fetchAll();
```

```
$sql = 'UPDATE animals SET name = ?, race = ?, picture = ?, habitat = ? WHERE id = ?';  
try {  
    $stmt = $pdo->prepare($sql);  
    $stmt->execute([$name, $race, $file_bdd, $habitat, $id]);  
    $_SESSION['message'] = "Animal mis à jour avec succès.";  
} catch (Exception $e) {  
    $_SESSION['error'] = "Erreur lors de la mise à jour de l'animal'.". $e->getMessage();  
}
```

ACCÈS AUX DONNÉES

NoSQL

- Connexion à la base NoSQL MongoDB.
- Opérations CRUD pour gérer les **documents** dans la collection.

```
$client = new Client($connect);  
$collection = $client->zoo_arcadia->animals_clicks;  
  
// Vérification si l'animal doit être enregistré  
$filter = ['animal_id' => $animal_id];  
$existingClick = $collection->findOne($filter);
```

```
if (!$existingClick) {  
  
    // Animal non enregistré, enregistrement dans MongoDB  
    $result = $collection->insertOne([  
        'animal_id' => $animal_id,  
        'animal_name' => $animal_name,  
        'click_count' => 1,  
    ]);  
}
```

```
// Animal déjà enregistré, mise à jour du compteur de clics  
$result = $collection->updateOne(  
    $filter,  
    ['$inc' => ['click_count' => 1]]  
);
```


RÉALISATION DU FRONT-END

Partie statique

- Balises **sémantiques** pour structurer et améliorer le référencement
- **Conformité et accessibilité**
- **Templates** pour header et footer

```
templates > header.php > html > body
1  <?php
2      require_once __DIR__ . '/../lib/session.php';
3      // Protéger contre les attaques ClickJacking
4      header('X-Frame-Options: DENY');
5  ?>
6
7  <!DOCTYPE html>
8  <html lang="fr">
9  <head>
10     <meta charset="UTF-8">
11     <meta name="viewport" content="width=device-width,
12         initial-scale=1.0">
13     <title>Parc animalier au coeur de la Bretagne - Zoo
14         Arcadia</title>
15     <meta name="description" content="Découvrez Zoo
16         Arcadia, un parc animalier unique situé au cœur de
17         la Bretagne. Plongez dans des habitats immersifs,
18         rencontrez des espèces fascinantes et profitez de
19         services exceptionnels pour une journée mémorable.">
20     <meta name="keywords" content="zoo, parc animalier,
21         Bretagne, animaux, habitats, Arcadia, journée
22         familiale, espèces rares, zoo Bretagne">
23     <meta name="author" content="Zoo Arcadia">
24     <meta name="robots" content="index, follow">
25     <link rel="stylesheet" type="texte/css"
26         media="screen" href="css/reset.css">
27     <link href="css/bootstrap.min.css" rel="stylesheet">
28     <link href="css/style.css" rel="stylesheet">
29 </head>
30 <body>
31     <!-- Image de fond -->
32     <div class="background">
33         
36     </div>
```

RÉALISATION DU FRONT-END

Style

- Reset CSS pour **uniformiser**
- Feuille de style pour l'apparence et la mise en page de l'application

```
5  /* image de fond */
6  .background {
7      position: fixed;
8      top: 0;
9      left: 0;
10     width: 100%;
11     height: 100%;
12     z-index: -1; /* Mettre en arrière-plan */
13 }
14 /* Style pour l'image de fond "blobs" */
15 .blob-image {
16     top: 0;
17     left: 0;
18     width: 100%;
19     height: 100%;
20     opacity: 1;
21     filter: brightness(0.5)
22 }
23
24 /* header */
25 header {
26     position: relative;
27     width: 100%;
28     height: auto;
29 }
30
31 /* navbar */
32 .navbar-custom {
33     background-color: rgb(18, 55, 42, 0.5);
34     position: absolute;
35     top: 0;
36     left: 0;
37     width: 100%;
38     z-index: 1000;
39 }
```


RÉALISATION DU FRONT-END

Responsive design

- Grille Bootstrap / Media Queries CSS

```
48 <!-- Début cards habitats/animaux -->
49 <div class="container accueil">
50   <div class="container-card text-center mt-5">
51     <div class="row">
52       <div class="col-12 col-md-6 col-lg-4 mb-3">
53         <div class="card h-100 card-savane">
54           
55           <div class="card-body">
56             <h3 class="card-title text-center">Savane</h3>
57             <p class="card-text">Bienvenue dans l'habitat
savane de notre zoo, une vaste étendue qui capture
l'essence des plaines africaines. Ici, les lions
règnent en maîtres, les élégantes girafes se
promènent gracieusement et les majestueux éléphants
complètent ce tableau.</p>
58             <a href="habitats.php#heading1" class="btn btn-dark
fs-5">Je découvre la savane !</a>
59           </div>
60         </div>
61       </div>
</div>
```

```
354 /* media queries mobile */
355 @media only screen and (max-width: 600px) {
356   .navbar-collapse {
357     background-color: #1a3d4d;
358   }
359   header {
360     height: 97px;
361   }
362
363   /* image header */
364   .img-accueil{
365     width: 100%;
366     height: 100%;
367     object-fit: cover;
368   }
369   .overlay .title-index {
370     margin-bottom: 300px;
371     font-size: 2rem;
372     font-weight: bold;
373   }
374
375   /* Article */
376   .photo img {
377     display: none;
378   }
379   .description {
380     width: 100%;
381     text-align: justify;
382   }
383   .container-presentation {
384     display: block;
385     width: 100%;
386   }
387   footer {
388     text-align: center
389   }
```

RÉALISATION DU FRONT-END

Partie dynamique

- **Animation** en zone visible de l'écran.
- **Intégration** de composants Bootstrap.
- **Enregistrement** des clics avec l'API fetch

```
js > JS script.js > window.addEventListener("scroll") callback
1  // animation de présentation //
2  window.addEventListener("scroll", () => {
3      const photo = document.querySelector(".photo");
4      const description = document.querySelector(".description");
5      const windowHeight = window.innerHeight;
6      const photoTop = photo.getBoundingClientRect().top;
7      const descriptionTop = description.getBoundingClientRect().top;
8      if (photoTop < windowHeight - 100) {
9          photo.classList.add("show");
10     }
11
12     if (descriptionTop < windowHeight - 100) {
13         description.classList.add("show");
14     }
15 });
```

RÉALISATION DU BACK-END

Rôles

- Chaque utilisateur a un rôle **fixe**, défini à la création.
- Les **sessions**, tokens **CSRF** et mots de passe **hashés** protègent les données.
- Rôle récupéré à la connexion.

```
$role = $_POST['selectRole'];  
// Vérifier que le rôle sélectionné est valide (employé ou  
vétérinaire)  
if (($role == 2 || $role == 3) && (!empty($email)) && (!empty  
($hashedPassword))) {  
    // Insérer le nouvel utilisateur  
    $sql = 'INSERT INTO users (email, password, role_id) VALUES  
    (?, ?, ?)';  
    try {  
        $stmt = $pdo->prepare($sql);  
        $stmt->execute([$email, $hashedPassword, $role]);  
        $_SESSION['message'] = "Compte utilisateur créé avec  
succès. Mail de confirmation envoyé.";  
        // Envoyer mail de confirmation  
        sendEmail($email);  
    } catch (Exception $e) {  
        $_SESSION['error'] = "Erreur lors de la création du compte.  
        " . $e->getMessage();  
    }  
} else {  
    $_SESSION['error'] = "Tous les champs doivent être remplis."  
}  
header('Location: administrateur.php');  
exit();
```

```
<div class="mt-2">  
    <label for="selectRole" class="form-label">Rôle :</label>  
    <select class="form-select" id="selectRole" name="selectRole" aria-label="Default select example" required>  
        <option value="2">Employé</option>  
        <option value="3">Vétérinaire</option>  
    </select>
```

RÉALISATION DU BACK-END

Dashboard administrateur

- Chaque clic sur un bouton d'accordéon envoie l'ID et le nom de l'animal au serveur via une **requête** POST en JSON.
- Vérifie et **enregistre** les données reçues.
- **Affiche** une liste triée dans un tableau.

```
// Enregistrer clics animaux
document.querySelectorAll('.accordion-button').forEach(button => {
  button.addEventListener('click', () => {
    const animalId = button.getAttribute('data-animal-id');
    const animalName = button.textContent.trim();

    // Envoi une requête HTTP à 'register_click.php'
    fetch('register_click.php', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      // Corps de la requête, conversion en chaîne JSON
      body: JSON.stringify({ animal_id: animalId, animal_name: animalName })
    })
    .then(response => response.json())
  });
});
```

SÉCURITÉ

Aspect	Mesures mises en œuvre
Protection contre les attaques par force brute	<ul style="list-style-type: none">- Hashage des mots de passe avec <i>bcrypt</i>.- Création de mots de passe selon les normes CNIL.
Protection contre les attaques XSS	<ul style="list-style-type: none">- Utilisation de <i>htmlspecialchars()</i> pour échapper les données avant affichage.- Filtrage et validation des entrées via <i>filter_input()</i> côté serveur.
Protection contre les attaques CSRF	<ul style="list-style-type: none">- Génération et vérification des tokens CSRF sur tous les formulaires.
Protection contre les injections SQL	<ul style="list-style-type: none">- Requêtes préparées pour éviter les injections SQL.
Sécurisation des formulaires	<ul style="list-style-type: none">- Ajout d'attributs <i>required</i> ou <i>pattern</i> pour garantir les entrées avant soumission.- Utilisation des balises <code><input type="email"></code> et <code><input type="password"></code> pour des validations spécifiques.
Téléchargements de fichiers	<ul style="list-style-type: none">- Validation des extensions et taille des fichiers uploadés.
Sécurité des sessions	<ul style="list-style-type: none">- Configuration des cookies de session avec les attributs <i>secure</i> et <i>httponly</i>.

DÉPLOIEMENT EN LIGNE

- Déploiement via Heroku, Git pour **pousser** les fichiers du projet.
- **Configuration** des add-ons JawsDB Maria et ObjectRocket for MongoDB.
- **Dump** SQL depuis la base locale vers Heroku.
- Variables de connexion dynamiques.



DÉMONSTRATION DU PROJET



CONCLUSION

- Application web répondant aux besoins variés des utilisateurs.
- Fonctionnalités clés mises en place.
- Montée en compétences.
- Perspectives pour la suite.



ZOO ARCADIA