

Rene

LELIARD

Date

Projet Arcadia



Présentation pour TP DWWM



Agenda



Introduction au
projet



Planification et
spécifications
techniques



Conception et
développement



Opérations et
qualité



Clôture

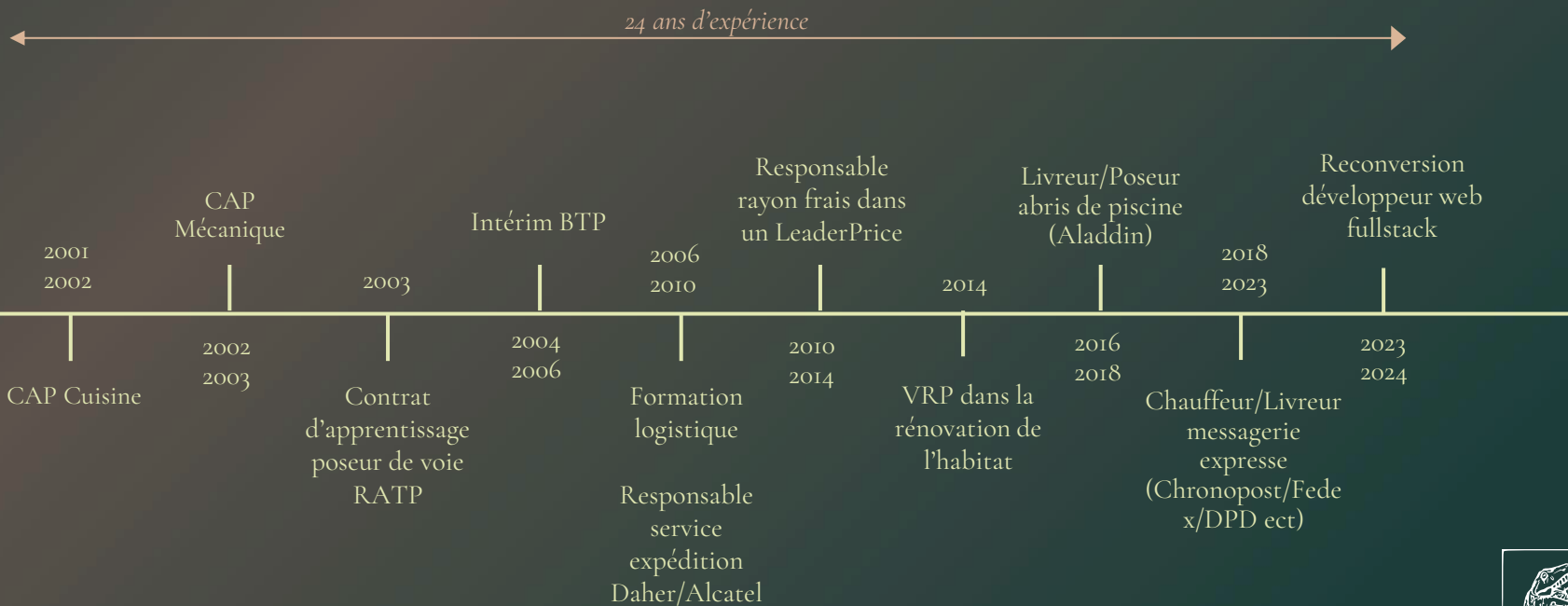


Introduction au projet



- ❖ Présentation personnelle
- ❖ Objectifs et vision du projet
- ❖ Importance et impact attendu.

Mon parcours professionnel



Rappel des consignes du projet

Evaluation en Cours de
Formation



Sujet & Cahier des charges
imposés



Projet en autonomie



Zoo Arcadia

Une application web destinée aux employés du zoo Arcadia et aux visiteurs du Zoo



A destination

- José, directeur du zoo
- Employées
- Vétérinaires
- Visiteur



Objectifs vue utilisateur

- Communication publique
- Opérations métiers quotidiennes

Planification et spécifications techniques



Cahier des charges :

- Détail des exigences fonctionnelles et non fonctionnelles

Stack technique :

- Technologies utilisées et justification de leur choix

Gestion de projet :

- Méthodologies de gestion utilisées (Agile, Scrum, etc.)
- Planification des sprints et des livraisons

Cahier des charges

L'application permettra une gestion efficace des comptes, des accès, des opérations quotidiennes, des avis clients, et des informations détaillées sur les services et les animaux.

L'application devra permettre :



Une gestion centralisée et sécurisée des comptes utilisateurs.



Le contrôle des accès privilégiés pour les différentes catégories de personnel.



L'exécution des opérations quotidiennes par les employés depuis un espace dédié.



La collecte et l'affichage des avis des clients et des demandes de contact.



L'affichage des informations relatives aux services, habitats, avis clients, animaux et leurs bilans vétérinaires.

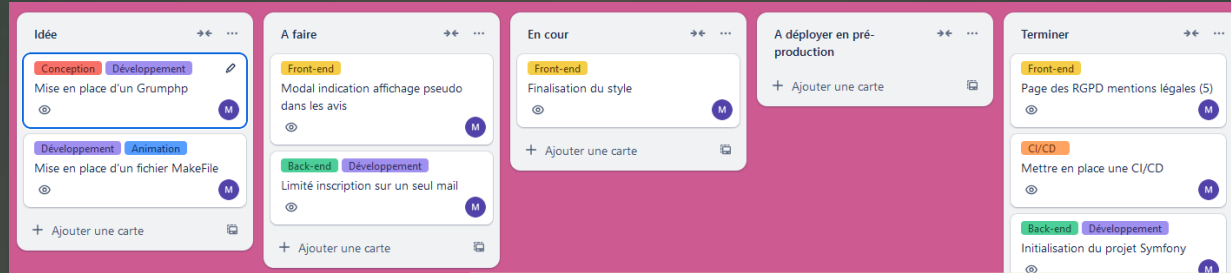
Stack technique

Choisir la bonne stack technique est crucial car elle influence la performance, la scalabilité, la maintenabilité et la sécurité de l'application ou du projet. Chaque élément de la stack doit bien s'intégrer avec les autres pour créer un système efficace et robuste

stack technique	nom	Complexité
Langages de programmation	HTML5 ,CSS3, JavaScript(ES6), PHP8.3.	☆☆☆
Frameworks et bibliothèques	Symfony	☆
Base de données	(SQL) comme MySQL, ou non-relationnels (NoSQL) comme MongoDB.	☆☆
Serveur web et serveur d'application	Nginx et Apache	
Infrastructure et plateformes d'exécution	cloud computing comme AWS, Google Cloud, ou Azure et plateformes comme Docker	
Outils DevOps	Jenkins, GitHub, Ansible	



Gestion de projet



Pourquoi avoir choisi TRELLO :

- Organisation facile: Trello facilite l'organisation de projets avec un système visuel intuitif.
- Haute adaptabilité: Il offre une personnalisation extensive pour s'adapter à divers besoins.
- Utilisation simple: Accessible et facile à utiliser, il permet un démarrage rapide

Conception et Développement



Outils de développement :

- Gestion de versions

Design et expérience utilisateur :

- Charte graphique, wireframes et maquettes

Architecture de données :

- MCD (Model conceptuel des données)
- Conception de la base de données
- Création des entités et relations

Développement Front-End :

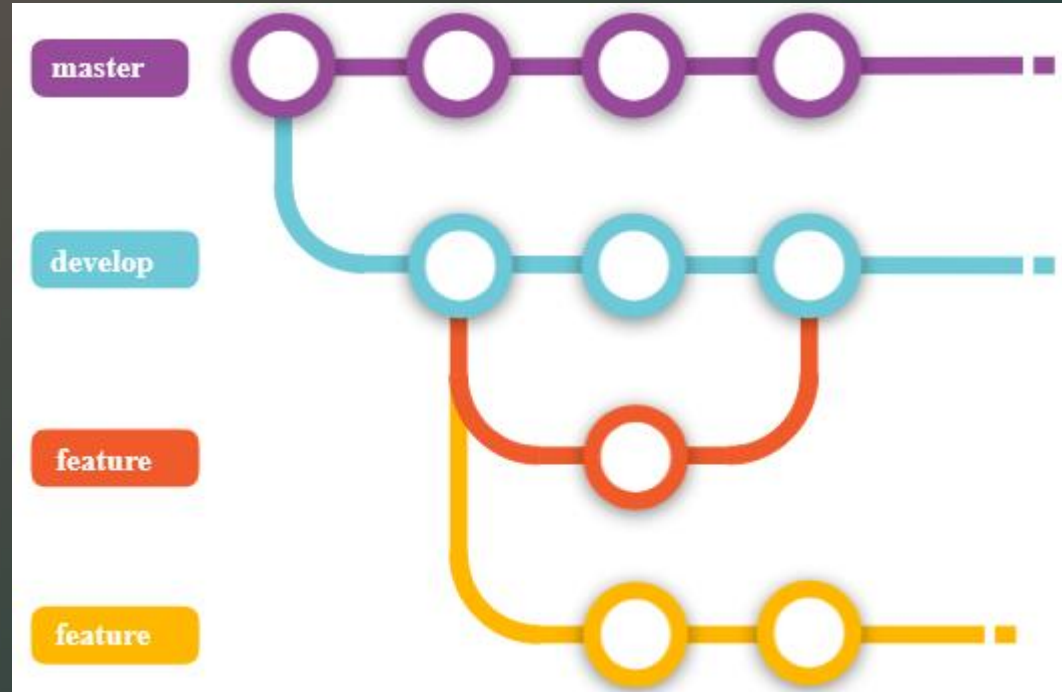
- Dynamisation de l'interface : technologies et approches utilisées

Gestion des versions

Branche Main ->
Branche de production

Branche Dev ->
Branche de
développement

Branche XXX -> Une
branche par
fonctionnalités



Conception Graphique

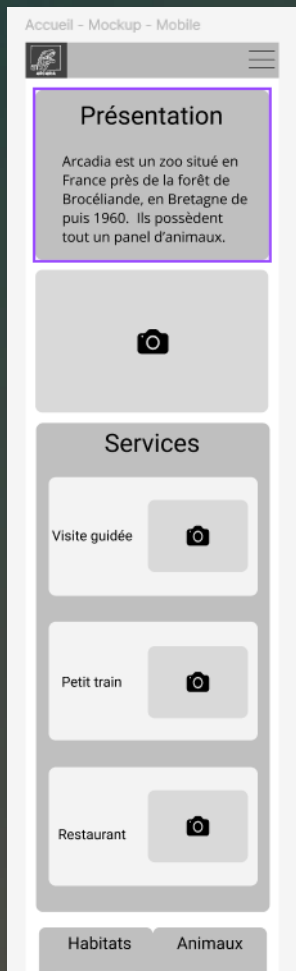
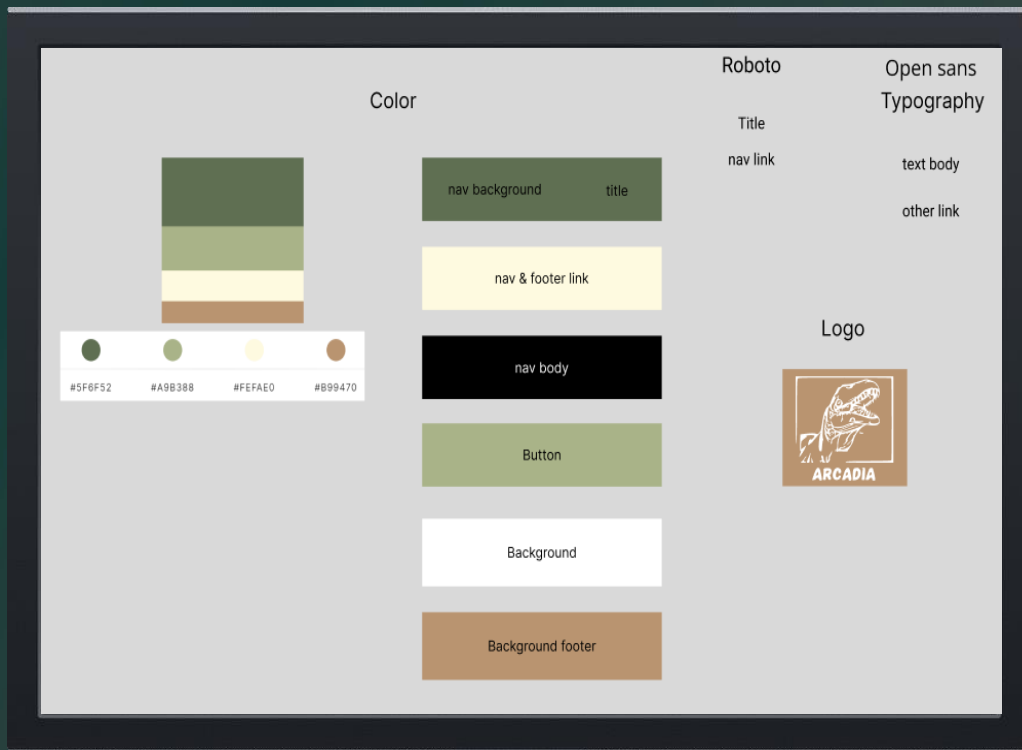
Charte graphique : choix des couleurs et typographies adaptées au thème animalier

Wireframes : conception des pages pour une navigation intuitive

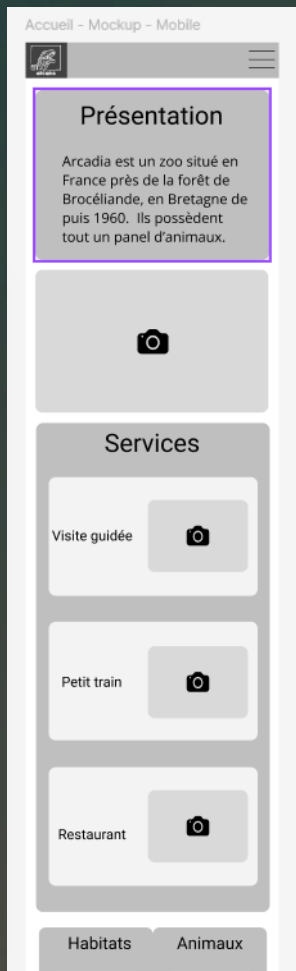
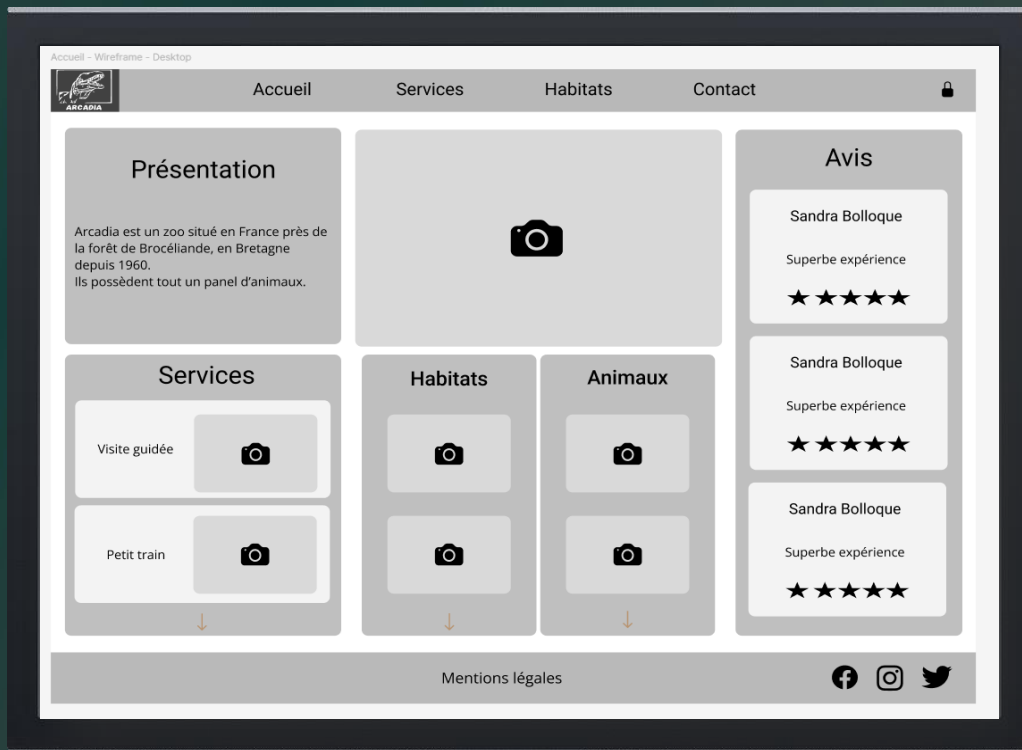
Maquettes : modèles visuels pour valider l'interface utilisateur



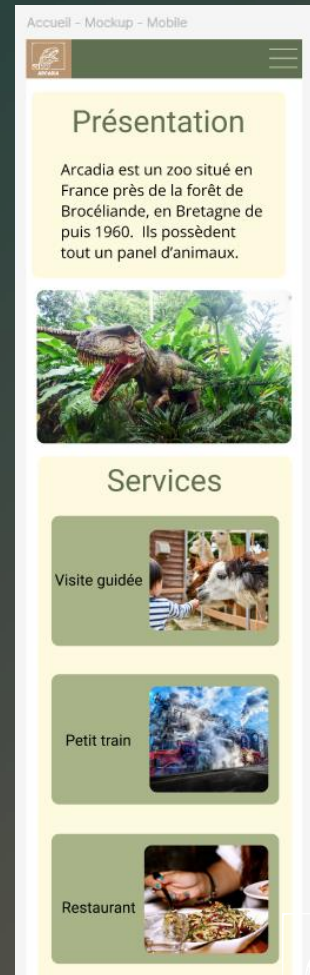
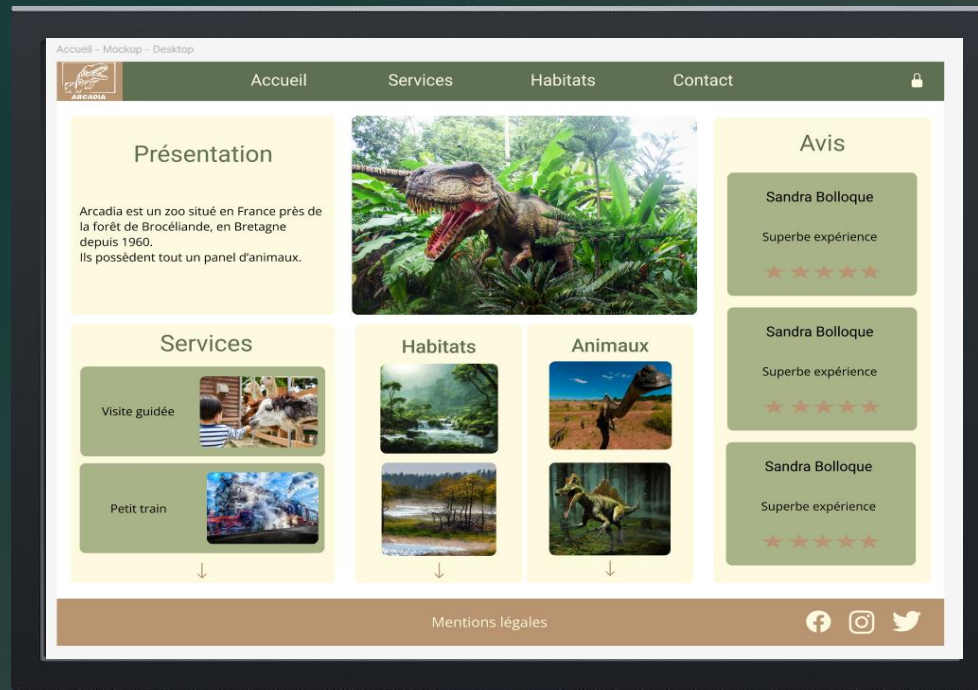
Charte graphique



Wireframe



Mokup (maquettes)



Modèle conceptuel de données



Représentation
graphique des
entités et
associations.



Identifier &
structurer
données



Comprendre
relations



Entités



Attributs

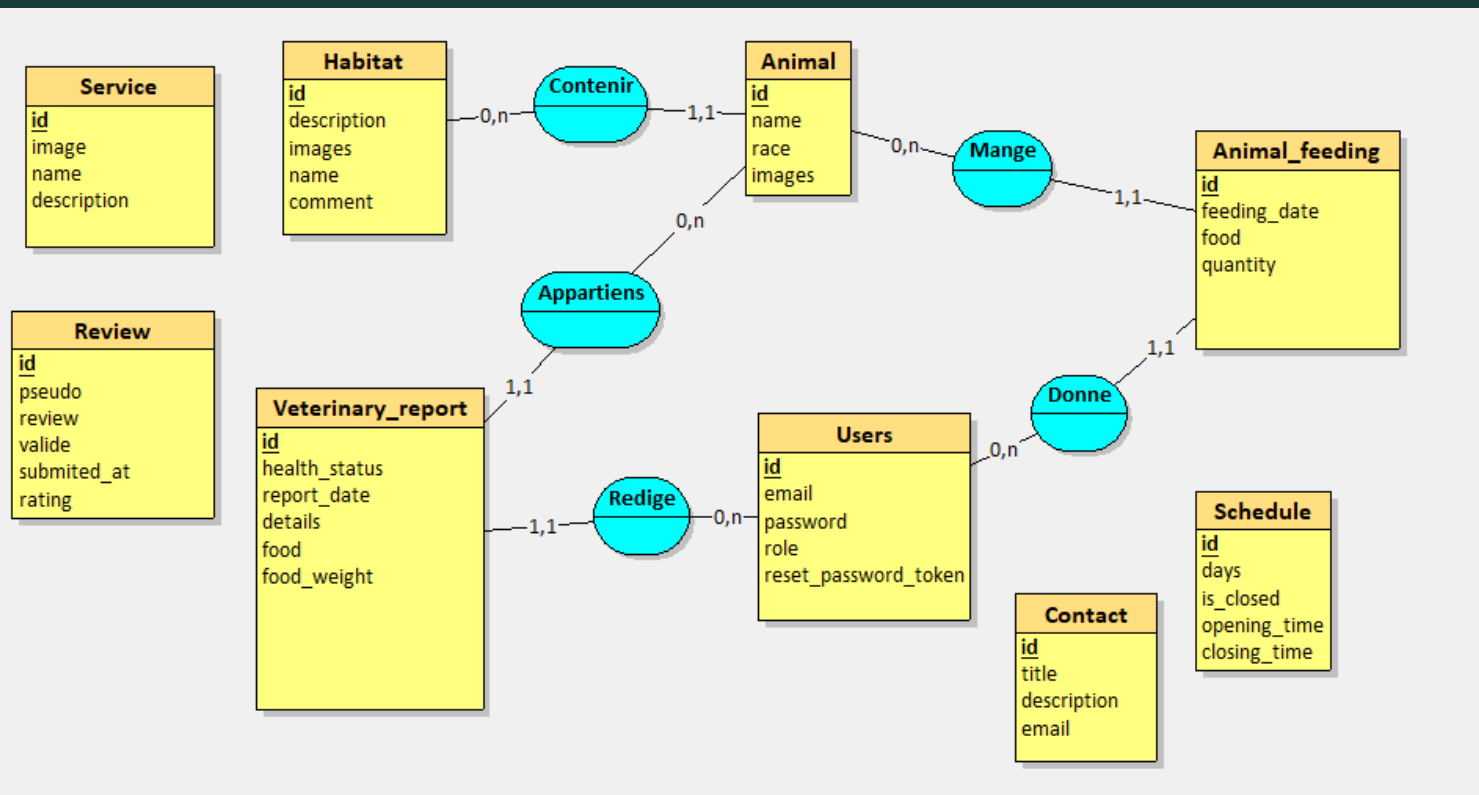


Association:
Relations entre
les entités



Cardinalités: Min
& Max liaisons
entre entités

MCD



MCD

```
docs > create_database.sql
```

```
1 CREATE DATABASE IF NOT EXISTS arcadia_db CHARACTER SET utf8 COLLATE utf8_general_ci;
2 USE arcadia_db;
3
```

```
2
3 CREATE TABLE IF NOT EXISTS habitat (
4     id INT AUTO_INCREMENT PRIMARY KEY,
5     name VARCHAR(100) NOT NULL,
6     description TEXT NOT NULL ,
7     image JSON NOT NULL ,
8     comment TEXT
9 );
10
11 CREATE TABLE IF NOT EXISTS animal (
12     id INT AUTO_INCREMENT PRIMARY KEY,
13     name VARCHAR(150) NOT NULL,
14     race VARCHAR(150) NOT NULL,
15     image JSON,
16     habitat_id INT,
17     FOREIGN KEY (habitat_id) REFERENCES habitat(id)
18 );
19
20 > CREATE TABLE IF NOT EXISTS contact ( ...
25 );
26
27 > CREATE TABLE IF NOT EXISTS service ( ...
32 );
33
34 CREATE TABLE IF NOT EXISTS review (
35     id INT AUTO_INCREMENT PRIMARY KEY,
36     pseudo VARCHAR(30) NOT NULL,
37     comment TEXT NOT NULL,
38     valid BOOLEAN DEFAULT FALSE,
39     submitted_at DATETIME NOT NULL,
40     rating INT NOT NULL
41 );
42
```

```
51 CREATE TABLE IF NOT EXISTS user (
52     id INT AUTO_INCREMENT PRIMARY KEY,
53     email VARCHAR(255) NOT NULL UNIQUE,
54     password VARCHAR(255) NOT NULL,
55     roles JSON NOT NULL,
56     reset_password_token VARCHAR(64) COLLATE utf8mb4_unicode_ci DEFAULT NULL
57 );
58
59 CREATE TABLE IF NOT EXISTS veterinary_report (
60     id INT AUTO_INCREMENT PRIMARY KEY,
61     animal_id INT,
62     user_id INT,
63     health_status TEXT,
64     food VARCHAR(100),
65     food_weight DECIMAL(5, 2),
66     report_date DATETIME DEFAULT CURRENT_TIMESTAMP,
67     detail TEXT,
68     FOREIGN KEY (animal_id) REFERENCES animal(id),
69     FOREIGN KEY (user_id) REFERENCES user(id)
70 );
71
72 CREATE TABLE IF NOT EXISTS animal_feeding (
73     id INT AUTO_INCREMENT PRIMARY KEY,
74     animal_id INT,
75     user_id INT,
76     feeding_date DATETIME DEFAULT CURRENT_TIMESTAMP,
77     food VARCHAR(100),
78     quantity DECIMAL(5, 2),
79     FOREIGN KEY (animal_id) REFERENCES animal(id),
80     FOREIGN KEY (user_id) REFERENCES user(id)
81 );
82
```



Création d'un projet Symfony

✓ TERMINAL

- PS C:\wamp64\www\studi-ecf-arcadia> **cd ../**
- PS C:\wamp64\www> **symfony** new exemple-jury --webapp
 - * Creating a new Symfony project with Composer
 - * Setting up the project under Git version control (running git init C:\wamp64\www\exemple-jury)

[OK] Your project is now ready in C:\wamp64\www\exemple-jury

- PS C:\wamp64\www> **cd** exemple-jury
- PS C:\wamp64\www\exemple-jury> █

Modification du fichier .env

```
# DATABASE_URL="sqlite:///kernel.project_dir%/var/data.db"
DATABASE_URL="mysql://root@127.0.0.1:3306/arcadia_db?serverVersion=8.0.32&charset=utf8mb4"
# DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/app?serverVersion=10.11.2-MariaDB&chars
```

Création de
l'entité
Habitat

```
PS C:\wamp64\www\exemple-jury> symfony console make:entity Habitat
created: src/Entity/Habitat.php
created: src/Repository/HabitatRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
> description

Field type (enter ? to see all types) [string]:
> text

Can this field be null in the database (nullable) (yes/no) [no]:
> no

updated: src/Entity/Habitat.php

Add another property? Enter the property name (or press <return> to stop adding fields):
>

Success!

Next: When you're ready, create a migration with symfony.exe console make:migration
```



Relations

▼ TERMINAL

- PS C:\wamp64\www\exemple-jury> **symfony** console make:entity Habitat
Your entity already exists! So let's add some new fields!

New property name (press <return> to stop adding fields):

> animal

Field type (enter ? to see all types) [**string**]:

> relation

What class should this entity be related to?:

> Animal

What type of relationship is this?

Type	Description
ManyToOne	Each Habitat relates to (has) one Animal . Each Animal can relate to (can have) many Habitat objects.
OneToMany	Each Habitat can relate to (can have) many Animal objects. Each Animal relates to (has) one Habitat .
ManyToMany	Each Habitat can relate to (can have) many Animal objects. Each Animal can also relate to (can also have) many Habitat objects.
OneToOne	Each Habitat relates to (has) exactly one Animal . Each Animal also relates to (has) exactly one Habitat .

Relation type? [**ManyToOne**, **OneToMany**, **ManyToMany**, **OneToOne**]:

> OneToMany



Dynamiser l'interface pour consulter les animaux

```
#[Route(path: '/habitat/{id}/animals', name: 'app_habitat_animals', methods: ['GET'])]
16 references | 0 overrides
public function getAnimalsByHabitat(
    Habitat $habitat,
    AnimalRepository $animalRepository,
    VeterinaryReportRepository $veterinaryReportRepository,
): Response
{
    $animals = $animalRepository->findBy(criteria: ['habitat' => $habitat]);

    $animalData = [];
    foreach ($animals as $animal) {
        $veterinaryReports = $veterinaryReportRepository->findBy(criteria: ['animal' => $animal]);

        $reportData = [];
        foreach ($veterinaryReports as $report) {
            $reportData[] = [
                'id' => $report->getId(),
                'health_status' => $report->getHealthStatus(),
                'food' => $report->getFood(),
                'food_weight' => $report->getFoodWeight(),
                'report_date' => $report->getReportDate()->format('Y-m-d H:i:s'),
                'detail' => $report->getDetail(),
            ];
        }

        $animalData[] = [
            'id' => $animal->getId(),
            'name' => $animal->getName(),
            'image' => $animal->getImage(),
            'veterinary_reports' => $reportData,
        ];
    }

    return $this->json(data: [
        'habitat_name' => $habitat->getName(),
        'habitat_detail' => $habitat->getDescription(),
        'animals' => $animalData,
    ]);
}
```



Utilisation de Fetch pour traiter la réponse Json

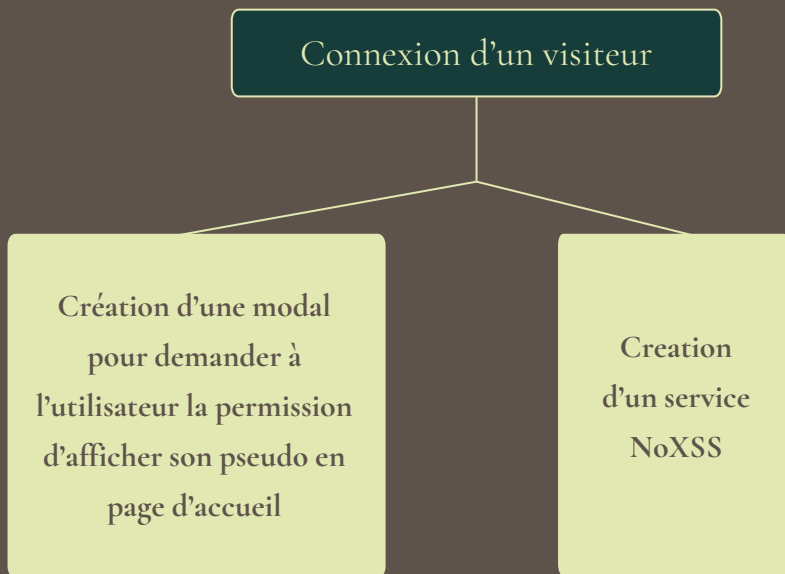
```
document.addEventListener('DOMContentLoaded', () => {
  document.querySelectorAll('.habitat-link').forEach(link => {
    link.addEventListener('click', (event) => {
      event.preventDefault();
      const habitatId = link.getAttribute('data-habitat-id');
      //console.log('Fetching habitat details for habitat:', habitatId);

      fetch(`/habitat/${habitatId}/animals`)
        .then(response => {
          //console.log('Response:', response);
          if (!response.ok) {
            throw new Error('HTTP error! Status: ${response.status}');
          }
          return response.json();
        })
        .then(data => {
          //console.log('Data:', data);
          const animalsList = document.getElementById('animals-list');
          animalsList.innerHTML = '';

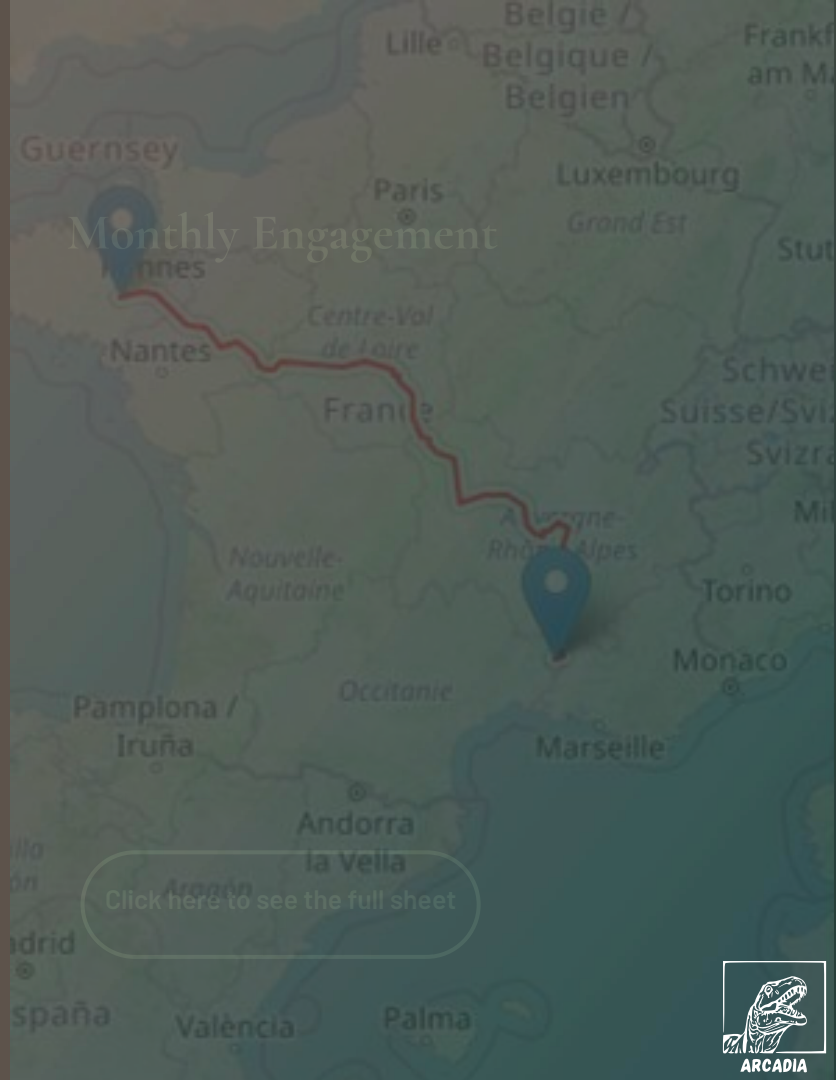
          const habitatDescriptionDiv = document.createElement('div');
          habitatDescriptionDiv.className = 'col-lg-12';
          habitatDescriptionDiv.innerHTML = `
            <div class="card m-4 cadre">
              <div class="card-body">
                <h3 class="card-title">Description de l'habitat (${data.habitat_name})</h3>
                <p>${data.habitat_detail}</p>
              </div>
            </div>
          `;
          animalsList.appendChild(habitatDescriptionDiv);
        });
    });
  });
});
```



Déposer un avis avec pseudo et faille XSS



Afin de respecter la réglementation RGPD et protection contre les failles XSS



Cycle de vie d'une fonctionnalité

- Conception : Choix des technos, recueil informations.
- Développement du service
- Développement des tests
- Intégration
- Déploiement
- Sécurité

Développement

<https://github.com/Exercise/HTMLPurifierBundle>

```
1  <?php
2
3  namespace App\Service;
4
5  use HTMLPurifier;
6  use HTMLPurifier_Config;
7
8  24 references | 0 implementations
9  class NoXSS
10 {
11     11 references | 0 overrides
12     public function nettoyage($text): string
13     {
14         $config = HTMLPurifier_Config::createDefault();
15         $config->set(key: 'HTML.Allowed', value: '');
16         $purifier = new HTMLPurifier(config: $config);
17
18         return $purifier->purify(html: $text);
19     }
20 }
```

<https://github.com/Exercise/HTMLPurifierBundle>

```
1  <?php
2
3  namespace App\Service;
4
5  use HTMLPurifier;
6  use HTMLPurifier_Config;
7
8  24 references | 0 implementations
9  class NoXSS
10 {
11     11 references | 0 overrides
12     public function nettoyage($text): string
13     {
14         $config = HTMLPurifier_Config::createDefault();
15         $config->set(key: 'HTML.Allowed', value: '');
16         $purifier = new HTMLPurifier(config: $config);
17
18         return $purifier->purify(html: $text);
19     }
20 }
```



Développement

```
$avisAAfficher = $reviewRepository->findBy(criteria: ['valid' => true]);  
foreach ($avisAAfficher as $review) {  
    $review->setComment($this->noXSS->nettoyage(text: $review->getComment()));  
    $review->setPseudo($this->noXSS->nettoyage(text: $review->getPseudo()));  
}
```

```
$review = new Review();  
$form = $this->createForm(type: ReviewType::class, data: $review);  
$form->handleRequest(request: $request);  
  
if ($form->isSubmitted()) {  
    $acceptPseudo = $request->request->get(key: 'accept_pseudo') === '1';  
  
    if (!$acceptPseudo) {  
        $this->addFlash(type: 'error', message: 'Vous devez accepter que votre pseudo soit affiché pour soumettre un avis.');    } elseif ($form->isValid()) {  
        $review->setComment(comment: $this->noXSS->nettoyage(text: $review->getComment()));  
        $review->setPseudo(pseudo: $this->noXSS->nettoyage(text: $review->getPseudo()));  
        $em->persist(object: $review);  
        $em->flush();  
  
        $this->addFlash(type: 'success', message: 'Votre avis a été soumis et est en attente de validation.');  
        return $this->redirectToRoute(route: 'app_home');    } else {  
        $this->addFlash(type: 'error', message: 'Erreur lors de la soumission de votre avis');    }  
}
```



Développement

Mise en place de la modal

```
const confirmButton = document.getElementById("confirmButton");  
const acceptPseudoInput = document.getElementById("accept_pseudo");  
const form = document.querySelector(".form-control.avis form");
```

```
// Active le bouton de soumission après confirmation  
if (confirmButton && form) {  
  confirmButton.addEventListener("click", function () {  
    if (acceptPseudoInput) {  
      acceptPseudoInput.value = "1";  
    }  
    form.submit();  
    const modal = bootstrap.Modal.getInstance(document.getElementById('confirmModal'));  
    modal.hide();  
  });  
};
```



Développement des tests

```
tests > NoXSSTest.php > NoXSSTest
1  <?php
2
3  namespace App\Tests;
4
5  use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;
6  use App\Service\NoXSS;
7
8  0 references | 0 implementations
9  class NoXSSTest extends WebTestCase
10
11     0 references | 0 overrides
12     public function testNoXSS(): void
13     {
14         $xss = '<script>alert("XSS")</script>';
15         $nettoyeur = new NoXSS();
16         $propre = $nettoyeur->nettoyage(text: $xss);
17         $this->assertSame(expected: $propre, actual: "");
```

• WebTestCase :

- Utilise les fonctionnalités de test de Symfony pour exécuter et valider le comportement du service `NoXSS`.

Création du test :

- Entrée : Une chaîne contenant un script malveillant

```
<script>alert("XSS")</script>.
```

Sortie attendue :

- Une chaîne vide après purification

• Vérification avec : `assertSame`

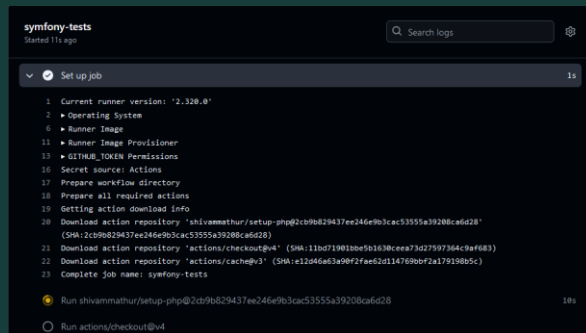
- S'assure que le contenu malveillant est totalement supprimé

Le résultat attendu est une chaîne vide ""



Intégration continue

```
- name: Install Dependencies
  run: composer install -q --no-ansi --no-interaction --no-scripts --no-progress --prefer-dist
- name: Create Database
  run: |
    mkdir -p data
    touch data/database.sqlite
- name: Clear Symfony cache
  run: php bin/console cache:clear --env=test
- name: Display Symfony logs
  run: tail -n 50 var/log/test.log
- name: Start Symfony server
  run: php -S 0.0.0.0:8000 -t public/ > /dev/null 2>&1 &
- name: Execute tests (Unit and Feature tests) via PHPUnit
  env:
    DATABASE_URL: sqlite:///kernel.project_dir/data/database.sqlite
  run: vendor/bin/phpunit
```



All workflows

Showing runs from all workflows

90 workflow runs



Add test NoXSS phpunit

TestDev #36: Commit [42bee69](#) pushed by MrLeoufff



Add test for map

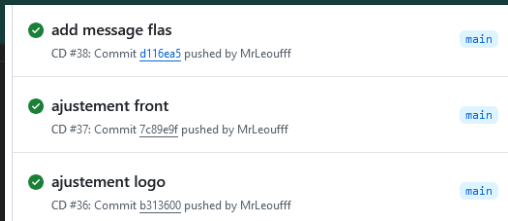
TestDev #35: Commit [4886a19](#) pushed by MrLeoufff



Déploiement continu

Objectif : Automatiser le processus de mise à jour de l'application sur le serveur et garantir un déploiement fiable et rapide à chaque modification du code.

```
1 name: CD
2 on:
3   push:
4     branches:
5       - main
6
7 jobs:
8   deploy:
9     runs-on: ubuntu-latest
10
11   env:
12     SSHPASS: ${ secrets.SSHPASS }
13
14   steps:
15     - name: Checkout repository
16       uses: actions/checkout@v2
17
18     - name: Install sshpass & update apt
19       run: |
20         sudo apt update
21         sudo apt install sshpass
22
23     - name: Deploy application
24       run: |
25         sshpass -e ssh -o strictHostKeyChecking=no ${{ secrets.SSH_USER }}@${{ secrets.SSH_IP }} "
26         cd htdocs &&
27         cd reneleliard.online/ &&
28         git pull https://github.com/MrLeoufff/ecf-arcadia.git &&
29         composer install
30         "
```



Etapes Principales :

- Checkout du code : GitHub Actions télécharge automatiquement la dernière version de votre dépôt.
- Installation des dépendances : installer toutes les bibliothèques nécessaires à l'application.

Déploiement sur le serveur :

- Connexion sécurisée via SSH.
- Mise à jour du code sur le serveur distant avec « git pull ».
- Relance des services nécessaires (installation des dépendances, clear cache, etc.).

Bénéfices :

Réduction des erreurs humaines grâce à l'automatisation.
Déploiement rapide après chaque validation de code.
Processus reproductible pour toutes les versions.



Mécanisme de sécurité

➤ **Authentification :**

Gestion des utilisateurs via des
firewalls.
Supporte les méthodes :
formulaire de connexion, JWT, OAuth2.
Mots de passe hashés avec
« bcrypt »

➤ **Autorisation :**

Gestion des rôles
(ROLE_ADMIN, ROLE_EMPLOYE ...)

➤ **Validation des Données Entrantes :**

Contraintes de validation
(@assert).
Nettoyage des données avec un
service comme HTMLPurifier.

➤ **Protection CSRF :**

Utilisation de tokens CSRF
dans les formulaires pour prévenir les
attaques.



Opération et qualité



Regroupement de donnée (conteneurisation) pour un environnement de développement :

- Docker et autres outils

Conformité et légalité

- Protection des données : conformité avec les réglementations (ex. GDPR)

Accessibilité :

- normes et implémentation pour l'inclusion



Docker est une plateforme de virtualisation légère basée sur des conteneurs



Permet d'emballer des applications et leurs dépendances dans des conteneurs pour une exécution portable



Principalement utilisé pour le développement, les tests et la mise en production

Avantages :

Rapidité, isolation, reproductibilité



Concepts clés de Docker

```
Dockerfile
1 FROM php:8.3.0-fpm
2 LABEL authors="René"
3
4 # Installer les extensions PHP nécessaires
5 RUN apt update && apt install -y \
6     libpng-dev \
7     libjpeg-dev \
8     libzip-dev \
9     libfreetype6-dev \
10    libonig-dev \
11    libxslt1-dev \
12    unzip \
13    build-essential \
14    git \
15    nginx \
16    libssl-dev \
17    default-mysql-client \
18    && docker-php-ext-configure gd --with-freetype --with-jpeg \
19    && docker-php-ext-install -j$(nproc) gd \
20    && docker-php-ext-install pdo_mysql \
21    && docker-php-ext-enable opcache \
22    && docker-php-ext-install xsl \
23    && docker-php-ext-install zip \
24    && docker-php-ext-install intl \
25    && docker-php-ext-install soap \
26    && pecl install mongodb \
27    && docker-php-ext-enable mongodb
28
29 # Installer Composer
30 COPY --from=composer:2.7.7 /usr/bin/composer /usr/bin/composer
31
32 # Copier les fichiers de l'application
33 WORKDIR /var/www/html
34
35 COPY . .
36
37 COPY build/nginx/conf/default.conf /etc/nginx/conf.d/default.conf
38 COPY build/php/custom.ini /usr/local/etc/php/conf.d/
39
40 COPY entrypoint.sh /usr/local/bin/entrypoint.sh
41 COPY build/supervisord.conf /etc/supervisor/conf.d/supervisord.conf
42
```

```
42
43 # Rendre le script exécutable
44 RUN chmod +x /usr/local/bin/entrypoint.sh
45 RUN mv "$PHP_INI_DIR/php.ini-production" "$PHP_INI_DIR/php.ini"
46
47 # Installer les dépendances Composer
48 ENV COMPOSER_ALLOW_SUPERUSER=1
49 RUN composer install --no-interaction --optimize-autoloader
50
51 # Configurer les permissions
52 RUN chown -R www-data:www-data /var/www/html \
53     && chmod -R 755 /var/www/html
54
55 # Assurez-vous que le répertoire des hydrators est accessible en écriture
56 RUN mkdir -p /var/www/html/var/doctrine \
57     && chown -R www-data:www-data /var/www/html/var/doctrine \
58     && chmod -R 777 /var/www/html/var/doctrine
59
60 # Exposer le port 9000
61 EXPOSE 9000
62
63 # Démarrer Apache
64 # CMD service nginx start && php-fpm
65 # Utiliser supervisord pour gérer les processus
66 RUN apt-get install -y supervisor
67 COPY build/supervisord.conf /etc/supervisor/conf.d/supervisord.conf
68
69 ENTRYPOINT ["entrypoint.sh"]
70 CMD ["supervisord", "-c", "/etc/supervisor/conf.d/supervisord.conf"]
```

- **Conteneur** : Instance en cours d'exécution d'une image.
- **Image** : Modèle immuable pour créer des conteneurs.
- **Dockerfile** : Script pour définir les étapes de création d'une image.
- **Registry** : Répertoire pour stocker et partager des images (ex. Docker Hub).

Commandes Docker Essentielles



« docker build » :
Construire une image à partir d'un Dockerfile.



« docker run » :
Démarrer un conteneur à partir d'une image.



« docker ps » :
Lister les conteneurs en cours d'exécution.



« docker stop » :
Arrêter un conteneur.



« docker exec » : Exécuter une commande dans un conteneur en cours d'exécution.

Cas d'Utilisation de Docker

- **Développement** : Créer un environnement isolé pour chaque projet.
- **Déploiement** : Déployer des applications via des conteneurs pour plus de portabilité.
- **Scalabilité** : Gérer plusieurs instances avec des orchestrateurs comme Kubernetes.
- **CI/CD** : Automatiser les builds et tests dans des conteneurs.


Avantages et Limites

Avantages :

- Isolation des applications.
- Portabilité entre les environnements.
- Démarrage rapide des conteneurs.

Limites :

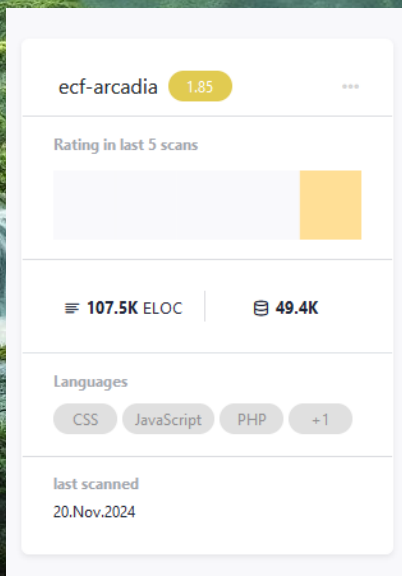
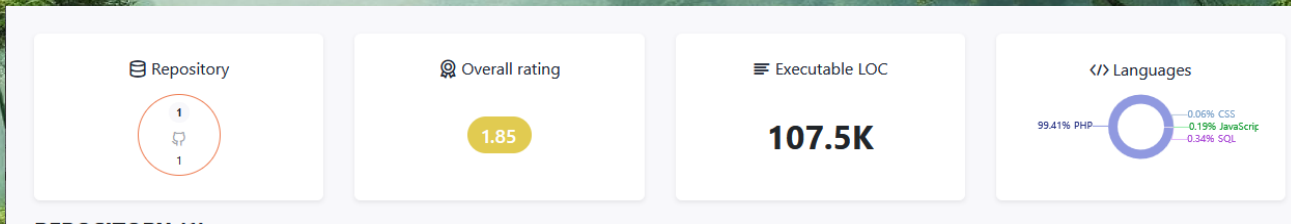
- Courbe d'apprentissage pour les débutants.
- Nécessite une gestion des ressources pour éviter la surcharge.
- Pas adapté pour certaines applications nécessitant un accès direct au matériel



RGPD (Règlement Général sur la Protection des Données)

- **Introduction :**
 - Le RGPD est un cadre juridique européen adopté en 2018 pour protéger les données personnelles.
- **Principes clés :**
 - **Transparence** : Informer clairement les utilisateurs sur l'utilisation de leurs données.
 - **Consentement** : Obtenir un accord explicite avant de collecter des données.
 - **Sécurité** : Assurer la confidentialité et l'intégrité des données.
- **Droits des utilisateurs :**
 - **Accès** : Consulter les données collectées.
 - **Rectification** : Modifier ou corriger leurs informations.
 - **Suppression** : Droit à l'oubli.
- **Impact pour les entreprises :**
 - Conformité obligatoire sous peine de sanctions financières.
 - Mise en place de politiques de gestion des données.

Vérification de la qualité du code:



Veille technologique

**Developpez.com**

FORUMS TUTORIELS FAQ BLOGS CHAT NEWSLETTER EMPLOI ÉTUDES DROIT CLUB

 [DI/DSI Solutions d'entreprise](#) [Cloud](#) [IA](#) [ALM](#) [Microsoft](#) [Java](#) [Dév. Web](#) [EDI](#) [Programmation](#) [SGBD](#) [Office](#) [Mo](#)

TUTORIELS FAQ WEBINAIRES LIVRES TÉLÉCHARGEMENTS SOURCES DÉBATS WIKI DICO HUMOUR

À LA UNE



Herb Sutter, président de l'ISO C++, quitte Microsoft après 22 années de collaboration et déclare que la prochaine version de C++ 26 est « la plus importante depuis C++11 »

Herb Sutter, l'un des leaders les plus influents de la communauté C++ et actuel président du comité ISO C++, quitte Microsoft après 22 années de collaboration. Sutter a surpris la communauté en déclarant que C++ 26, la prochaine grande version du langage prévue pour 2026, serait la version « la plus impactante depuis C++11 ». Cette annonce a de quoi susciter l'enthousiasme et l'impatience, tant C++11 avait introduit des avancées majeures et modernisé le langage de manière significative

 3 commentaires



Les Européens passent 575 millions d'heures par an à cliquer sur des bannières de cookies qui interrompent leurs flux de travail, subissant ainsi des pertes économiques importantes et de productivité

Les bannières de cookies constituent un élément central du Web moderne et sont censées contribuer à la protection de la vie privée en ligne. Mais elles ont aussi un impact négatif important. Elles interrompent le flux de travail, créant une « fatigue de la bannière » et nuisant à la productivité. Pour les entreprises, ces interruptions entraînent des coûts cachés, des pertes de temps et une baisse de la satisfaction des utilisateurs

 18 commentaires

ACTUALITÉ INFORMATIQUE



La récession des emplois dans la tech : après une décennie marquée par une croissance des effectifs sans précédent, les embauches dans le secteur de l'informatique ont chuté de 27 % selon LinkedIn

 135  15:56



Un scandale provoqué par des deepfakes sexuellement explicites générés par l'IA entraîne la fermeture d'une école en Pennsylvanie, les défis liés à la technologie deepfake exacerbés par l'IA générative

 526  15:05



ReSharper 2024.3 disponible : prise en charge complète des fonctionnalités de C# 13, amélioration du nettoyage et du formatage de

 0

PUBLICATIONS INFORMATIQUE



[Raspberry Pi Pico][SDK C/C++] Le convertisseur analogique-numérique : erreurs DNL et INL - Partie 2/2, un billet blog de f-leb

 0  10:12



[Raspberry Pi Pico][SDK C/C++] Test du convertisseur analogique-numérique - Partie 1/2, un billet blog de f-leb

 2  15/11

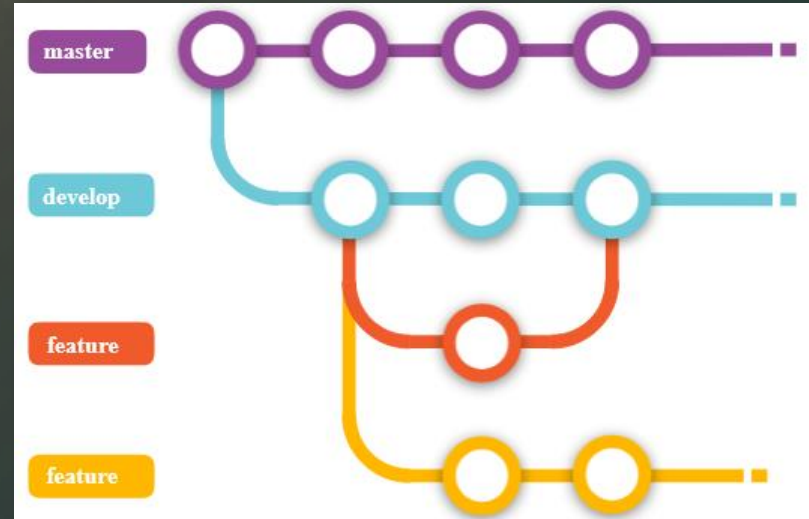
Organisation :

- Branche Main -> Branche de production
- Branche Dev -> Branche de développement
- Branche XXX -> Une branche par fonctionnalités

Pourquoi avoir utilisé Git et GitHub

Ensemble, Git et GitHub offrent un cadre robuste pour le développement de logiciels, de la gestion du code source à la collaboration en équipe et à l'intégration continue. Ils sont indispensables pour les équipes de développement modernes, permettant une gestion efficace des projets de toute taille

Git & GitHub

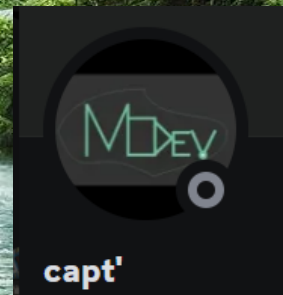
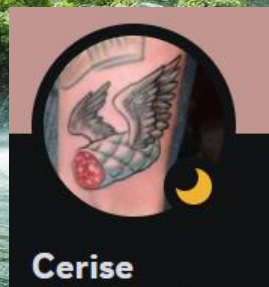


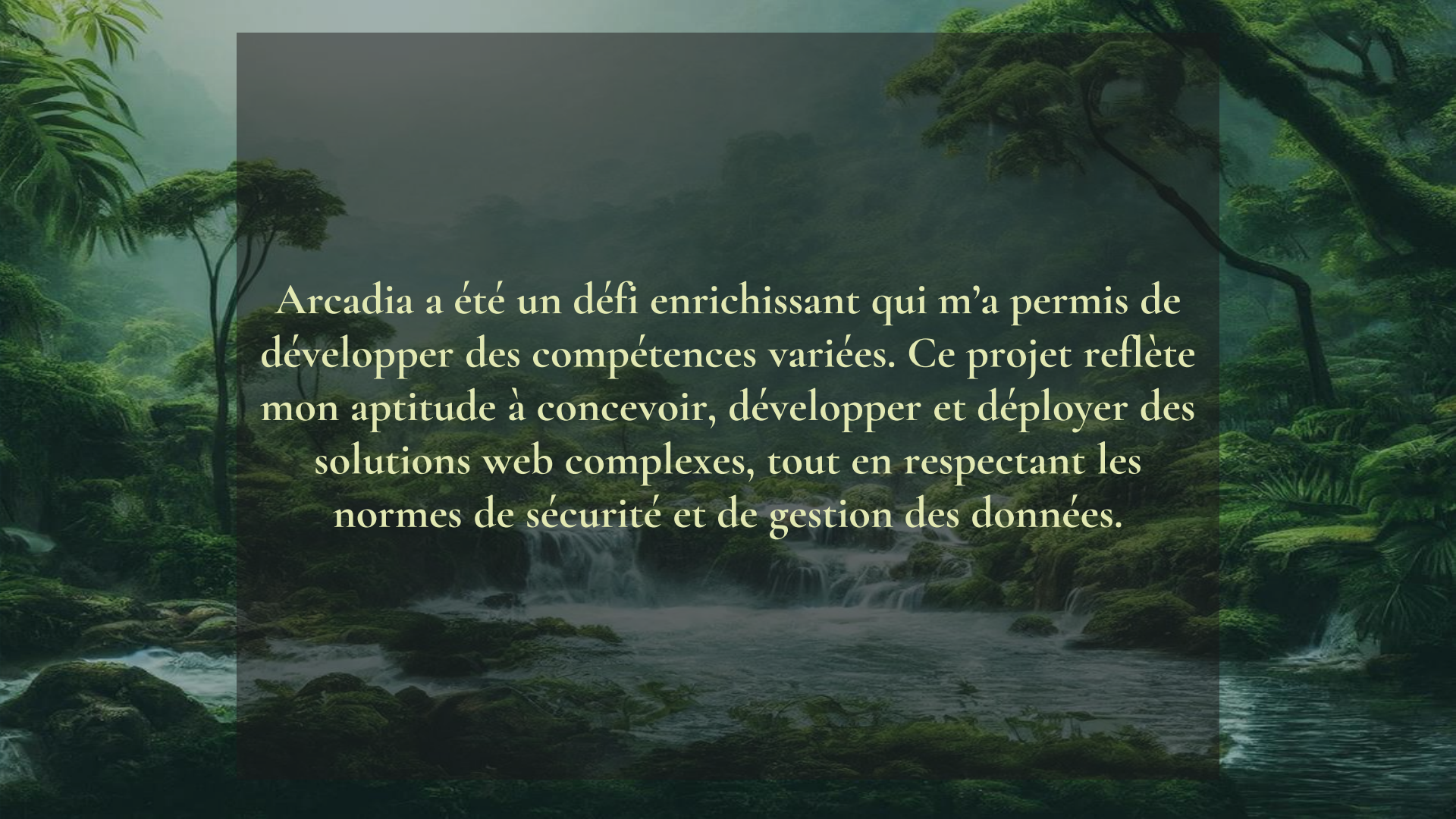
studi

mdn



 Symfony





Arcadia a été un défi enrichissant qui m'a permis de développer des compétences variées. Ce projet reflète mon aptitude à concevoir, développer et déployer des solutions web complexes, tout en respectant les normes de sécurité et de gestion des données.