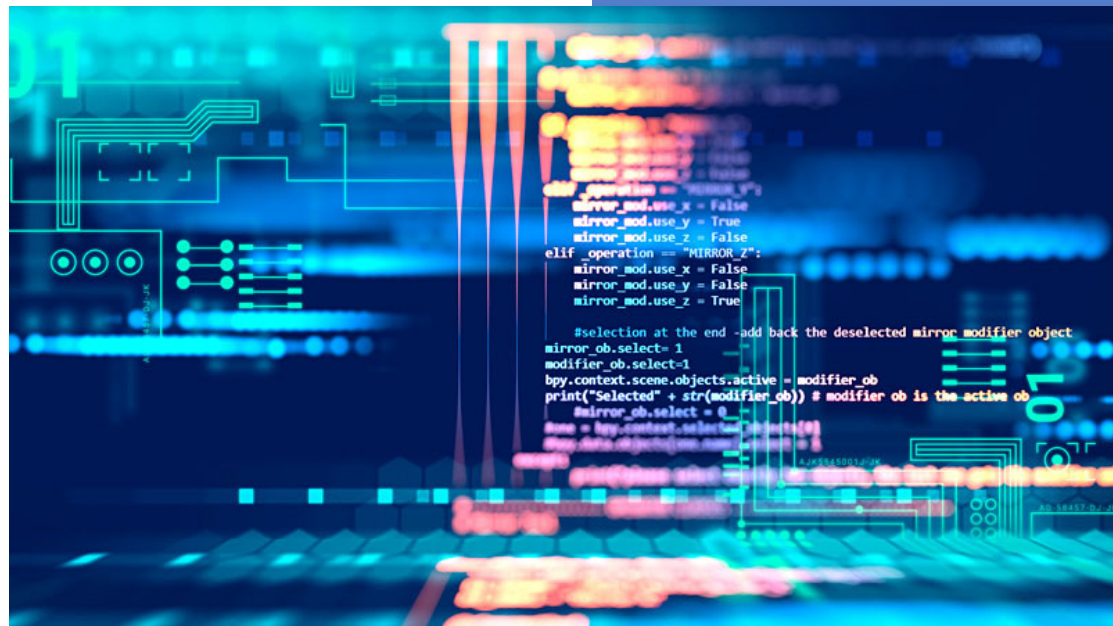


Développeur Web et Web Mobile



REMERCIEMENTS

Je tiens à remercier toutes les personnes qui m'ont aidé lors de la rédaction de ce rapport.

Mr Mathias Gaudin, formateur à l'AFPA qui a su se rendre disponible à tout moment. Son écoute et ses conseils m'ont permis de progresser et d'avancer sur mon projet.

Mr Julien Colomb, mon maître de stage, développeur au sein de la société Sokeo pour son accueil, le temps passé ensemble et le partage de son expertise. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats.

Enfin, je tiens à remercier toutes les personnes qui m'ont soutenu tout le long de ma formation.

SOMMAIRE

1. Compétences du référentiel couvertes par le projet	3
2. Cahier des charges	5
2.1. Présentation du projet	5
2.2. Spécifications fonctionnelles	5
• Fonctionnement de l'application.....	5
• Users stories.....	6
• Arborescence du site web.....	8
• Charte graphique	8
• Maquettage	9
2.3. Spécifications techniques.....	11
• Editeur de code.....	11
• Gestion de version et stockage.....	11
• L'hébergement du site web	11
• Choix de l'architecture logicielle.....	11
• Langages web	13
3. Réalisation du projet	14
3.1. Mise en œuvre du contrôleur frontal	14
• Routage.....	14
• Analyse des paramètres transmis	15
• Gestion des erreurs	15
3.2. Réalisation et développement front end	17
• Structure type d'une page html	17
• Référencement naturel ou SEO.....	18
• Gestion du contenu.....	20
• Personnalisation du thème de Bootstrap avec SASS.....	20
• Affichage dynamique d'images avec JavaScript.....	23
3.3. Développement back end.....	24
• Création de la base de données	24
• Veille technologique sur les vulnérabilités de sécurité	26
• Développer les composants d'accès aux données	26
• Présentation d'un jeu d'essai.....	30
4. Veille technologique réalisée au cours du stage	31
5. Annexes.....	35

1. COMPETENCES DU REFERENTIEL COUVERTES PAR LE PROJET

Activités types 1

*Développer la partie front-end d'une application web ou web mobile
en intégrant les recommandations de sécurité*

- Maquetter une application
- La maquette prend en compte les spécificités fonctionnelles décrites dans les cas d'utilisation ou les scénarios utilisateur
- La maquette est conforme à l'expérience utilisateur et à l'équipement ciblé
- La maquette respecte les principes de sécurisation d'une interface utilisateur

Réaliser une interface utilisateur web statique et adaptable

- L'interface est conforme à la maquette de l'application
- Les bonnes pratiques de structuration sont respectées y compris pour le web mobile
- Les pages web s'adaptent à la taille de l'écran
- Le site respecte les règles de référencement naturel
- La démarche de recherche permet de résoudre un problème technique ou de mettre en œuvre une nouvelle fonctionnalité

Développer une interface utilisateur web dynamique

- Les pages web sont conformes à l'expérience utilisateur y compris pour l'expérience mobile
- L'architecture de l'application répond aux bonnes pratiques de développement et de sécurisation d'application web
- L'application web est optimisée pour les équipements mobiles
- La démarche de recherche permet de résoudre un problème technique ou de mettre en œuvre une nouvelle fonctionnalité
- La veille sur les vulnérabilités connues permet d'identifier des failles potentielles

Activités types 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Créer une base de données

- La base de données est conforme au schéma physique
- Les scripts de création de bases de données et d'insertion des données de test s'exécutent sans erreurs
- Les besoins de sécurité du SGBD sont exprimés selon l'état de l'art et les exigences de sécurité identifiées
- La démarche de recherche permet de résoudre un problème technique ou de mettre en œuvre une nouvelle fonctionnalité

Développer les composants d'accès aux données

- Les traitements relatifs aux manipulations des données répondent aux fonctionnalités décrites dans le dossier de conception technique
- Les composants d'accès à la base de données suivent les règles de sécurisation reconnues
- La démarche de recherche permet de résoudre un problème technique ou de mettre en œuvre une nouvelle fonctionnalité
- La veille sur les vulnérabilités connues permet d'identifier des failles potentielles

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

- Les bonnes pratiques de développement sont respectées
- Les composants serveur contribuent à la sécurité de l'application
- Le code source des composants est documenté ou auto-documenté
- La démarche de recherche permet de résoudre un problème technique ou de mettre en œuvre une nouvelle fonctionnalité
- La veille sur les vulnérabilités connues permet d'identifier des failles potentielles

2. CAHIER DES CHARGES

2.1 Présentation du projet

J'ai choisi de vous présenter un projet personnel, ayant l'avantage d'aborder toutes les étapes de développement d'un site web et de remplir les critères d'évaluation de mes compétences professionnelles.

Ce projet est un guide sur la reconnaissance des champignons.

A travers ce site je partage une de mes passions et propose à ceux qui le désirent de participer à l'élaboration d'une base de données collaboratives, regroupant photos et descriptions d'espèces.

2.2 Spécifications fonctionnelles

Fonctionnement de l'application

Comme le site propose une documentation illustrée, et collaborative je dois pouvoir stocker ces informations dans une base de données.

Pour commencer j'ai défini trois types d'utilisateurs :

- Visiteur
- Membre
- Administrateur

Fonctionnalités en fonction des permissions

Les 3 types d'utilisateurs ont accès à l'espace public

- Une page d'accueil
- A la liste des espèces donnant accès au détail de chacune des
 - Fiches descriptives

Les membres visualisent les informations de profil, la liste des fiches qu'ils ont créées, accèdent à travers des liens aux pages :

- Ajouter une fiche descriptive
- Modifier une fiche descriptive
- Modifier ou supprimer son profil

Restrictions

Lors de la création d'une fiche descriptive par un membre, celui-ci peut la consulter mais ne sera accessible dans l'espace public qu'après validation par l'administrateur.

L'administrateur peut visualiser l'ensemble des comptes utilisateurs, la liste des fiches présentes sur le site. En outre il a la possibilité de masquer une fiche dans l'espace public, supprimer un compte ou une fiche descriptive.

Users stories

Cette étape décrit les fonctionnalités à développer. Bien que cette étape ne soit pas suffisamment détaillée elle me permet d'identifier mes objectifs.

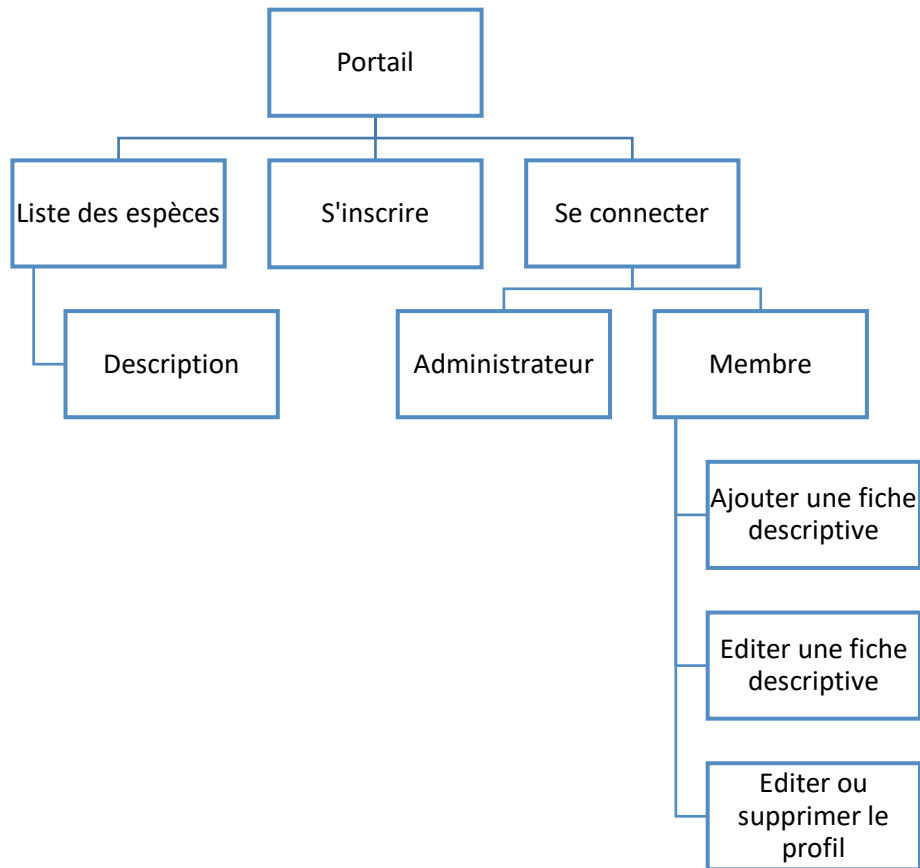
En tant que...	Je souhaite...	Afin de...
Visiteur	Consulter la liste des espèces	Visualiser chaque espèce sous forme de vignette m'informant sur le nom et sa comestibilité.
Visiteur	M'inscrire	Créer mon profil et bénéficier du statut de membre.
Membre	Me connecter	Gérer mon espace personnel.
Membre	Me déconnecter	Quitter mon espace.
Membre	Voir mon profil	Voir mes informations personnelles.
Membre	Modifier mon profil	Changer/corriger mes informations personnelles.
Membre	Créer une fiche descriptive	Collaborer au développement du site en créant une documentation illustrée.
Membre	Modifier une fiche descriptive	Mettre à jour le contenu d'une fiche détaillée.

Tableau n°1 récapitulatif des users stories

En tant que...	Je souhaite...	Afin de...
Membre	Supprimer une fiche	Eliminer des données erronées.
Membre	Supprimer mon profil	Quitter la communauté et répondre aux exigences du RGPD.
Administrateur	Consulter la listes des utilisateurs	Accéder aux informations détaillées pour chaque utilisateur inscrit.
Administrateur	Supprimer un utilisateur	Modérer les utilisateurs.
Administrateur	Visualiser les fiches au moyen d'un tableau regroupant les images miniatures, les noms, dates de création et noms d'auteurs.	D'avoir une vue globale sur la documentation rédigée par la communauté.
Administrateur	Consulter une fiche descriptive.	M'assurer que son contenu n'enfreint pas la législation.
Administrateur	Supprimer une fiche	Modérer le contenu si celle-ci ne respecte pas la législation.
Administrateur	Autoriser la publication d'une fiche descriptive	De la rendre accessible au visiteur

Tableau n°2 récapitulatif des users stories

Arborescence du site web

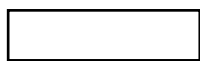


Charte graphique

C'est l'identité visuelle du site, j'ai choisi des couleurs en accord avec la nature, une courte vidéo en page d'accueil permet de retenir l'attention du visiteur.

Pour personnaliser mon site, j'ai choisi d'intégrer des images d'arrière-plan légèrement floutées pour agrémenter les pages.

Code couleurs



rgb (255, 255, 255) : liens



rgb(224, 139, 94) : liens actifs



rgb(214, 206, 195) : liens au survol



rgb(85, 84, 84) : couleur de texte



rgb(63, 80, 62) : bouton de pagination, titre de chaque vignette



rgba(Texte 0.8) : formulaires

Police de caractères

J'ai choisi une combinaison de 3 polices de caractères :

- **Roboto** que j'utilise pour la barre de navigation et le pied de page
- **Tangerine** employée pour le titre principal et secondaire de la page d'accueil
- **Libre Baskerville** appliquée sur le contenu de chaque page.

Maquettage



J'ai utilisé **Adobe XD** pour réaliser les maquettes dédiées aux support mobile et Desktop.

J'ai tout d'abord découpé visuellement chaque page en plusieurs zones, cette organisation appelée le **zoning** me simplifiera la tâche lors du développement.

En effet cela me permet d'identifier les différents blocs qui constituent la page.

J'ai ensuite réalisé le **Mockup**, rendu visuel étant le plus proche possible du produit final, après m'être inspiré en parcourant quelques sites et divers ouvrages.



Aperçu du zoning de la page d'accueil



Mockup page d'accueil

VOIR LES MAQUETTES EN ANNEXE 1 - 2 - 3 - 4

2.3 Spécifications techniques

Editeur de code



Pour développer mon projet j'utilise Vscode un éditeur open source, il prend en charge presque tous les principaux langages de programmation et fourni des extensions téléchargeables gratuitement.

Gestion de version et stockage



Je me sers de **Git**, un logiciel gestion de versions.

Il me permet de réaliser des sauvegardes régulières de mon projet en local et sur un serveur d'hébergement comme **GitHub**.



C'est un outil incontournable pour tout développeur, il offre en outre, la possibilité de partager son code avec d'autres développeurs.

L'hébergement du site web



Les offres gratuite d'hébergement propose souvent des packs PHP MySQL, ce qui a orienté mon choix sur les technologies web. J'ai choisi d'utiliser 000webhost qui se présente comme une plate-forme d'apprentissage avec une prise en main aisée.

<https://petit-guide-des-champignons.000webhostapp.com/>

Choix de l'architecture logicielle

Pour la réalisation de mon projet j'ai choisi une architecture MVC personnalisée se décomposant en trois parties : Modèle, Vue, Contrôleur.

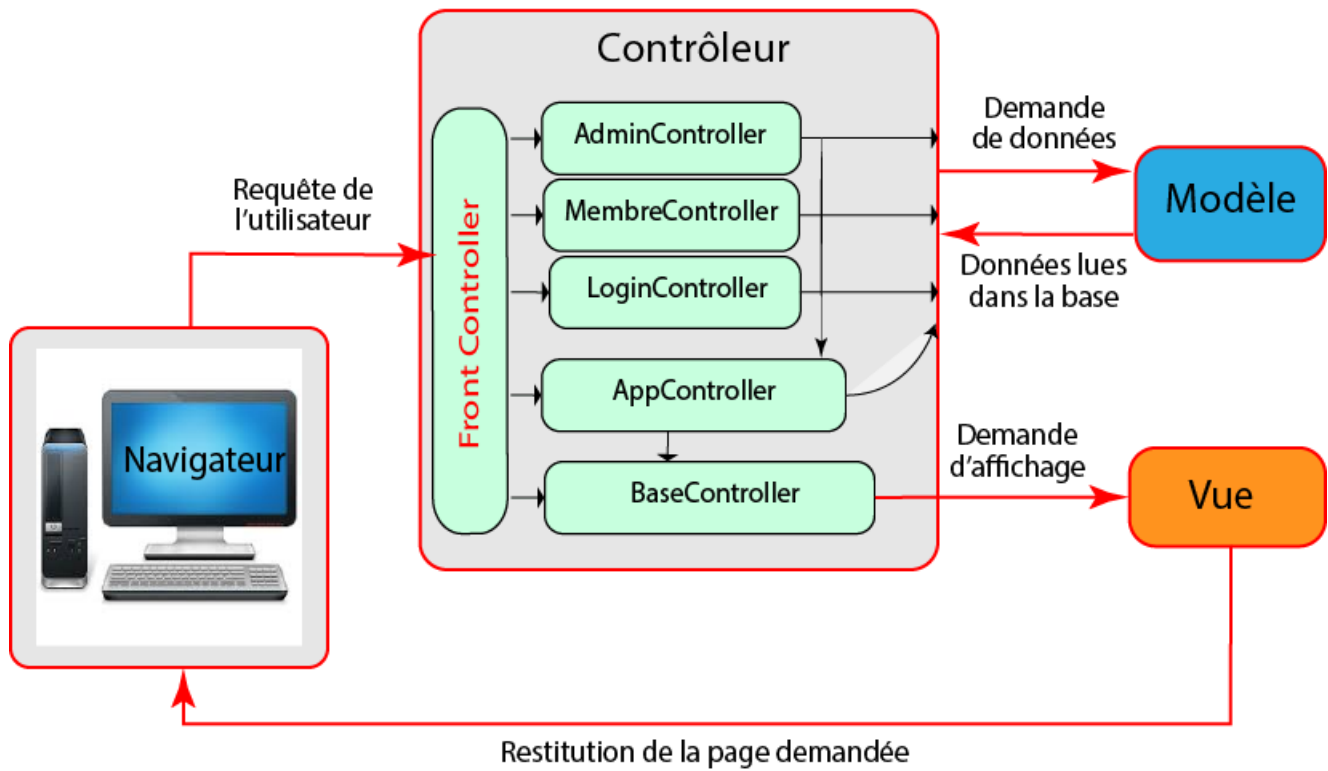
Ainsi je sépare les responsabilités, le code est mieux organisé, plus lisible et donc plus facile à maintenir.

Le **Modèle** gère les données stockées dans une base.

La **Vue** permet l'interaction avec l'utilisateur, présentation, saisie et validation des données.

Le **Contrôleur** fait le lien entre l'utilisateur et le reste de l'application.

ARCHITECTURE MVC



Langages web

Langages pour le développement front-end



Dans la première étape de développement, je structure mes documents, mets en forme le contenu de mes pages tel que je l'ai défini dans la phase de maquettage.

Les zonings élaborent l'architecture de mes pages, pour cela j'utilise **HTML** qui est un langage de balise.

CSS – SASS - BOOTSTRAP



Le langage **CSS** change l'apparence des éléments HTML, donne du style à mes pages, c'est grâce à ce langage que je vais pouvoir appliquer l'apparence esthétique réalisée avec mes mockups.



Ce préprocesseur CSS, je l'emploie conjointement avec la méthodologie BEM (Bloc Élément Modifier), il rend mon code plus lisible mieux organisé et surtout structuré.



Avec ce Framework CSS je développe et je conçois des pages rapidement, et j'écris moins de code.

L'utilisation des grilles m'a permis de découper et d'organiser mes pages web en fonction du support numérique.

Je me suis servi de plusieurs composants pour illustrer mes pages, comme Bootstrap exploite jQuery, j'obtiens un rendu dynamique pour certains composants.



JavaScript a été utile pour afficher un rendu des images lorsque un membre souhaite créer ou éditer une fiche descriptive.



J'emploi Twig, moteur de Template pour personnaliser, assembler, à l'aide de blocs, mes pages web. Le résultat offre un code plus clair et lisible, en évitant les répétitions, par ailleurs, le traitement des données est séparé.

Langage pour le développement back end



PHP est un langage coté serveur que je peux utiliser conjointement avec ma base de données relationnelle de type SQL.

3. REALISATION DU PROJET

3.1 Mise en œuvre du contrôleur frontal

J'ai mis en œuvre un Contrôleur Frontal à partir du fichier « index.php » qui constitue le point d'entrée unique du site.

Routage

Son rôle permet de centraliser la gestion des requêtes entrantes.

Les paramètres de la route sont transmis dans L'URL sous forme de clé valeur.

<https://.../index.php?routing=portail>

Dans cet exemple un point d'interrogation sépare le nom de la page « index.php », des paramètres de routage « routing=portail ».

```
// ROUTAGE PAR DEFAULT (premier chargement)
$page = 'portail';
if(isset($_GET['routing'])) {
    $page = $_GET['routing'];
}
// ROUTING
```

Paramètre de routage du contrôleur frontal

Un bloc « try » placé au plus haut niveau de l'application, centralise la gestion des erreurs, bien que cela puissent ralentir le chargement de la page cette solution a l'avantage de simplifier les procédures de contrôle.

Le bloc « try » entoure la commande « switch » une structure conditionnelle qui parcourt et vérifie l'existence de la route passée dans l'URL.

```
// ROUTING
try
{
    switch ($page)
    {
        /*******
        | *          ACCES PUBLIC
        |*****/
        // Rendu de la page d'accueil.
        case 'portail':
            $appController->home();
            break;
```

Bloc try entourant la commande switch

Analyse des paramètres transmis

Dans cet exemple suivant, je souhaite afficher la page listant toutes les espèces de champignons sous forme de vignettes, en apportant quelques informations (comestibilité, nom commun).

Je vérifie les paramètres fournis, sélectionne et appelle **l'action** à réaliser.

```
// Rendu page liste des vignettes.
case 'guide-des-champignons':
    $page = isset($_GET['pageIndex']) ? $_GET['pageIndex'] : '1';
    if(is_numeric($page))
    {
        $appController->getListMushrooms($page,isset($_GET['allFiches']));
    }
    else
    {
        throw new ExceptionWithRedirect("Erreur impossible de charger la page !",
        404, "guide-des-champignons");
    }
break;
```

Instruction case du contrôleur frontal : index.php

Gestion des erreurs

Dans l'exemple vu précédemment, dans le cas où la requête est incohérente je signale l'erreur à l'utilisateur en **lançant une exception avec** ("[*throw*](#)").

J'ai édité une classe nommée « `ExceptWhithRedirect` » qui hérite de la classe « `Exception` » fourni par PHP, ceci me permet d'ajouter un lien de redirection dans la page d'erreur.

J'instancie la classe avec « `new` ».

```
<?php
class ExceptionWithRedirect extends Exception {

    protected $redirect_url;

    function __construct($message = "", $code = 0, $redir_url = "portail")
    {
        parent::__construct($message, $code);
        $this->redirect_url = $redir_url;
    }

    public function getRedirectUrl(){
        return $this->redirect_url;
    }
}
```

Gestion des erreurs

La méthode constructeur est automatiquement appelée au moment de l'instanciation de la classe « ExceptWhithRedirect ».

Par ailleurs j'ai besoin du constructeur parent (Exception) pour attraper l'erreur il est donc nécessaire de faire appel à « *parent::__construct()* ».

J'attrape l'erreur avec ("[*catch*](#)").

```
catch(Exception $e)
{
    $appController->handleError($e);
}
```

Je fais le rendu de la page d'erreur grâce à l'objet Twig.

```
function handleError(Exception $e){
    echo $this->render('erreur.html.twig',[
        'error' => $e,
        'with_redirect' => is_a($e, "ExceptionWithRedirect")
    ]);
}
```

Méthode du contrôleur controller/app.php

L'utilisateur consulte la page d'erreur et active la redirection vers une autre page du site.



3.2 Réalisation et développement front-end

Mon projet est bâti sur une architecture MVC, pour constituer mes pages j'utilise twig qui sépare le code PHP du code HTML.

Structure type d'une page html

Un document HTML se compose de plusieurs balises :

- **<!DOCTYPE html>** : définit le type de document
- **<html ></html>** : délimite le début et la fin de la page
- **<head></head>** : définit le codage des caractères, contrôle la mise en page sur les navigateurs mobiles, fournit des informations sur l'auteur du site, indique les liens vers les feuilles de style, informe sur le contenu de la page.
- **<body></body>** : contient la partie visible de la page, aussi appelée « corps de page ». C'est ce qui sera affiché à l'écran.

J'édite dans un fichier Twig la structure commune à toutes les pages du site.

En-tête du Layout. « default.html.twig »

Référencement naturel ou SEO

Je cherche à améliorer la visibilité de mon site sur les pages de résultats des moteurs de recherche et plates-formes de réseaux sociaux.

J'ai ajouté dans la balise **<head>** de mon layout parent, 3 blocs (title, meta et.opengraph), qui seront renseignés dynamiquement lors de la génération du rendu par twig.

```
<!-- Titre de la page -->
<title>
    {% block title %}{% endblock %}
</title>

<!-- meta description -->
<link rel="shortcut icon" type="image/x-icon" href="/assets/icones/logo.png" />
{% block meta %}{% endblock %}

<!-- autoriser l'indexation par les moteurs de recherche -->
<meta name="robots" content="index">

<!-- SEO opengraph -->
<meta property="og:locale" content="fr_FR"/>
<meta property="og:type" content="website"/>
{% block og %}{% endblock %}
```

Layout : default.html.twig

Je dois à présent définir le titre et les métadonnées pour chaque Template enfant du site. Voici un exemple pour la réalisation de la page d'accueil.

Balise <title>

La balise <title> est importante. Elle est utilisée par les moteurs de recherche pour indexer évaluer et positionner la page web et les navigateurs pour donner un titre à la page au niveau de l'onglet, comme titre des marque-pages (favoris), etc. Elle doit refléter le contenu de la page et contenir des mots clés pertinents.

```
{% block title %}
    Guide des champignons - base de données collaborative !
{% endblock %}
```

Titre de la page d'accueil

Balise meta description

Elle renseigne les moteurs de recherche sur le sujet traité et sera ajoutée par Google dans la page de résultat pour détailler le contenu.

```
{% block meta %}
    <meta name="description" content="Reconnaître facilement un
    champignon comestible ? Trouvez ces informations sur le guide des
    champignons. Participer à l'élaboration d'une base de données
    interactive."/>
{% endblock %}
```

Meta description de la page d'accueil

Balise meta robots

Elle donne les directives aux moteurs de recherche sur la manière d'explorer, indexer et parcourir les liens sur une page.

Balise Open Graph

Les balises Open Graph permettent d'améliorer la présentation de mes pages web lorsque des liens sont partagés sur les réseaux sociaux.

```
{% block og %}
    <meta property="og:title" content="Guide des champignons - base de
    données collaborative !"/>
    <meta property="og:description" content="Reconnaître facilement un
    champignon comestible ? Trouvez ces informations sur le guide des
    champignons. Participer à l'élaboration d'une base de données
    collaborative."/>
    <meta property="og:url" content="{{url}}/index.php?routing=portail"/>
    <meta property="og:image" content="{{url}}/assets/icones/logo.png"/>
    <meta property="og:image:width" content="150"/>
    <meta property="og:image:height" content="150"/>
{% endblock %}
```

Meta données Open Graph de la page d'accueil

Gestion du contenu

L'élément **<body>** représente le contenu principal du document HTML, visible par l'utilisateur.

```
<body>
  <section class="d-flex flex-column default">
    <header>
      {% include 'includes/navbar.html.twig' %}
    </header>

    <main class="flex-grow-1">
      {% block body %}{% endblock %}
    </main>

    <footer>
      {% include 'includes/footer.html.twig' %}
    </footer>

    {% include 'includes/script.html.twig' %}
  </section>
</body>
</html>
```

Balise body du layout principal

Dans cet exemple j'ai placé dans un conteneur flexible la balise **<header>**, **<main>** et **<footer>**, ces balises dites sémantiques définissent le rôle du contenu qu'elles englobent.

J'ai ajouté dans la balise **<main>** une classe « flex-grow-1 » pour qu'elle remplisse l'espace restant, ainsi le footer se positionnera toujours en bas de page.

Le `{% block body %}` permet d'injecter le contenu (images texte...) de manière dynamique lors du rendu.

Personnalisation du thème de Bootstrap avec SASS

Je souhaite ajouter les couleurs définies dans la charte graphique, modifier les couleurs du composant de pagination ainsi que de la barre de navigation.

Pour réaliser cette opération proprement, je dois intégrer les **fichiers sources** de Bootstrap dans mon projet.

Travaillant sous Windows, j'installe au préalable **Node JS** pour bénéficier du gestionnaire de package **NPM**.

J'installe Bootstrap avec la commande : `$ npm install bootstrap`

Puis **SASS**, le préprocesseur CSS : `$ npm install -g sass` pour compiler mon code CSS et celui de bootstrap et enfin je définis dans le fichier « package.json » un script pour automatiser la compilation.

```
"scripts": {  
  "compile-sass": "node-sass sass/main.scss public_html/assets/css/main.  
  css --watch",  
  "test": "echo \"Error: no test specified\" && exit 1"  
},  
...
```

Je crée un fichier que je nomme « change-colors-bootstrap.scss » et j'ajoute les couleurs personnalisées.

```
// ajout de couleurs personnalisées  
// Titre et texte description  
$gray: #555;   
// Pagination  
$greenDark: #38761d;  
// Navbar active, nom latin  
$orange: #f57614;  
// Formulaire  
$greenLight: #5cb85c;  
// composants Navs  
$whiteLight: #fff;  
$navigate: #337ab7;
```

J'attribue ces couleurs aux composants cités en amont.

```
// Modification de la couleur du composant de pagination pour la classe active.  
$pagination-active-bg: $white;  
$pagination-active-color: $greenDark;  
// Modification de la couleur du composant nav pour la classe active.  
$nav-pills-link-active-color: $whiteLight;  
$nav-pills-link-active-bg: $whiteLight;  
// Modification de la couleur des liens.  
$link-color: $whiteLight;  
// Modification de la couleur barre de navigation theme navbar-dark.  
$navbar-dark-color: $white;  
$navbar-dark-hover-color: #337ab7;  
$navbar-dark-active-color: $orange;  
// Modification de la couleur du bouton lorsque la navbar s'est effondrée.  
$navbar-dark-toggler-border-color: $white;
```

Je dois maintenant l'ajouter au thème par défaut de Bootstrap.

```
$theme-colors: (  
  gray: $gray,  
  whiteLight: $whiteLight,  
  greenDark: $greenDark,  
  greenLight: $greenLight,  
  orange: $orange,  
  navigate: $navigate,  
  pagination-active-bg: $pagination-active-bg,  
  nav-pills-link-active-color: $nav-pills-link-active-color,  
  nav-pills-link-active-bg: $nav-pills-link-active-bg,  
  link-color: $link-color,  
  navbar-dark-color: $navbar-dark-color,  
  navbar-dark-hover-color: $navbar-dark-hover-color,  
  navbar-dark-active-color: $navbar-dark-active-color,  
  navbar-dark-toggler-border-color: $navbar-dark-toggler-border-color  
);
```

Et pour terminer j'importe Bootstrap

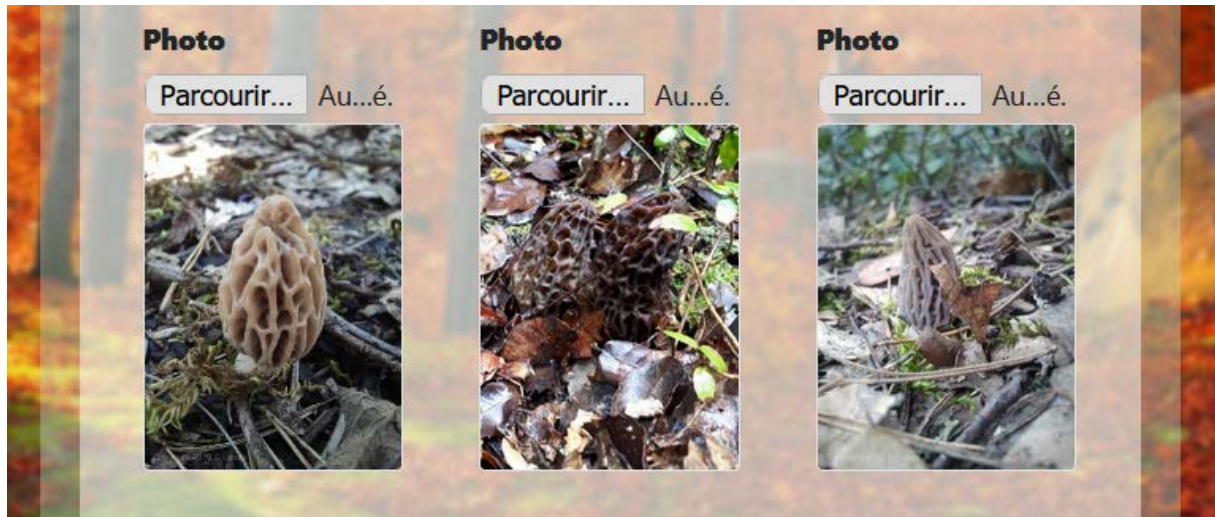
```
@import "../node_modules/bootstrap/scss/bootstrap";
```

Je lance dans le terminal le compilateur SASS, pour obtenir un fichier CSS combinant mon code SCSS et celui de Bootstrap.

```
lanza@LAPTOP-1VAEL500 MINGW64 /c/wamp64/www/2020-04-25-guide-des-champignons-dos  
sier_public-query-string (master)  
$ npm run compile-sass  
> encyclopedie@1.0.0 compile-sass C:\wamp64\www\2020-04-25-guide-des-champignons  
-dossier_public-query-string  
> node-sass sass/main.scss public_html/assets/css/main.css --watch
```


Affichage dynamique d'images avec JavaScript

Je souhaite afficher une miniature lorsqu'un membre ajoute ou modifie une image.



L'extrait de code ci-dessous permet de sélectionner et afficher une image.

addOrupdateFile.html.twig

```
<div class="row p-3 added__upload" id="added__upload">
...
<input type="file" class="form-control-file" img="photo-1" name="photo[]" >

```

J'ai rédigé un script client avec ECMAScript, un langage reconnu par tous les navigateurs.

La fonction écoute un évènement dans le conteneur d'image.

javascript.js

```
document.querySelector('.added__upload').addEventListener('change',(event)
=> {
let my_element = event.target;
let fichierSelectionne = my_element.files[0];
const idPhoto = my_element.getAttribute("img");
document.getElementById(idPhoto).src = window.URL.createObjectURL
(fichierSelectionne);
});
```


Lorsque l'utilisateur sélectionne une image :

Je stocke dans la variable « my_element » la balise input.

La variable « fichierSelectionne » récupère l'objet file qui correspond aux fichiers sélectionnés.

La constante « idphoto » me permet de cibler la balise image qui devra afficher la photo.

Enfin je transmets à la balise image la référence de l'objet binaire stocké dans la mémoire du navigateur pour afficher la miniature.

3.3 Développement back end

Création de la base de données

J'ai choisi d'utiliser une base de données de type relationnel pour conserver les informations de profils utilisateurs et la documentation relative aux différentes fiches descriptives.

J'utilise l'utilitaire JMerise pour modéliser ma base de données.

Je recueille et regroupe les informations à traiter en un ensemble cohérent qui devront figurées sur le site.

Je récence trois groupes de données liés les uns aux autres.

Celle des utilisateurs, fiches descriptives et images puis je rédige **le dictionnaire de données**.

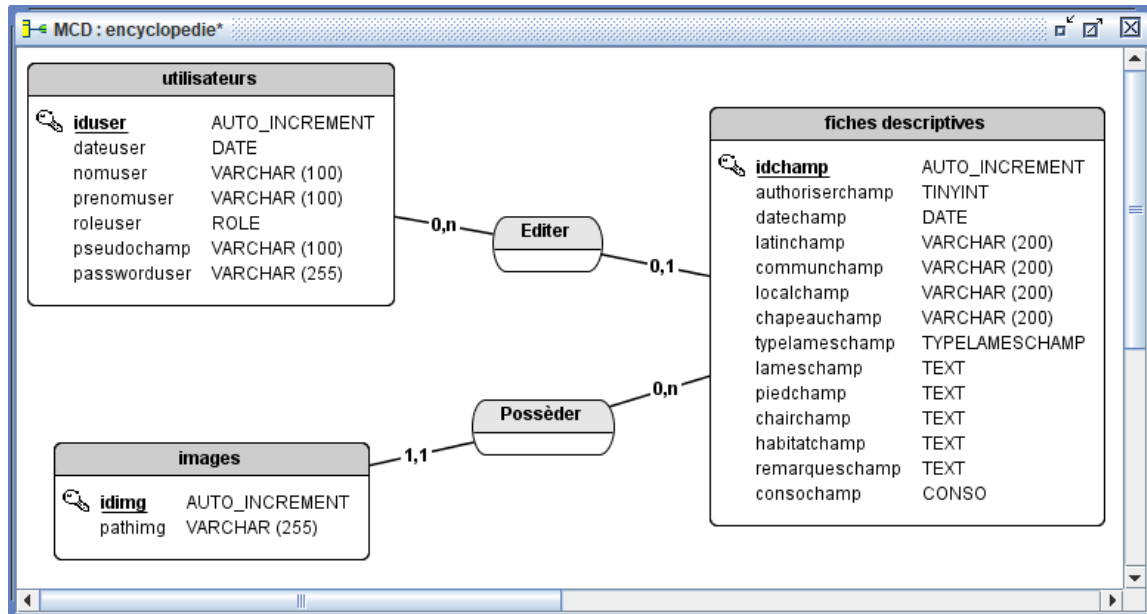
Num	Nom	Code	type	taille	decim...	Util...
1	iduser	IDUSER	Auto_increment			<input type="checkbox"/>
2	dateuser	DATEUSER	Date			<input type="checkbox"/>
3	nomuser	NOMUSER	Varchar	100		<input type="checkbox"/>
4	prenomuser	PRENOMUSER	Varchar	100		<input type="checkbox"/>
5	speudouser	SPEUDouser	Varchar	100		<input type="checkbox"/>
6	idchamp	IDCHAMP	Auto_increment			<input type="checkbox"/>
7	datechamp	DATECHAMP	Date			<input type="checkbox"/>
8	autoriserchamp	AUTORISERCHAMP	TinyINT			<input type="checkbox"/>
9	latinchamp	LATINCHAMP	Varchar	200		<input type="checkbox"/>

Le modèle conceptuel de données

J'élabore le Modèle Conceptuel de Données (MCD) basé sur deux notions principales : les **entités** et les **associations**.

Je définis les relations :

- 1 utilisateur possède de 0 à n fiche mais une fiche appartient à 0 ou 1 utilisateur.
- 1 fiche possède 0 à n images mais une image appartient à 1 et 1 seule fiche.

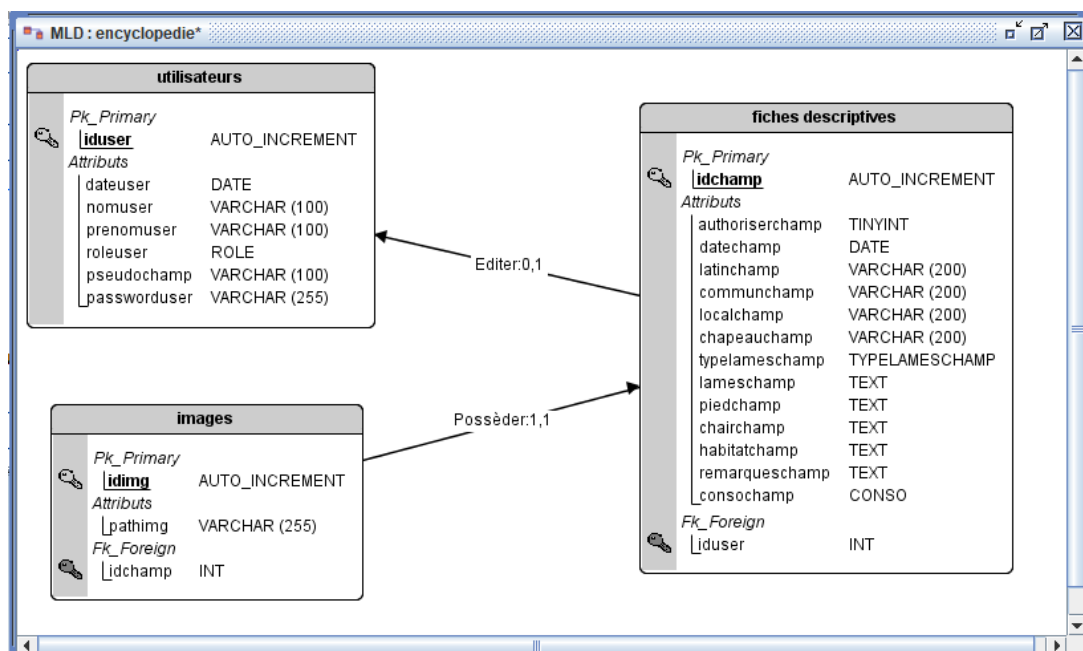


Modèle Conceptuel de données

Le modèle physique des données

A cette étape, les entités deviennent des tables, les identifiants deviennent des clés primaires.

L'utilitaire génère les tables en fonction des cardinalités décrites ci-dessus.



Modèle Physique de données

Le script

Je récupère le script généré par JMerise et j'ajoute deux contraintes de table pour la suppression d'enregistrement.

- « ON DELETE SET NULL », Lorsque l'utilisateur supprime son profil, la clé étrangère des fiches qu'il aura créée sera définie a « NULL ».
- « On DELETE CASCADE », si l'utilisateur supprime une fiche, les données images liées, seront automatiquement détruites.

```
--  
-- Contraintes pour la table `champ`  
--  
ALTER TABLE `champ`  
  ADD CONSTRAINT `champ_utilisateurs_FK` FOREIGN KEY (`iduser`) REFERENCES `utilisateurs` (`iduser`) ON DELETE SET NULL;  
--  
-- Contraintes pour la table `images`  
--  
ALTER TABLE `images`  
  ADD CONSTRAINT `images_ibfk_1` FOREIGN KEY (`idchamp`) REFERENCES `champ` (`idchamp`) ON DELETE CASCADE;  
COMMIT;
```

Veille technologique sur les vulnérabilités de sécurité

Pour limiter les failles de sécurité dans la phase de développement de mon application, j'ai identifié plusieurs risques :

- La faille XSS
- L'injection SQL

Ces vulnérabilités sont traitées dans les paragraphes suivants :

Développer et sécuriser les composants d'accès aux données

Création d'un compte utilisateur

Sur la page d'inscription, l'utilisateur saisit dans le formulaire un pseudo et un mot de passe et valide son choix.

La faille de sécurité XSS (cross-site scripting) : permet l'injection de code HTML ou JavaScript dans des variables mal protégées.

En développant mes Template avec Twig j'évite l'injection de code malveillant, en effet par défaut Twig protège les variables en appliquant un filtre qui échappe le texte affiché.

Validation des données : un premier contrôle de validation rend obligatoire le remplissage des champs, côté client en ajoutant dans les balises « input » l'attribut « require » et contrôle la conformité du mot de passe avec l'attribut « pattern »

Une requête http est envoyée au serveur, le contrôleur frontal « index.php » vérifie l'existence de la route et appelle la méthode dans le contrôleur cible.

```
// Créer un compte
case 'creation-du-compte':
    // si une session est ouverte on redirige vers la page admin
    if($appController->get_value_session())
    {
        header("location:index.php?routing=mon-compte");
    }
    else
    {
        // Détermine si une variable est déclarée et est différente de NULL
        if(isset($_POST['pseudo']) && isset($_POST['password']) )
        {
            $loginController->createAnAccount($_POST['pseudo'],$_POST
            ['password']);
        }
        else
        {
            throw new ExceptionWithRedirect("Une erreur c'est produite !",
            404, "portail");
        }
    }
}
break;
```

Contrôleur frontal

Dans le contrôleur cible j'exécute un second contrôle de validation des données cette fois coté back end et renvoie les messages d'erreur de saisie à l'utilisateur.

```
// Password
if(empty($password)){
    $messageErrorPassword = "Merci de saisir un mot de passe";
}
else
{
    // Vérifie la conformité du mot de passe
    if(preg_match('#(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}#', $password))
    {
        $isOkPassword = true;
    }
    else
    {
        $messageErrorPassword = "Votre mot de passe n'est pas conforme.";
    }
}
}
```

Vérifie la conformité du mot de passe

Si les informations du profil sont conformes elles sont enregistrées dans la base de données, procédure détaillée au paragraphe modèle.

```
// Si le pseudo et password sont conforme, créer le compte.
if( $isOkPseudo && $isOkPassword)
{
    // Création du compte
    $response = $this->sqlCommande->createAccount($pseudo,$password);
    if($response['profilUser'])
    {
        $_SESSION['user'] = $response['profilUser'];
        // Redirection vers la page membre
        header('location:index.php?routing=mon-compte');
    }
    else
    {
        // Si bug pendant l'enregistrement
        throw new ExceptionWithRedirect("Une erreur c'est produite !", 404,
            "portail");
    }
}
}
```

Création du compte et ouverture d'une session

Le modèle

J'ai choisi d'utiliser MySQL un système de gestion de base de données relationnelles open source et gratuit, ses performances sont excellentes et multi-utilisateurs et prévu pour fonctionner parfaitement avec PHP.

Pour me connecter et communiquer avec ma base de données j'utilise PDO (**PHP Data Objects**), un middleware ou interface logiciel.

L'extension « pdo_mysql » activée par défaut autorise l'accès de PHP aux bases de données MySQL.

L'accès à la base de données se fait en instanciant un objet de la classe PDO. Les paramètres à indiquer au constructeur sont :

- La source de la base de données
- Le jeu de caractères utilisés
- L'identifiant et le mot de passe.

```
// CONNECTION A MYSQL
$this->_dataBase = new PDO('mysql:host=localhost:3306;dbname='.$myDB.';
charset=utf8', 'root', '');
```

Je stocke dans la variable « \$database » l'objet PDO.

Ajout du profil utilisateur dans la base de données

Sécuriser le mot de passe

Par sécurité, le mot de passe doit être crypté avant d'être stocké dans la base de données. En effet, il pourrait être volé si le support de stockage (typiquement la base de données) est compromis.

PHP fournit la fonction « password_hash » pour créer une clé de hachage. La constante « PASSWORD_DEFAULT » est conçue pour changer dans le temps, au fur et à mesure que des algorithmes plus récents et plus forts sont ajoutés à PHP.

Sécuriser ses requête SQL

Pour se protéger de la faille de sécurité appelée « **injection SQL** » qui consiste à modifier une requête SQL en injectant des morceaux de code généralement par le biais d'un formulaire, j'ai utilisé les requêtes préparées avec des marqueurs nommés.

Exécution de la requête SQL

La méthode « bindParam » me permet de lier les variables « \$pseudo » et « \$password » aux marqueurs nommés. La requête est exécutée et le profil est ajouté à la base de données.

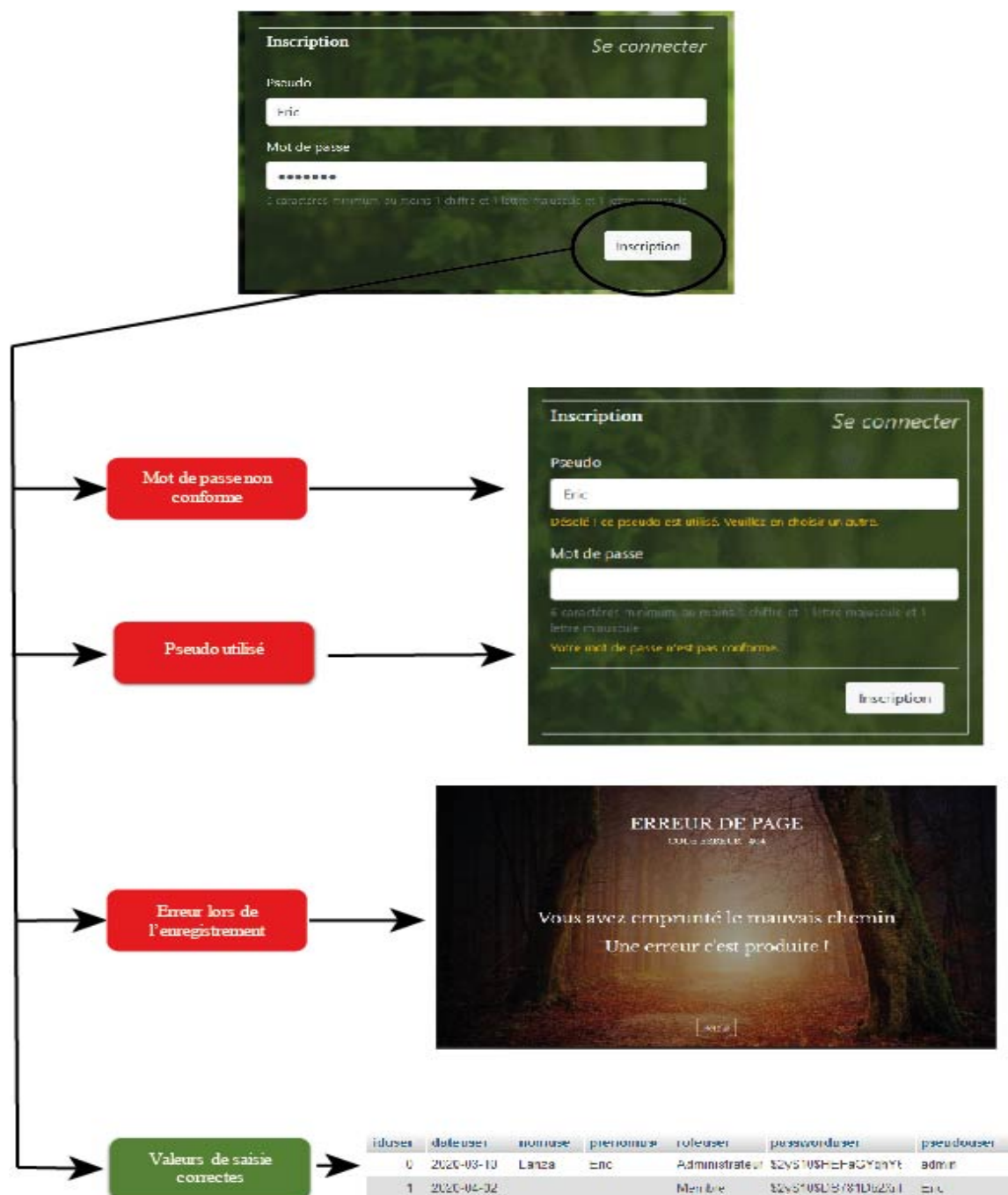
```
public function createAccount($pseudo,$password)
{
    $is_exist_pseudo = $this->isExistPseudo($pseudo);
    $hash_password = password_hash ($password ,PASSWORD_DEFAULT);
    $query = $this->_dataBase->prepare('INSERT INTO utilisateurs (pseudouser,
passworduser) VALUES (:pseudouser, :passworduser)');
    $query->bindParam(':pseudouser', $pseudo);
    $query->bindParam(':passworduser', $hash_password);
    $query->execute();
    $query->closeCursor();
    $id = $this->_dataBase->lastInsertId();
    $query = $this->_dataBase->prepare('SELECT * FROM utilisateurs where iduser
= :id');
    $query->bindParam(':id', $id);
    $query->execute();
    $result = $query->fetch(PDO::FETCH_ASSOC);
    $query->closeCursor();
    return ['profilUser' => $result];
}
```

Présentation d'un jeu d'essai

Description du jeu d'essai pour la création d'un compte utilisateur.

Pour s'inscrire l'utilisateur saisit dans le formulaire un pseudo, un mot de passe et valide son choix.

Les données en entrée	Les données attendues
<ul style="list-style-type: none"> - Pseudo - Mot de passe 	<ul style="list-style-type: none"> - Pseudo - Mot de passe crypté
Les contraintes	
Pseudo unique dans la base de données Mot de passe avec au moins 6 caractères, un chiffre, une lettre minuscule et 1 lettre majuscule.	



4. VEILLE TECHNOLOGIQUE (PERIODE DE STAGE)

J'ai réalisé mon stage auprès de la société Sokeo. Cette jeune startup située à Marseille est spécialisée dans le développement de sites, d'applications web, et propose d'optimiser le référencement naturel SEO.

Les technologies web évoluent en permanence, il est donc important de se tenir régulièrement informé.

Les missions qui m'ont été confiées reposent sur l'utilisation du Framework Cake PHP. J'ai effectué des veilles technologiques afin d'exploiter les outils fournis par le Framework.

Envoyer un email avec CV via un serveur SMTP.

La société Ikosoft fournit une solution de gestion pour les salons de coiffure et SPA. A partir de leur page recrutement, où plusieurs offres d'emplois sont publiées, je dois envoyer un email à l'administrateur du site et joindre le CV du postulant en pièce-jointe. Un formulaire recueille les données à transmettre (l'intitulé du poste et le CV au format PDF).

VOIR ANNEXE 6

Je recherche dans la documentation du Framework.

Voici ce qu'indique la documentation officielle de Cake PHP.

Email

Before version 3.1, the Email and Transport classes were under the Cake\Network\Email namespace instead of the Cake\Mailer namespace.

```
class Cake\Mailer\Email(mixed $profile = null)
```

Email is a new class to send email. With this class you can send email from any place inside of your application.

Basic Usage

First of all, you should ensure the class is loaded:

```
use Cake\Mailer\Email;
```

After you've loaded Email, you can send an email with the following:

```
$email = new Email('default');  
$email->from(['me@example.com' => 'My Site'])  
->to('you@example.com')  
->subject('About')  
->send('My message');
```


Email has several methods for defining recipients - `to()`, `cc()`, `bcc()`, `addTo()`, `addCc()` and `addBcc()`. The main difference being that the first three will overwrite what was already set and the latter will just add more recipients to their respective field.

Configuration

Configuration for Email defaults is created using `config()` and `configTransport()`. You should put your email presets in the **config/app.php** file. The **config/app.default.php** file is an example of this file. It is not required to define email configuration in **config/app.php**.

Configuring Transports

Email messages are delivered by transports. Different transports allow you to send messages via PHP's `mail()` function. Configuring transports allows you to keep configuration data out of your application code and makes deployment simpler as you can simply change the configuration data.

Voici ma traduction

Email

Avant la version 3.1, les classes Email et Transport étaient utilisables avec l'espace de nom `Cake\Network\Email` alors que maintenant on l'utilise avec l'espace de nom `Cake\Mailer`

```
class Cake\Mailer\Email(mixed $profile = null)
```

Email est une nouvelle classe. Avec cette classe vous pouvez envoyer des Email à partir de n'importe quel endroit de votre application.

Utilisation basic

Premièrement, vous devez vous assurer que la classe est chargée.

```
use Cake\Mailer\Email;
```

Après avoir chargé Email, vous pouvez envoyer des Email avec ce qui suit :

```
$email = new Email('default');  
$email->from(['me@example.com' => 'My Site'])  
    ->to('you@example.com')  
    ->subject('About')  
    ->send('My message');
```

Email offre plusieurs méthodes pour définir les destinataires- `to()`, `cc()`, `bcc()`, `addTo()`, `addCc()`. La grande différence est que les trois premières méthodes vont

réinitialiser ce qui était déjà défini alors que les suivantes vont ajouter des destinataires dans les champs respectifs

Configuration

On utilise `config()` et `configTransport()` pour la configuration par défaut. Vous devez mettre vos préconfigurations d'email dans le fichier `config/app.php`. Le fichier `config/app.default.php` est un exemple. Ce n'est pas une obligation de définir la configuration d'email dans le fichier `conf.php`.

Configuration des transports

Les mails peuvent être délivrés par transport. Différents transports permettent d'envoyer des messages avec la fonction `mail()`. La configuration des transports permet de séparer les données du code et simplifie le déploiement puisque pouvez simplement changer les données de configuration.

Je configure le fichier « `app.php` » en ajoutant les informations de transport SMTP fournies par « `SendInBlue` ». Puis je développe une méthode pour envoyer le mail avec le CV.

VOIR ANNEXE 7

Autres tâches réalisées

Refonte du site Gontero

J'ai réalisé la refonte du site web de la société Gontero, entreprise spécialisée dans l'exploitation de carrières.

A partir du contenu extrait du site existant et d'une maquette, j'ai développé plusieurs interfaces. Le site originel n'étant pas responsif et adaptable, l'utilisation du Framework CSS Bootstrap m'a permis d'effectuer rapidement cette tâche.

Ajouter un contact client sur le compte sendInBlue.

Ikosoft propose de remporter une somme d'argent en jouant à un jeu. Pour participer l'utilisateur doit trouver 3 fleurs en parcourant le site web, ensuite il remplit un formulaire avec ses informations personnelles.

Je dois récupérer les informations personnelles de l'utilisateur (nom, prénom, adresse mail et numéro de téléphone) et les ajouter aux contacts sur SendInBlue.

Après quelques recherches sur la méthode à employer, j'ai trouvé une librairie utilisable sous PHP, la source est disponible sur l'hébergeur GitHub. J'ai suivi la documentation et installé la librairie, puis j'ai développé cette fonctionnalité dans un Contrôleur.

Mettre en cache les pages du site Ikosoft.

Pour accélérer le temps de chargement des pages du site web, j'ai utilisé la classe cache avec les méthodes « ::read » et « ::write ».

VOIR ANNEXE 8

Réalisation d'interfaces web

A partir de maquettes fournies par la société Ikosoft je devais réaliser plusieurs pages web proposant des conseils marketing et tutoriels destinées aux clients du logiciel.

VOIR ANNEXE 9 - 10

Ikosoft souhaite élargir sa clientèle au-delà du territoire français, le site web devient multilingue.

Cake PHP fournit les fonctionnalités internationalisation et de localisation.

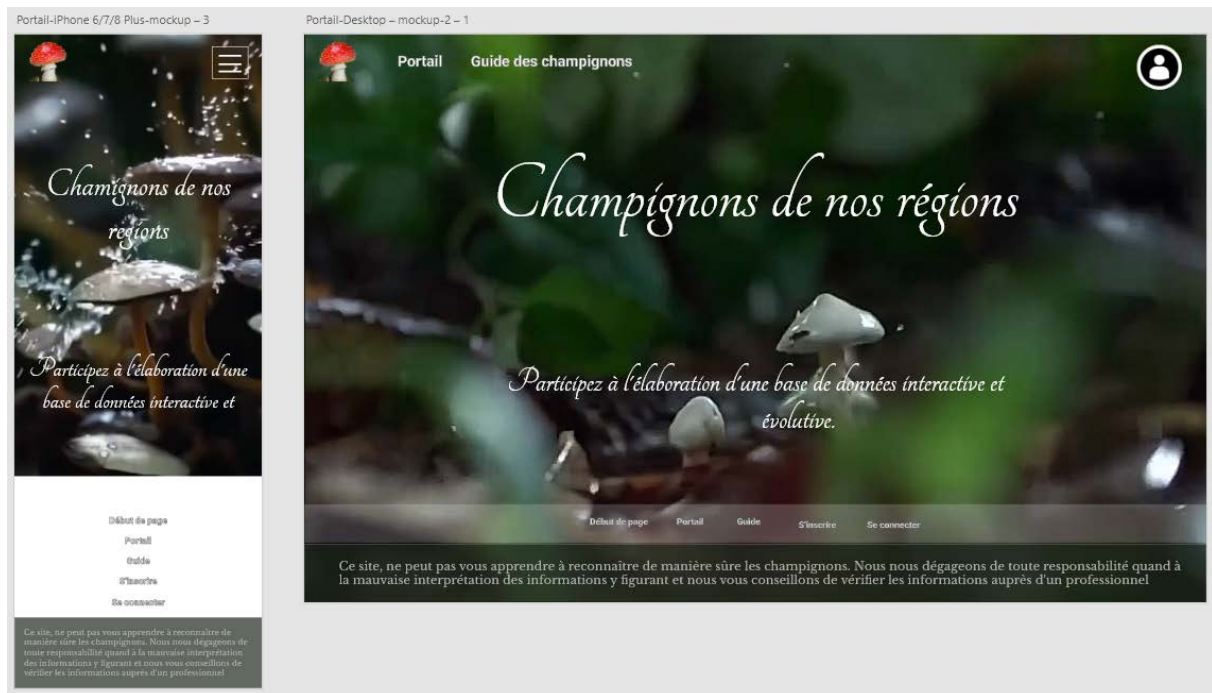
Je dois ajouter pour l'ensemble du site les traductions en anglais et en espagnol à partir d'un fichier Excel.

Les chaînes de caractères sont stockées dans une table de la base de données et accessibles via une interface d'administration.

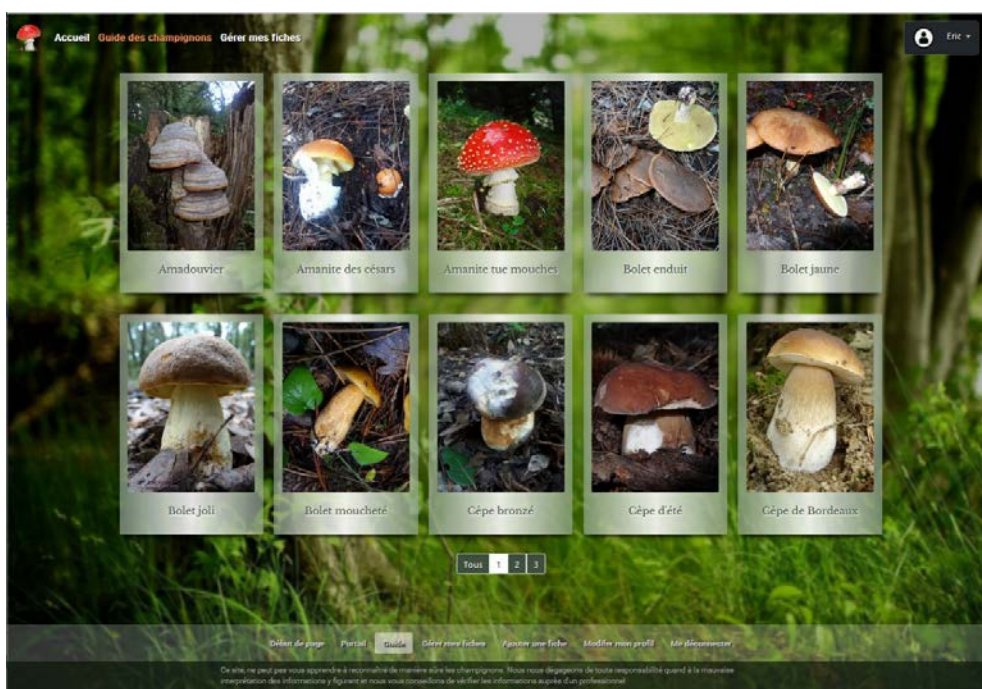
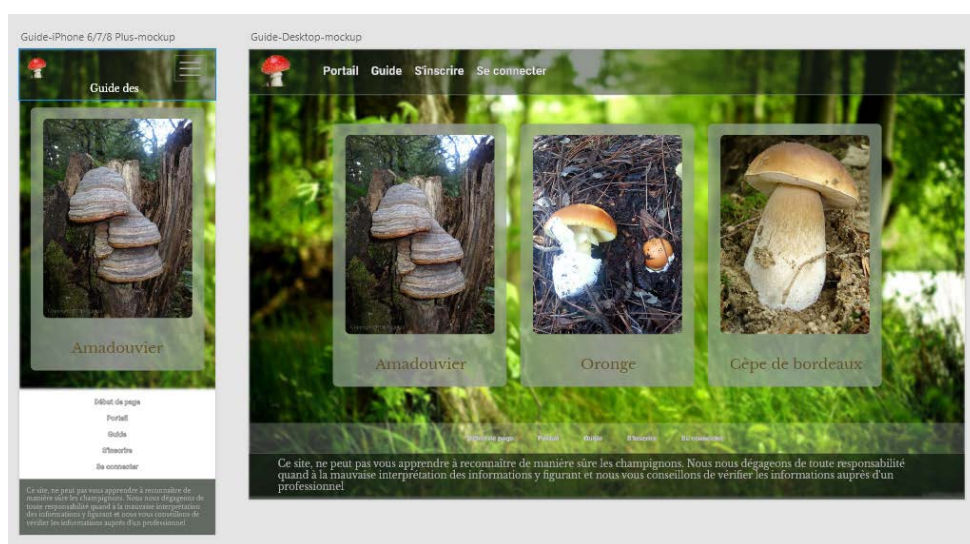
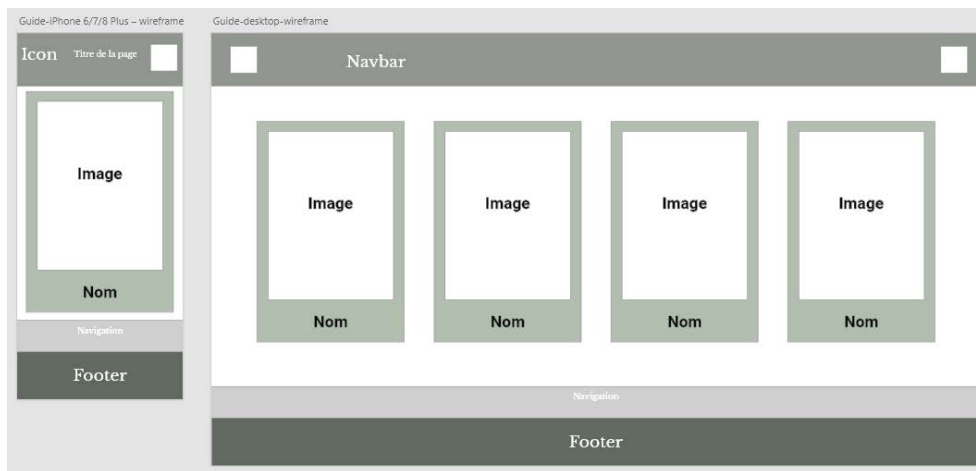
Si les chaînes sont en dur dans le code, je les remplace par des variables twigs et entre la traduction pour chaque langue.

5. ANNEXES

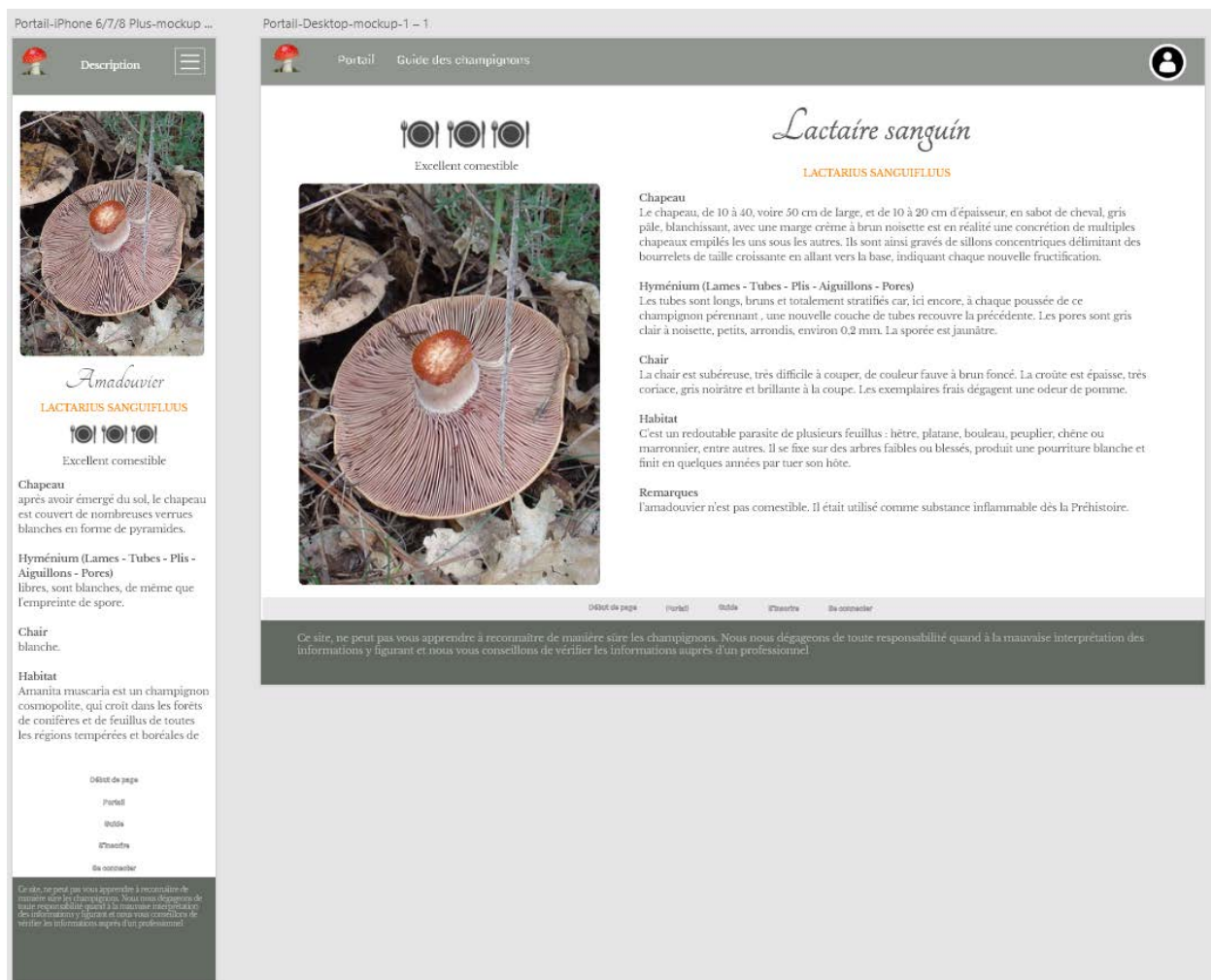
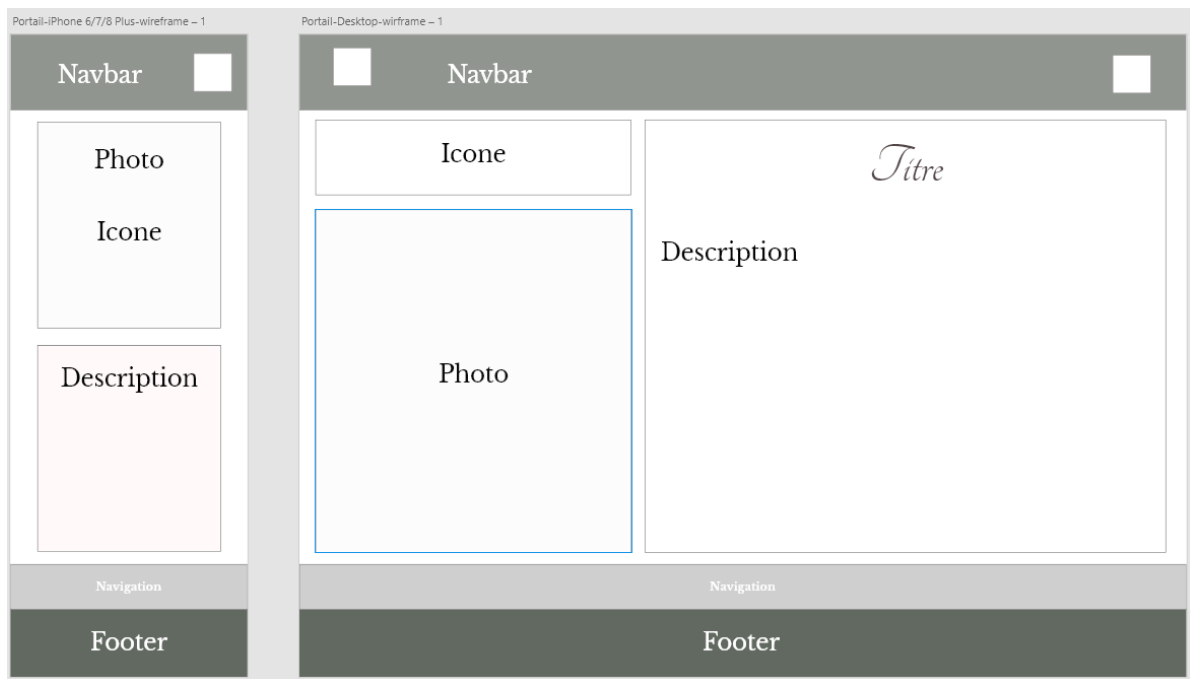
ANNEXE 1 - Page d'accueil



ANNEXE 2 - Guide des champignons




ANNEXE 3 - Fiches descriptives





ANNEXE 4 - Page membre

[Accueil](#) [Guide des champignons](#) [Gérer mes fiches](#)


 Eric ▾

Profil












Date d'inscription	Status	Pseudo	Nom	Prénom	Modifier mon profil	Supprimer mon compte
2020-04-02 08:51:30	Membre	Eric				

Liste des champignons

Voir toute les fiches ☐

Nouvelle fiche descriptif 

Lorsque vous créer une fiche descriptive celle-ci n'est visible qu'à partir de votre compte, après validation elle sera accessible depuis l'espace public.

Photos	Date de création	Nom commun	Auteur	Modifier	Supprimer
	2020-05-29 16:52:21	a	Eric		
	2020-03-21 00:00:00	Amadouvier	Eric		
	2020-03-21 00:00:00	Amanite des césars	Eric		
	2020-03-21 00:00:00	Amanite tue mouches	Eric		
	2020-03-21 00:00:00	Bolet enduit	Eric		

1 2 3 4 5 6

ANNEXE 5 - Ajouter une description.

Accueil Guide des champignons Gérer mes fiches

Eric

Ajouter

Nom commun Veuillez remplir ce champ !

Nom latin

Nom local

Attention a ne pas dépasser la capacité d'upload : 25 mo. (total du poids de toutes vos photos).
Les types d'images acceptées : jpg, jpeg, gif, png.

Photo Aucun fichier chargé.

Photo Aucun fichier chargé.

Photo Aucun fichier chargé.

Photo Aucun fichier chargé.

Photo Aucun fichier chargé.

Photo Aucun fichier chargé.

Descriptif

Comestible

Chapeau

Type de lames

☒ Néant ☐ Adnées ☐ Décurrentes ☐ Echancrées ☐ Emarginées ☐ Libres ☐ Sécédentes

ANNEXE 6

```
function uploadFiles(event) {
    event.preventDefault();
    changeStatus("Uploading...");

    var formData = new FormData();
    for (var i = 0, file;
        (file = droppedFiles[i]); i++) {
        formData.append(fileInput.name, file, file.name);
    }
    formData.append('candidature', titleCandidature.value);

    /* Envoie une requete au serveur */
    fetch(dropFileForm.action, {
        method: dropFileForm.method,
        body: formData
    })
    .then(res => res.json())
    .then(returnData => {
        console.log(returnData.response);
        change_modal_footer();
        fileLabelText.innerHTML = returnData.response;
    })
    .catch((error) => {
        console.log(error.message);
        change_modal_footer();
        fileLabelText.innerHTML = "Désolé, le serveur ne reponds pas.";
    });
}
```

ANNEXE 7

```
class UploadController extends ApplicationController
{
    public function index()
    {
        // Récupère les données du formulaire
        $datas = $this->request->getData();
        $refCandidature = $datas['candidature'];

        // Gestion d'upload du CV
        if($datas['files'][0]['error'] == 0)
        {
            $fileTmp = $datas['files'][0]['tmp_name'];
            $nameFile = $datas['files'][0]['name'];
            if(!file_exists('./uploadPDF/'))mkdir('./uploadPDF/');
            $fileCv = './uploadPDF/'.$nameFile;
            // Transfert de la piece jointe vers le dossier webroot/uploadPDF
            move_uploaded_file($fileTmp,$fileCv);
            // Récupère le chemin absolu de la piece jointe
            $pathAbsolute = realpath('uploadPDF/'.$nameFile);

            // Envoi du mail a l'administrateur
            $email = new Email('sendInBlue');
            $email->setFrom(['hello@ikosoft.com' => 'Ikosoft'])
                ->setTo('hello@ikosoft.com')
                ->setSubject("Réception d'une candidature")
                ->setAttachments($pathAbsolute)
                ->send("Bonjour <br> Vous avez reçu une nouvelle candidature pour le poste de :
                    ".$refCandidature);

            // Reponds au client (mail bien envoyé )
            $responseResult = json_encode(['response' => "Nous accusons réception de votre offre
            de collaboration et nous vous remercions de l'intérêt que vous portez à la société
            ikosoft.<br>Votre dossier sera traité dans les plus brefs délais.<br>Cordialement"]);
            // Supprime la piece jointe après envoie.
            unlink($fileCv);
        }
        else
        {
            // Reponds au client (echec d'envoi du mail)
            $responseResult = json_encode(['response' => "Une erreur c'est produite, impossible
            d'envoyer votre CV."]);
        }

        $this->response->withType('json');
        $this->response->body($responseResult);
        return $this->response;
    }
}
```

ANNEXE 8

```
// On recupere le slug ($path) on verifie s'il a déjà été chargé dans le cache
if (($pageCache = Cache::read($lang.$urlAsked)) === false) {

    // le cache est vide on effectue la requete
    $page = TableRegistry::get('Pages')->find()->where(['slug' => $urlAsked])->first();
    if (!$page) {
        $page = TableRegistry::get('Pages')->find()->select('slug')->where(['shortName' => 'index'])
        ->first();
        return $this->redirect('/'.$lang.'/'.$page->slug);
    }

    // Récupère le layout de la page
    $layout = $page->layout;
    $metaTitle = $page->nom;
    $metaDesc = $page->description;

    //detection de la page active
    $page = $this->request->params['pass'][0];

    /*****
    | PREPARE LES DONNEES AVANT DE LES TRANSMETTRE AU FONCTIONS TWIG
    | PUIS AJOUTE LES FONCTIONS TWIG PERSONNALISEE
    *****/
    $page = $this->loadPage($page,$lang);

    /*****
    | LE RENDER QUI REMPLACE LES VARIABLES TWIG PAR LEUR VALEURS
    *****/
    $myLayout = $this->renderPage($layout);
    // on crée le cache pour cette route
    $contentCached = ['metaTitle' => $metaTitle, 'metaDesc' => $metaDesc, 'myLayout' => $myLayout];
    Cache::write($lang.$urlAsked, $contentCached);
} else {
    // si le cache existe pour cette page on lit juste le contenu.
    $contentCached = Cache::read($lang.$urlAsked);
    $metaTitle = $contentCached['metaTitle'];
    $metaDesc = $contentCached['metaDesc'];
    $myLayout = $contentCached['myLayout'];
}
```

ANNEXE 9

HELLO !
FONCTIONNALITÉS
PRIX
CONTACT
CLUB MERLIN

BOOKER UNE DÉMO

ACTUALITÉS

MCB 2020:

venez nous retrouver!

Édition toute particulière puisqu'elle accueillera la coupe du monde de la Coiffure organisée par l'OMC (Organisation Mondiale de Coiffure) pour la 3e fois !

Pour cette occasion le MCB by Beauté Selection met en lumière 1300 talents et compétiteurs internationaux sur deux jours d'épreuves dédiés à la coiffure, aux soins et au maquillage.

Ikasoftware sera bien entendu présent cette année encore avec de nombreuses nouveautés et des surprises !

On en reparle vite !

LES INFOS UTILES

Quand ? Les 12-13-14 septembre 2020

Où ? Au parc des Expos, Porte de Versailles à Paris

N° de stand : À venir

Site du MCB : www.mcbbybeautyselection.com

#Stayhome

Nos mesures d'accompagnement Covid-19

Geste solidarité Covid-19

Les bons d'achats

Geste solidarité Covid-19

La Boutique

Tutoriel Merlin

Encouragez la solidarité de vos clients

Tutoriel Merlin

Rassembler ses chiffres et analyser la situation

Tutoriel Merlin

Modifier les horaires de votre établissement

Tutoriel Merlin

Créer des jours d'ouverture exceptionnelle

Consell

Les astuces pour bien gérer le home office

MCB 2019

Retour en image sur la dernière édition du MCB

Formation

Merlin offre ses logiciels aux écoles

Marketing

Vos plans marketing créés et exécutés par Ikasoftware

PLAN DU SITE

- Hello !
- Fonctionnalités
- Prix
- Club Merlin
- Contact
- Mentions légales
- Logiciel de caisse coiffure
- Logiciel de caisse institut de beauté
- Logiciel de caisse spa

Édité par la société IKASOFT

90 rue Frédéric Joliot - CS 30334
Aix-en-Provence Cedex 3
04 42 90 07 77

Jeux, conseils, cadeaux, rencontres...

Suivez-nous sur nos réseaux sociaux !

Français

Télécharger

TeamViewer

43

ANNEXE 10

[HELLO !](#)
[FONCTIONNALITÉS](#)
[PRIX](#)
[CONTACT](#)
[CLUB MERLIN](#)

[BOOKER UNE DÉMO](#)

CONSEIL

Les astuces pour bien gérer le home office

Nous sommes tous dans le même bateau et vivons ensemble au rythme du confinement. Pas question de se laisser aller, voici nos astuces pour tenir bon et même en tirer de bonnes habitudes.

1. Maintenir le rythme

Vous vous levez à 7h30 pour aller travailler ? Continuez ! Le rythme est la clé de l'équilibre chez nous adultes comme chez les tout-petits. Par contre, vous pouvez profiter des moments que vous passez dans les transports pour prendre du temps pour vous : un super petit-déjeuner, un masque pour vos cheveux, un peu de sport. À vous de choisir ! Si vous n'êtes pas en télétravail, appliquez le même principe et essayez de maintenir des heures fixes pour vos différentes activités : ne vous laissez pas embarquer par la saison complète de votre série préférée à 13h30, déprime assurée !

2. Renouer avec les bonnes choses

S'il y a bien une chose qui rime avec confinement c'est gourmand. La France entière, riche de sa culture gastronomique, s'est mise aux fourneaux. Les plus grands chefs se mobilisent pour nous proposer chaque jour de nouvelles recettes, pour tous les niveaux. Pourquoi ne pas s'y mettre ? Re-créez des moments en famille autour des repas... Plus cool que la lunch-box scotchée à l'écran de son ordinateur non ?

3. Prendre conscience du temps

Ne plus être emporté par la frénésie et le stress du quotidien peut laisser chez certains un grand sentiment de vide. Prenez conscience de ce temps, utilisez-le. Soyez à chaque chose que vous faites à 100%, ne bâcliez pas votre séchage de cheveux car vous avez peur d'être en retard, finissez le chapitre de votre livre, profitez des rayons du soleil, mais vraiment. Vous verrez vos journées auront un tout nouveau goût et seront encore plus remplies !

4. Garder le contact

La vie sociale est importante, elle est le socle de toute société. Nous avons la chance de vivre une époque ultra connectée qui nous permet de garder le lien avec nos proches. Visio call avec vos parents, vos amis, vos collègues de travail... Les enfants eux aussi en ont besoin. De nombreuses applications ont vu le jour pour faciliter cette démarche.

5. Être en forme

Nous avons assisté dès le début du confinement à de nombreux challenges sportifs sur les réseaux sociaux. Votre corps n'a pas forcément besoin de devenir votre objectif exclusif pendant cette période, prenez le temps de faire les choses, à votre rythme. Pourquoi ne pas suivre des cours de Yoga ou de gym douce 15 à 20 minutes par jour et vous y tenir, juste pour vous sentir bien et toujours sans pression.

PLAN DU SITE

[Hello !](#)
[Fonctionnalités](#)
[Dns](#)
[Club Merlin](#)
[Contact](#)
[Mentions légales](#)
[Logiciel de caisse](#)
[Logiciel de caisse](#)
[Logiciel de caisse](#)
[Logiciel de caisse](#)

Édité par la société IKOSOFT

90 rue Frédéric Joliot - CS 38034

Aix-en-Provence Cedex 3

04 42 90 01 11

Jeu, conseil, cadeaux, rencontres...

Suivez nous sur nos réseaux sociaux !

[Français](#)

[@ikosoft](#)