

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ LANZA
Nom d'usage
Prénom ▶ Eric
Adresse ▶ 23 rue Villeneuve
13001 Marseille

Titre professionnel visé

Développeur web et web mobile

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	5
▶ Intitulé de l'exemple n° 1 Construire la maquette d'une application météo.....	p.	5
▶ Intitulé de l'exemple n° 2 Utiliser le Framework de type adaptatif Bootstrap	p.	8
▶ Intitulé de l'exemple n° 3 Réalisation d'une Picture Box.....	p.	13
Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	19
▶ Intitulé de l'exemple n° 1 Créer une base de donnée de type relationnelle	a	19
▶ Intitulé de l'exemple n° 2 Développer les composants d'accès aux bases de données.....	p.	23
▶ Intitulé de l'exemple n° 3 Développement d'un service d'email transactionnel	p.	27
Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i>	p.	
Déclaration sur l'honneur	p.	
Documents illustrant la pratique professionnelle <i>(facultatif)</i>	p.	
Annexes <i>(Si le RC le prévoit)</i>	p.	

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

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

Exemple n°1 ► Construire la maquette d'une application météo

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. Le projet

Dans le cadre d'une mise en application sur le web dynamique dispensée par le centre de formation, il m'était demandé de développer une application web 2.0, permettant de se connecter à un service web (API), afin de récupérer diverses informations météorologiques.

Le cahier des charges prévoit :

- Une recherche par ville,
- D'afficher la date,
- L'icône du jour,
- La température minimum, moyenne et maximum
- Les prévisions pour les 4 prochains jours.

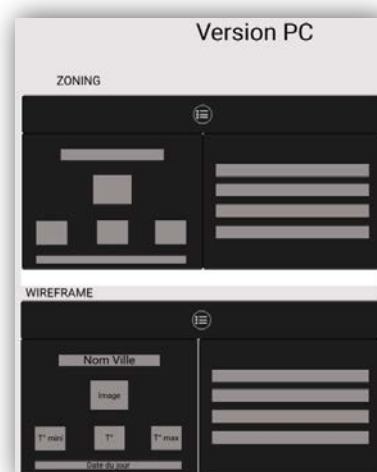
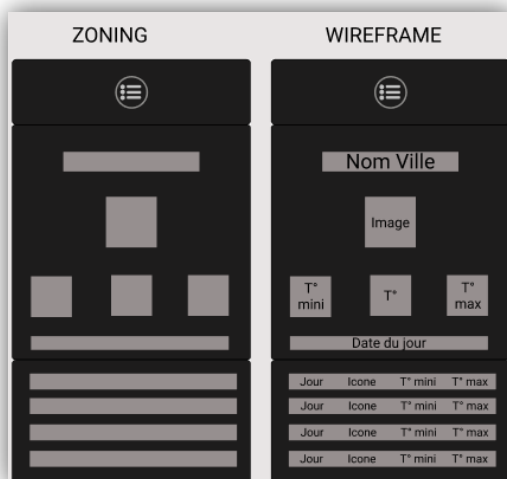
2. Etapes de réalisation de la maquette

La création de la maquette web peut être réalisée de façon manuelle en dessinant des schémas soit à partir de vignette représentant les divers blocs de l'application, soit avec des logiciels (Illustrator, Photoshop...) ou en utilisant des outils en ligne tels que AXURE, BALSAMIQ, FLAIRBUILDER, FIGMA...

C'est ce dernier qui m'a permis de mettre en œuvre les différentes vues de ma page web tant pour la partie fonctionnelle que graphique.

Dans un premier temps, j'ai conçu la **maquette fonctionnelle**, c'est une représentation basse fidélité de l'application.

Elle permet optimiser l'expérience utilisateur et peut être subdivisée en 3 étapes :



- J'ai d'abord utilisé le **zoning** pour l'organisation visuelle, afin de découper la page en trois parties, un bloc pour le header, c'est à partir d'ici que je permettrais la recherche pour une ville, un autre pour les informations météo du jour, puis un dernier conteneur qui affichera les prévisions météo des 4 prochains jours.
- En second lieu le **wireframing** m'a permis de me focaliser sur les aspects ergonomiques, en évacuant toutes les considérations d'ordre esthétique, mais apporte quelques détails supplémentaires, précisant l'organisation interne de certaines sections : emplacement des boutons, des titres et des contenus photos ou vidéos, etc...
- Vient ensuite la phase de **prototypage**, teste les interactions principales de l'application d'une manière similaire au produit final, afin de m'assurer qu'elle soit responsive et adaptable.

Dans un deuxième temps, j'ai conçu la **maquette graphique**, qui est la représentation haute-fidélité de l'application avant que celle-ci ne passe en production.

Le **style guide** se rapproche fortement de la charte graphique et recense tous les éléments utiles concernant le webdesign d'un site (typographie, couleur, apparence des boutons...).

- J'ai choisi une couleur de fond proche du noir pour contraster avec le contenu textuel de l'interface.
 - Le type de police de caractère « roboto » et une couleur blanche pour le texte et les marges.
- Ma dernière étape fut la réalisation du **mockup**, présentation finale de l'interface.



La maquette étant prête, je peux passer à la phase de développement.

DOSSIER PROFESSIONNEL (DP)

L'identification des différents blocs me permet d'être plus productif :

- Je peux ainsi définir la structure du document HTML grâce à la vue fonctionnelle.

Réaliser la mise en forme de la page en utilisant un langage de feuille de style CSS, préprocesseur SASS, LESS ou encore un Framework type Bootstrap.

2. Précisez les moyens utilisés :

Un PC sous Windows avec connexion internet.

Le détail du projet.

Un outil de maquettage en ligne (Figma).

Photoshop pour dessiner des contrôles bouton.

3. Avec qui avez-vous travaillé ?

Il s'agit d'un exercice individuel.

4. Contexte

Nom de l'entreprise, organisme ou association ► *AFPA*

Chantier, atelier, service ► Utiliser un service web

Période d'exercice ► Du : *19/12/2019* au : *09/01/2020*

5. Informations complémentaires (facultatif)

Activité-type 1

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

Exemple n° 2 ► Utiliser le Framework de type adaptatif Bootstrap

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. Le projet

Je dois développer une application single page permettant d'éditer une liste de tâches stockées dans une base de données relationnelles.

Je décompose cette application en 3 pages :

- La page principale liste l'ensemble des tâches
- La deuxième page permet d'en afficher le détail
- La dernière page offre la possibilité de modifier

2. Maquettage

Dans un premier temps je crée la maquette, afin de construire une mise en page adaptée à mon objectif et contenu.

Travaillant seul sur le projet, je commence par dessiner sur feuilles libres les différentes parties qui constituent chacune des interfaces web.

Je commence par la vue mobile pour terminer par la vue desktop.

3. Structure d'un document HTML

Je démarre mon éditeur de code et ajoute ma première page, j'inclus la structure de base d'une feuille html le **Doctype**, la balise **html** composée d'une section **head** et **body**.

4. Mise en forme des documents HTML

4.1. CSS

Pour mettre en forme mes pages web, je peux utiliser le concept de feuille de style CSS (*Cascading Style Sheets*), langage déclaratif.

Une feuille de style est structurée en règles CSS basées sur le même modèle.

Elle se décompose en un sélecteur indiquant au navigateur les éléments HTML qui doivent être mis en forme, appelées bloc de déclaration se trouvant entre accolades.

Ceux-ci renferment des propriétés sous forme clé valeurs, permettant de modifier l'aspect des éléments HTML.

4.2. BOOTSTRAP Framework CSS

J'ai choisi d'utiliser les classes de la bibliothèque Bootstrap car elle propose une alternative au développement CSS pur.

Son principal avantage réside dans l'amélioration de la productivité de développement.

En effet, il corrige les problèmes de compatibilité entre navigateurs (*cross-browser*) offrant une présentation quasi identique, normalise la présentation en proposant un ensemble homogène de styles, propose un système de grille pour faciliter le positionnement des éléments, est responsive et adaptable autrement dit, les blocs de contenus (textes et images) se redimensionnent et se réorganisent en fonction de l'espace disponible sur l'écran.

Par ailleurs, il propose des éléments stylisés et des composants qui nécessitent l'utilisation de JavaScript pour fonctionner.

4.3. L'installation du framework

Tout d'abord, je dois installer sa bibliothèque, plusieurs méthodes sont disponibles :

- Télécharger une version des fichiers nécessaire au fonctionnement de Bootstrap (bootstrap.css, jquery.js, popper.js, bootstrap.js),
Pour le rendre fonctionnel je dois l'ajouter à mon projet, dans ce cas j'insère une balise **link** dans l'entête de ma page section **head** pointant vers le fichier CSS de Bootstrap, puis j'ajoute dans une balise script les 3 liens vers les fichiers JavaScript.
- Avec le CDN (*content delivery network*), qui est constitué d'ordinateurs reliés en réseau à travers Internet et qui coopèrent afin de mettre à disposition du contenu ou des données à des utilisateurs, j'utiliserai la même méthode déclarative citée précédemment.
- L'ajouter via le gestionnaire de paquets officiel de Node.js **npm** avec cette commande **npm install bootstrap**, je bénéficie en plus de la version portable en **Sass**, *préprocesseur CSS* qui permet de personnaliser Bootstrap mais aussi d'utiliser des fonctionnalités comme des variables, l'imbrication de règles CSS, des fonctions, l'import de fichiers, les mixins... simplifiant le développement et la maintenabilité.
Cependant, le code nécessite une compilation des fichiers **scss** en **css** car les navigateurs n'intègrent pas ce langage.

C'est cette dernière solution que je choisis. Je crée un dossier SASS, j'ajoute un fichier **main.scss**, j'importe le fichier SASS de Bootstrap `@import "node_modules/bootstrap/scss/bootstrap.scss";`.
Suivi des règles personnalisées qui viennent en complément des fonctionnalités de Bootstrap, sans oublier d'ajouter dans une balise script les 2 liens vers les fichiers JavaScript.

```
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
  integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous">
</script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
  integrity="sha384-Q6E9RHvIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous">
</script>
```

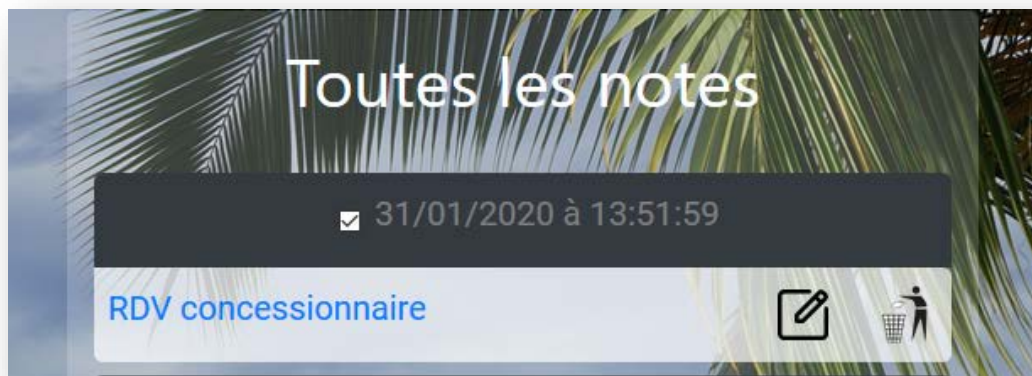
Pour la compilation du fichier SASS je saisi dans le fichier **package.json** (fichier répertoriant les dépendances) le nom de la source et la destination du fichier à compiler.

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

4.4. Le principe des grilles

Pour constituer mes pages, j'utilise le principe des grilles de Bootstrap, un élément conteneur (container) englobant des lignes (row) elles-mêmes subdivisées jusqu'à 12 colonnes (col).

Voici un exemple pour la réalisation d'une tâche (une vue mobile).



Pour créer ce design j'ai placé le container dans la page d'accueil, le code html est chargé dynamiquement en JavaScript.

```
<div class="container todos">  
  <!-- insertion du code js -->  
</div>
```

Afin que ma page s'adapte au différents types de support numériques j'utilise les classes colonnes, 5 tailles sont proposées du mobile aux écrans ultra larges.

- col- : Extra Small
- col-sm : Small
- col-md : Medium
- col-lg : Large
- col-xl : Extra Large

```
<div class="row listTodo m-2">  
  <div class="col-12">  
    <div class="row">  
      <div class="col-12 col-md-6 col-lg-4 d-flex flex-nowrap  
        justify-content-center align-items-center bg-dark p-1 listTodo_b1">  
        <input type="checkbox" class="listTodo_b1_done ml-3" data-id="{todo.id}"  
          "{stateCheckBox.etat}">  
        <h5 class="p-2 listTodo_b1_Date" style="color:{stateCheckBox.color};">${  
          {new Date(todo.createdAt).toLocaleString()}}</h5>  
      </div>  
    </div>  
  </div>
```

DOSSIER PROFESSIONNEL (DP)

Ci-dessus dans cette portion de code on définit la taille d'une colonne « col-12 col-md-6 col-lg-4 », celle-ci s'étire sur toute la largeur de son conteneur sur un support mobile puis se réduit de moitié sur une tablette et enfin elle occupera 1/3 de la taille d'un écran large.

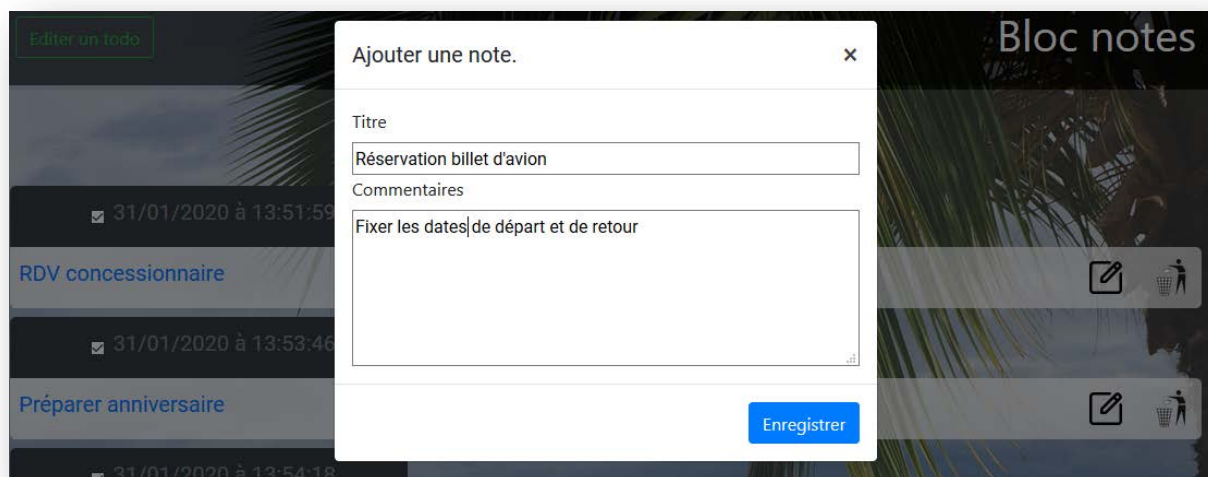
4.5. Utilisation du composant modal

Afin d'ajouter une nouvelle tâche je me suis servi d'un plug-in ou composant le modal, il permet d'afficher une boîte de dialogue il utilise HTML, CSS, JavaScript.

Je copie un des exemples fournis sur le site et modifie son contenu pour l'adapter à mes besoins en ajoutant dans le corps du modal un formulaire.

Le bouton avec l'attribut **data-target** pointant sur l'identifiant du modal permet le déclenchement de l'ouverture de la fenêtre.

La fermeture s'effectue via un bouton contenant l'attribut **data-dismiss**.



2. Précisez les moyens utilisés :

Un PC sous Windows avec connexion internet.

Pour le développement : un éditeur de code VsCode.

Utilisation des langages : HTML/CSS/BOOTSTRAP.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Développer les composants d'accès aux données.

Période d'exercice ► Du : 29/01/2020 au : 10/02/2020

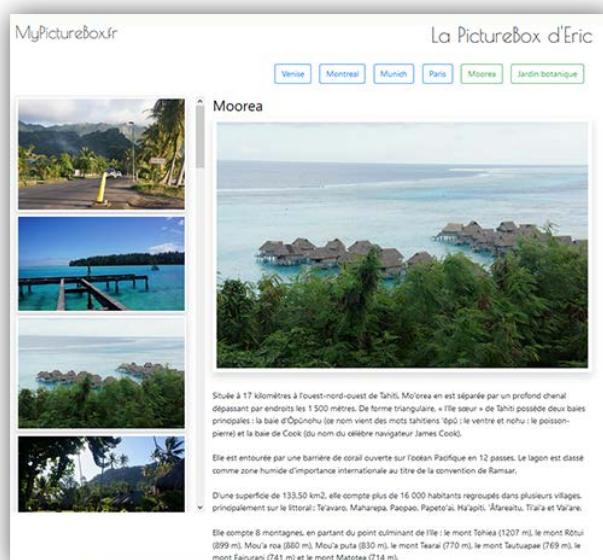
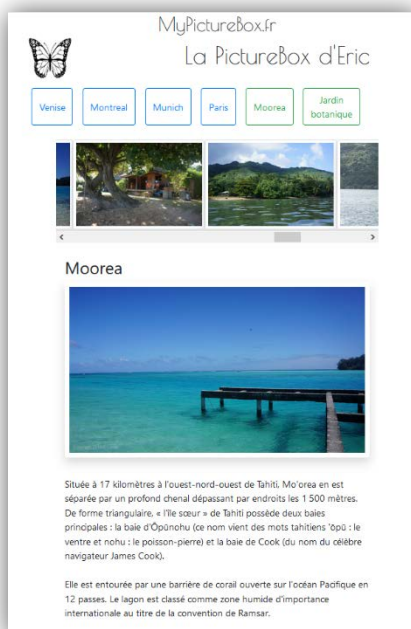
5. Informations complémentaires (facultatif)

Activité-type 1

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

Exemple n° 3 ► Réalisation d'une Picture Box

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :



1. Le projet

La réalisation de ce projet s'appuie sur un cahier des charges et plusieurs vues fonctionnelles de type Wireframe.

L'interface doit être décomposée en 3 parties :

- Un accès aux différentes catégories
- Les vignettes regroupant les miniatures de chaque catégorie
- Un bloc affichant les détails de chaque vignette (photo, titre, commentaires)

Dans ce projet j'aborde les méthodes que j'ai mises en place dans le cadre d'une application single page, le code html étant généré dynamiquement.

Pour démarrer mon projet j'utilise VsCode un éditeur de code multiplateforme, open source et gratuit, supportant une dizaine de langages.

2. Ecriture du code html

Je commence par créer un fichier HTML en m'assurant qu'il respectera les recommandations du w3C, organisme de standardisation chargé de promouvoir la compatibilité des technologies du World Wide Web.

Je décide d'utiliser le principe des grilles de Bootstrap pour contenir mes blocs éléments et obtenir une page responsive et adaptable.

Au début du fichier HTML j'indique le Doctype instruction qui décrit le type de document.

La balise HTML qui contient les balises enfants est divisée en deux parties :

L'entête : <head> partie invisible de la page.

- Je précise le format de codage des caractères utilisés Utf-8
- Un lien vers le CDN de Bootstrap, un Framework CSS
- Le lien vers ma feuille de style
- Le titre de ma page.

Le corps de page : <body> partie visible de la page.

- J'ajoute les scripts nécessaires pour certaines fonctionnalités de Bootstrap.

```
index.html X # style.css
index.html > html > body
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
   integrity="sha384-Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
7   <link rel="stylesheet" href="style.css">
8   <title>PictureBox</title>
9 </head>
10 <body>
11
12
13 <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849b1E2
+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1lyYfoR5SJoZ+n" crossorigin="anonymous"></script>
14 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft
+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
15 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj003uMBJnJUUD4Ih7YwaYd1iqfktj0Uod8GCExl30g8ifwB6" crossorigin="anonymous"></script>
16 </body>
17 </html>
```

Je crée un ensemble de balises sémantiques imbriquées contenant chacune des classes Bootstrap, un conteneur principal (container), une ligne (row), de colonnes (col) afin d'agencer le contenu de ma page.

La ligne contient 4 colonnes que je définis de type Flex :

- Ma première colonne est une balise sémantique <header> pour contenir le titre principal et un sous-titre
- La seconde, regroupe l'ensemble des boutons permettant la navigation entre les différentes pages
A chaque bouton j'ajoute un attribut « identifiant » avec pour valeur le nom de la ville
Un évènement « onclick » pointant sur une fonction de mon code JavaScript.
- La troisième, dispense les photos sous forme de vignettes, avec une classe « overflow ».
Le contenu est généré à partir d'un fichier JavaScript.
- La dernière est un bloc dans lequel apparaît le titre, la photo et les commentaires associés.

```
<!-- BOX CATALOGUE -->
<aside class="col-12 mt-4 text-center d-flex justify-content-end flex-row">
  <button identifiant="venise" type="button" class="btn btn-outline-primary mr-3 btn-md"
    onclick="infoCatalogue();">Venise</button>
  <button identifiant="montreal" type="button" class="btn btn-outline-primary mr-3 btn-md"
    onclick="infoCatalogue();">Montreal</button>
  <button identifiant="munich" type="button" class="btn btn-outline-primary mr-3 btn-md"
    onclick="infoCatalogue();">Munich</button>
  <button identifiant="paris" type="button" class="btn btn-outline-primary mr-3 btn-md"
    onclick="infoCatalogue();">Paris</button>
  <button identifiant="moorea" type="button" class="btn btn-outline-success mr-3 btn-md"
    onclick="infoCatalogue();">Moorea</button>
  <button identifiant="harrisonSmith" type="button" class="btn btn-outline-success mr-3 btn-md"
    onclick="infoCatalogue();">Jardin botanique</button>
</aside>

<!-- BOX VIGNETTE -->
<aside id="containerImg" class="col-12 col-md-4 mt-4 d-flex flex-row flex-md-column overflow-auto">
  <!-- Ajout code js -->
</aside>

<!-- BOX TITRE, PHOTO, COMMENTAIRES -->
<aside id="boxZoom" class="mt-4 col-12 col-md-8">
  <!-- Ajout code js -->
</aside>
```

3. Code CSS

Un fichier CSS pour définir quelques règles :

- Le type de police,
- La taille des vignettes
- La taille maximum de la photo
- Et une animation au survol des vignettes

4. Code JAVASCRIPT

J'ajoute un fichier JavaScript afin de manipuler le DOM interface de programmation qui permet de manipuler les éléments d'un document HTML.

Je définis 2 constantes, pour chaque bloc parent : vignettes et détail.

```
// Stocke dans des constantes, les ID de balise,
// on pourra modifier, créer, supprimer des balises au niveau du fichier HTML via le DOM.
const $vignettesContainer = document.getElementById("containerImg");
const $boxZoom = document.getElementById("boxZoom");
```

Pour stocker toutes les informations : nom de ville, titre, commentaires et photos je crée un tableau d'objet avec des clefs identifiables au travers de mon code.

```
// mini base de donnée sous forme d'un tableau à 2 dimensions.
var myDataBase = [{
  'identifiant': 'paris',
  'titre': 'Paris',
  'commentaires': "<blockquote>Paris, capitale de la France, est une grande ville européenne et un centre mondial de l'art, de la mode, de la gastronomie et de la culture. Son paysage urbain du XIXe siècle est traversé par de larges boulevards et la Seine. Outre les monuments comme la tour Eiffel et la cathédrale gothique Notre-Dame du XIIe siècle, la ville est réputée pour ses cafés et ses boutiques de luxe bordant la rue du Faubourg-Saint-Honoré.</blockquote>",
  'dossier_photo': 'img/paris/',
  'files': ['1.jpg', '2.jpg', '3.jpg', '4.jpg', '5.jpg', '6.jpg', '7.jpg', '8.jpg', '9.jpg', '10.jpg', '11.jpg', '12.jpg', '13.jpg', '14.jpg'],
},
```


Au chargement de la page web, JavaScript exécute une instruction pointant sur une fonction, celle-ci reçoit en paramètre le nom de la ville et affiche les données associées.

Affichage d'une nouvelle page lors d'un clic sur le bouton ville

Lors du clic sur le bouton ville la propriété **on click** appelle la fonction « info Catalogue ».

```
<button identifiant="venise" type="button" class="btn btn-outline-primary mr-3 btn-md"
  onclick="infoCatalogue();" >Venise</button>
```

« Info Catalogue » écoute les évènements survenus sur le document, récupère l'élément puis j'extrais la valeur de l'attribut « identifiant » que j'envoie en paramètre vers la fonction « setAlbum ».

```
// EVENEMENT CLICK SUR UN BOUTON DU CATALOGUE
function infoCatalogue() {
  // Récupère l'ensemble des attributs de la balise pour les stocker dans une variable objet.
  var ProprietesBalise = event.target;
  // Extraît l'attribut identifiant et donc sa valeur (bouton catalogue),
  // pour la comparer à celle de la base de donnée.
  var albumName = ProprietesBalise.getAttribute("identifiant");
  setAlbum(albumName);
}

function setAlbum(albumName) {
  var album = getAlbum(albumName);

  if (album != null) {
    // CREER LES VIGNETTES
    const newHTML = createVignettesNewHTML(album);
    updateVignettesContainerHTML(newHTML);

    // CREER LES BALISES DANS LE CONTENEUR (BOX TITRE, PHOTO, COMMENTAIRES)
    setImgInternal(album.dossier_photo + album.files[0], album.titre, album.commentaires);
  }
}
```

« setAlbum » appelle 4 fonctions :

- La fonction « getAlbum » ci-dessous va parcourir le tableau et comparer le nom de la ville passée en paramètre avec les clefs stockés dans le tableau, elle retourne un objet contenant les données associées à la recherche ou nulle si les valeurs ne correspondent pas.

```
// Récupère la valeur pour la comparer a celle du tableau definit en haut de script.
function getAlbum(attributBalise) {
  // Parcourt le tableau.
  for (var i = 0; i < myDataBase.length; i++) {
    // compare l'identifiant balise avec l'identifiant base de donnée.
    if (myDataBase[i].identifiant == attributBalise) {
      // Bingo l'identifiant balise est = à l'identifiant base de donnée.
      return myDataBase[i];
    }
  }
  return null;
}
```

Maintenant que j'ai récupéré l'ensemble des informations appartenant à la ville je dois générer le code html

dans leur bloc respectif.

- J'actualise l'élément parent pour l'affichage des vignettes avec la seconde fonction « createVignettesNewHTML » qui parcourt chaque index du tableau images, appelle la sous fonction « createVignetteHTML » pour générer le code HTML.

```
/** *****
 *
 *          CREER LES VIGNETTES
 * ***** */
function createVignettesNewHTML(album) {
  // Initialise une nouvelle variable.
  let newHTML = "";
  // Parcourt le sous-tableau contenant les noms de fichiers.
  for (const fileName of album.files) {
    const idx = myDataBase.indexOf(album);
    // Appel la fonction qui genere chaque balise html.
    newHTML += createVignetteHTML(album.dossier_photo, fileName, idx);
  }
  return newHTML;
}

// Génère une balise html
function createVignetteHTML([dossierPhoto, fileName, albumIndex = -1]) {
  //methode n°2
  return ``;
}
```

- J'ajoute l'intégralité des vignettes dans la balise du document HTML.

```
// mise à jour de la balise.
function updateVignettesContainerHTML(html) {
  $vignettesContainer.innerHTML = html;
}
```

- Pour afficher le reste des informations, le titre, l'image et les commentaires je procède de la même manière.

```
// GENERE LES BALISES DANS LE CONTENEUR PARENT.
function setImgInternal(imagePath, titre, commentaires) {
  if (imagePath != null) {
    let newHTML = "";
    // AJOUTE LE TITRE
    newHTML = `<h3>${titre}</h3>`;
    // AFFICHE LA PHOTO
    newHTML += `<div class="d-block w-100 d-flex flex-row flex-wrap justify-content-center"></div>`;
    // AJOUTE LES COMMENTAIRES
    newHTML += `<p>${commentaires}</p>`;
    // Cree les balises
    $boxZoom.innerHTML = newHTML;
  }
}
```

DOSSIER PROFESSIONNEL (DP)

L'interface est fonctionnelle, j'ai ajouté la possibilité d'afficher l'image en plein écran dans un modal.

2. Précisez les moyens utilisés :

Dans cette démonstration j'ai utilisé un éditeur de texte vsCode sous windows.
Pour le développement : HTML, CSS, BOOTSTRAP, JAVASCRIPT.
Effectué des recherches sur différents site web : w3schoom.com, MDN web docs de Mozilla.

3. Avec qui avez-vous travaillé ?

Cet exercice était proposé dans le cadre de la mise en pratique de technologies Front-end web, je l'ai réalisé seul.

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Développer une interface utilisateur web dynamique

Période d'exercice ► Du : 15/10/2019 au : 05/11/2019

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 1 ► Créer une base de donnée de type relationnelle

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. Définition du projet

Développer une application web médical, de gestion de patients et permettre la prise de rendez-vous. Pour stocker ces informations, je commence par créer une base de données puis je développerai l'application afin de pouvoir ajouter, modifier ou supprimer un profil pour chaque patient pour pouvoir gérer ces rendez-vous.

Je choisis de travailler avec une base de données de type relationnelle. Les systèmes de gestion de base de données ou SGBD tel que MySQL offre cet avantage.

Pour réaliser le schéma de ma base de données, j'utilise **Jmerise** un logiciel développé en JAVA, dédié à la modélisation des modèles conceptuels de données (MCD), il permet la généralisation et la spécialisation des entités, la création des relations et des cardinalités ainsi que la généralisation des modèles logiques de données (MLD) et des script SQL.

Pour commencer je télécharge et installe l'environnement d'exécution de JAVA (JRE), puis j'effectue la même démarche avec Jmerise.

2. La mise en place de règles de gestion

Je recueille les besoins des futurs utilisateurs de mon application. Et à partir de ces besoins, je suis en mesure d'établir les règles de gestion des données à conserver.

3. Etablir le dictionnaire des données

Ce recueil d'informations me permet d'établir un dictionnaire de données, c'est un document qui regroupe toutes les données que j'aurai à conserver dans ma base (et qui figureront donc dans le MCD). Pour chaque donnée, il indique :

- Le **code mnémonique** : il s'agit d'un libellé désignant une donnée.
- La **désignation** : il s'agit d'une mention décrivant la donnée.
- Le **type de donnée** :
 - **Alphabétique** : lorsque la donnée est uniquement composée de caractères alphabétiques.
 - **Numérique** : lorsque la donnée est composée uniquement de nombres (entiers ou réels)
 - **Date** : lorsque la donnée est une date (au format AAAA-MM-JJ)
 - **Booléen** : Vrai ou Faux (TiniINT)
- La **taille** : elle s'exprime en nombre de caractères ou de chiffres.

DOSSIER PROFESSIONNEL (DP)

Dictionnaire de données

Filtre Attribut composé ☒ Utilisation ☒

Num	Nom	Code	type	taille	decimal	Utilisé
1	idpatients	IDPATIENTS	Auto_increment			<input type="checkbox"/>
2	nompatsients	NOMPATIENTS	Varchar	100		<input type="checkbox"/>
3	prenompatients	PRENOMPATIENTS	Varchar	10		<input type="checkbox"/>
4	anniversairepatients	ANNIVERSAIREPATI...	Date			<input type="checkbox"/>
5	telephonepatients	TELEPHONEPATIENTS	Varchar	100		<input type="checkbox"/>
6	mailpatients	MAILPATIENTS	Varchar	255		<input type="checkbox"/>
7	idrdv	IDRDV	Auto_increment			<input type="checkbox"/>
8	daterdv	DATERDV	Date			<input type="checkbox"/>

Importer Exporter Attribut utilisé par ... Supprimer Att. non utilisés

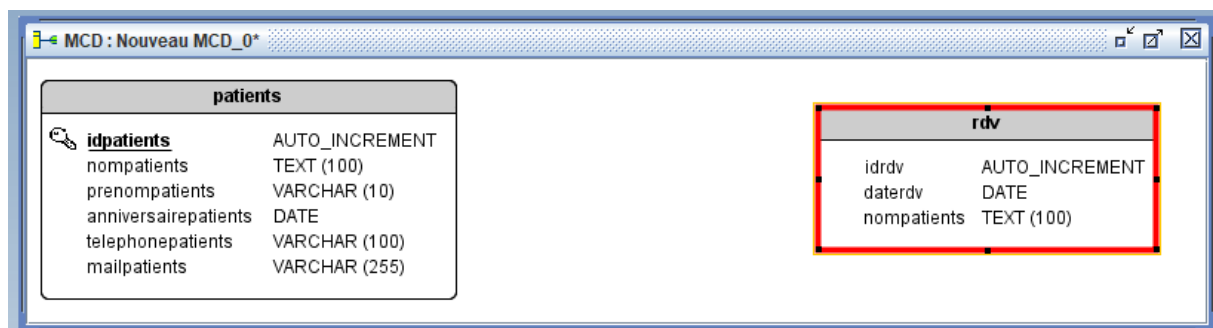
☐ Vérifier l'unicité des codes des attributs ☒ Vérifier les attributs

4. Le Modèle Conceptuel de Donnée (MCD)

J'élabore le Modèle Conceptuel de Donnée (MCD) basé sur deux notions principales : les **entités** et les **associations**, d'où sa seconde appellation : le **schéma Entité/Association**.

4.1. Les entités

A l'aide du dictionnaire de donnée je regroupe les informations à traiter en un ensemble cohérent.

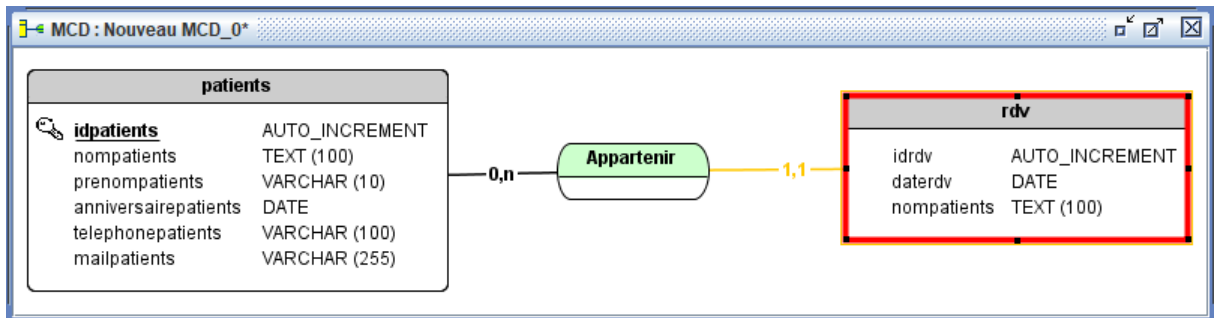


4.2. Les associations

J'établis les relations entre les 2 entités, et définit leurs **cardinalités**, un couple de valeurs *minimum*, *maximum*.

Dans l'exemple ci-dessous on déclare qu'un patient peut avoir de **0 à n** rendez-vous, et qu'un rendez-vous n'appartient qu'à **un et un** seul patient.

DOSSIER PROFESSIONNEL (DP)

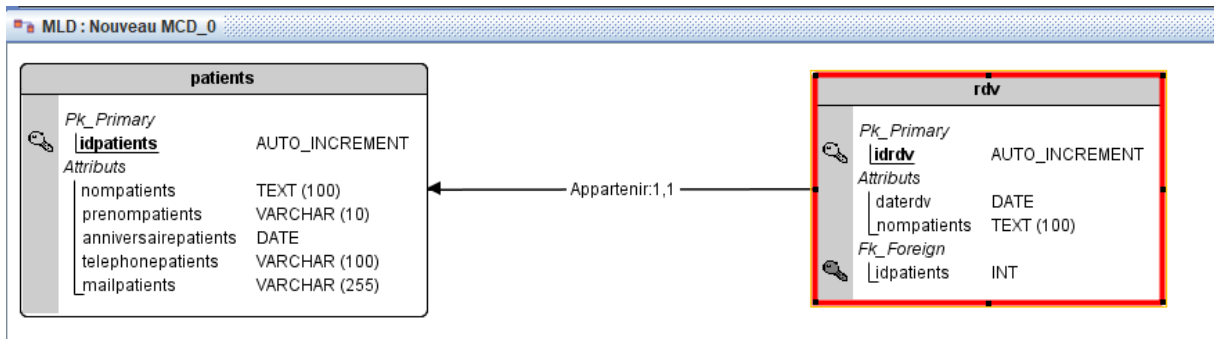


4.3. Modèle physique des données

Avec la validation du modèle conceptuel de données, l'utilitaire Jmerise génère le Model Physique des données.

Les **entités** deviennent des **tables**, les identifiants deviennent des **clés primaires**.

Les **associations** dépendent de la **cardinalité** avec trois scénarios possibles, création de clés étrangères et dans le cas d'une cardinalité dite many to many (0,n | 0,n) une table liaison est créé contenant autant de clés étrangères faisant référence aux clés primaires des tables liées.



4.4. Auto-génération du script de base de donnée

Je dispose à présent du script SQL (langage permettant de communiquer avec une base de données).

WAMP fournit un utilitaire : PhpMyAdmin est un outil d'administration de bases de données. Cet outil écrit en PHP facilite l'administration de MySQL sur les serveurs LAMP et WAMP. Importer, exporter, composer des requêtes complexes, supprimer et créer des tables, effectuer des recherches complexes ne sont que quelques-unes des possibilités offertes par phpMyAdmin.

Je crée une base de données qui respecte le standard d'encodage UTF-8, j'importe le script.

Ma base de données est prête à être utilisée, je peux passer à la phase de développement.

2. Précisez les moyens utilisés :

J'ai utilisé différents logiciels : JMerise et Photoshop.

Pour l'intégration du script phpMyAdmin.

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

Cette exercice a été réalisé individuellement.

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Développer les composant d'accès aux données.

Période d'exercice ► Du : 10/02/2020 au : 24/02/2020

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 2 ► Développer les composants d'accès aux données

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. L'objectif

Dans le cadre d'un exercice, je devais développer les composants d'accès aux données de type relationnelles.

Cette application web doit conserver des profils de patients et permettre la gestion de ces rendez-vous en ligne.

Les informations clients sont extraites de formulaires puis stockées dans une base de données pour garantir la pérennité de ces informations.

2. Les étapes

- Après avoir défini l'objectif, je dessine plusieurs maquettes sur papier.
- Développer la partie front end, chaque page disposera d'une barre de navigation pour accéder aux différentes rubriques de mon application.
- Développer la partie back end en PHP.

3. Développement des composants

J'utilise le langage PHP qui est un préprocesseur HTML, il fournit, en outre, une interface (PDO), ensemble de classes pour communiquer avec un système de gestion de base de données (SGBD).

3.1. Les SGBD

Ce sont des logiciels servant à stocker, à manipuler à gérer et à partager des informations dans une base de données. Ils comportent des mécanismes pour assurer la cohérence des informations, et éviter des pertes d'informations dues à des pannes, assurent la confidentialité et permettent son utilisation par d'autres logiciels.

J'ai la possibilité d'utiliser plusieurs SGBD pour la gestion de base de donnée relationnelles : **MySQL**, Microsoft Access, SQLite. J'ai choisi de me servir de MySQL installé avec WAMP.

Le moteur de base de données est le cœur du SGBD (*MyISAM*, *InnoDB*), mon choix se porte sur **innodb** car je dispose de 2 table liées *par des clés étrangères*.

3.2. SQL

C'est un langage déclaratif normalisé servant à exploiter des bases de données relationnelles.

Il permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles, est reconnu par la grande majorité des SGBD.

3.3. PDO (PHP Data Objects)

C'est une extension PHP qui définit une interface d'accès à une base de données. Son principal avantage est qu'il permet une abstraction pour l'accès aux données. C'est-à-dire que les fonctions pour exécuter des requêtes et pour récupérer des données sont les mêmes, quelque-soit le serveur SQL utilisé (MySQL, PostgreSQL, ...).

Je m'assure que cette extension PHP est activée, via WAMP SERVER.

4. Exemple de code pour l'ajout d'un patient

Ma première fonction PHP doit ajouter un nouveau patient dans la base de données.

```
// Ajoute un patient
function add_newPatient($nom,$prenom,$naissance,$tel,$email)
{
    $this->sqlCommand->addNewPatient($nom,$prenom,$naissance,$tel,$email);
    header('Location: index.php?routing=liste-des-patients');
    exit;
}
```

4.1. Connection à la base de données

J'ai regroupé dans une classe toute les méthodes qui permettent de communiquer avec la base de données. Je dois donc instancier cette classe.

```
// CONNEXION A LA BASE DE DONNEE
try {
    $sqlCommand = new SQLComponents();
    $connect = $sqlCommand->connectDataBase('hopital2n');
} catch (Exception $e) {
    die('Erreur SQL : ' . $e->getMessage());
}
```

4.2. Manipulation des données (CRUD)

Le CRUD est un acronyme pour désigner les quatre opérations de base pour la persistance des données. Chaque composante de l'acronyme peut être associée à un type de requête SQL, dont voici la liste :

Operations	CRUD	SQL
Créer	Create	INSERT
Lire	Read	SELECT
Editer	Update	UPDATE
Supprimer	Delete	DELETE

4.3. Notions de sécurités (failles ou injection SQL).

Pour me prémunir des failles de sécurité, j'utilise une requête préparée. En effet, l'injection dans une requête en cours, d'un morceau de requête non prévu par le système, peut compromettre la sécurité et l'intégrité d'un enregistrement d'une table et pourrait entraîner la suppression de la base de données.

```
// Ajoute un nouveau patient
public function addNewPatient ($lastname,$firstname, $birthdate,$phone,$mail){
    $sql = 'INSERT INTO patients (lastname,firstname,birthdate,phone,mail)
    VALUES (:lastname, :firstname,:birthdate, :phone,:mail)';
    $query = $this->_dataBase->prepare( $sql );
    $query->bindParam(':lastname', $lastname);
    $query->bindParam(':firstname', $firstname);
    $query->bindParam(':birthdate',$birthdate);
    $query->bindParam(':phone',$phone);
    $query->bindParam(':mail',$mail);
    $query->execute();
    $query->closeCursor();
}
```

On retrouve dans cette fonction la notion de CRUD dans la requête SQL avec l'opération INSERT, j'utilise des marqueurs nommés pour éviter les injection SQL puis j'appelle la méthode bindParam pour lier mes variables PHP au marqueur nommé.

2. Précisez les moyens utilisés :

Un PC sous Windows avec connexion internet.

Un éditeur de code VsCode.

Pour le développement : HTML, CSS, BOOTSTRAP, JAVASCRIPT, PHP.

3. Avec qui avez-vous travaillé ?

Il s'agit d'un exercice individuel.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► AFPA

Chantier, atelier, service ► Développer les composants d'accès aux données

Période d'exercice ► Du : 10/02/2020 au : 24/03/2020

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 3 ► Développement d'un service d'email transactionnel

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. Définition de l'objectif

Au cours de mon stage j'ai travaillé sur le développement du site web Ikosoft, entreprise de développement de logiciels informatiques, spécialisée dans la gestion de salons de coiffure, instituts de beauté et Spa, créée en 1993.

Une de mes tâches consiste à envoyer à partir d'un formulaire client un mail automatique avec le CV en pièce jointe du postulant à l'administrateur du site web.

Je dois utiliser un **service d'email transactionnel** SendInBlue qui dispose d'un **serveur SMTP**. Protocole de communication utilisé pour transférer les emails vers les serveurs de messageries électroniques.

Un email transactionnel est envoyé pour attester d'une transaction ou d'un événement.

SendInBlue garantit l'arrivée des emails transactionnels en boîte de réception. Il permet aussi de consulter l'historique des emails envoyés, les ouvertures, les clics et beaucoup d'autres informations.

Le site d'Ikosoft est développée avec CakePHP Framework MVC (Model View Controller).

2. Envoi des données via une requête asynchrone

Pour envoyer les données du client vers le serveur PHP, j'ai utilisé une requête asynchrone dans le but d'éviter le rechargement de la page.

2.1. Préparation des données

J'utilise l'objet **FormData** qui me permet de compiler un ensemble de paires clé/valeur à envoyer à l'aide de l'API *fetch*. Le format des données transmises est le même que celui qu'utiliserait la méthode *submit()* du formulaire.

```
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);
```

2.2. Requête asynchrone

J'envoie les données en utilisant l'API *fetch*, alternative à l'objet *XMLHttpRequest* plus puissante et plus flexible. Elle fournit une définition pour trois interfaces Request, Response et Headers et implémente également le mixin Body qu'on va pouvoir utiliser avec nos requêtes.

La méthode `fetch()` renvoie une promesse.

```
/* 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.";
});
});
```

3. Configuration du Framework CakePHP

Pour pouvoir utiliser les paramètres du serveur SMTP je dois déclarer les informations de transport dans le fichier de configuration de CakePHP : `conf/app.php`.

3.1. Transport

```
'EmailTransport' => [
    'sendInBlue' => [
        'className' => 'Smtp',
        /*
         * The following keys are used in SMTP transports:
         */
        'host' => 'smtp-relay.sendinblue.com',
        'port' => 587,
        'username' => 'clients@sokeo.fr',
        'password' => 'aWny9jswQ1qkgLpS',
        'client' => null,
        'tls' => null,
    ],
],
```

3.2. Email

```
'Email' => [  
    'sendInBlue' => [  
        'transport' => 'sendInBlue',  
        'from' => 'noreply@ikosoft-france.fr',  
        //'charset' => 'utf-8',  
        //'headerCharset' => 'utf-8',  
    ],  
],
```

4. Envoi du mail avec pièce jointe et réponse au client

J'importe la classe Email.

```
use Cake\Mailer\Email;  
  
// Récupère les données du formulaire  
$datas = $this->request->getData();  
$refCandidature = $datas['candidature'];
```

Je développe la procédure d'upload du CV.

```
// 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);  
}
```

J'instancie la classe Email puis envoie du mail avec sa pièce jointe.

```
// 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);
```

Prépare la réponse pour le client au format Json si le mail a bien été envoyé.

```
// 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);
```

DOSSIER PROFESSIONNEL (DP)

Sinon je signale l'erreur.

```
// Reponds au client (echec d'envoi du mail)
$responseResult = json_encode(['response' => "Une erreur c'est produite, impossible d'envoyer votre CV."]);
```

Envoi de la réponse au client.

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

2. Précisez les moyens utilisés :

Un PC sous Windows avec connexion internet.

Pour le développement : un éditeur de code VsCode, le Framework CakePHP, Wamp serveur.

3. Avec qui avez-vous travaillé ?

j'ai travaillé seul sur le développement de cette fonctionnalité.

4. Contexte

Nom de l'entreprise, organisme ou association ► *Sokeo*

Chantier, atelier, service ► *Site ikosoft*

Période d'exercice ► Du : *23/03/2020* au : *26/03/2020*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

Déclaration sur l'honneur

Je soussigné.....Eric LANZA..... ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur des réalisations jointes.

Fait à ..Marseille..... le ..06/06/2020.....

Pour faire valoir ce que de droit.

Signature :

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)