



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ Hamadaïne
Nom d'usage ▶ Entrez votre nom d'usage ici.
Prénom ▶ Giovan
Adresse ▶ 1 impasse du petit bootz 53000 Laval

Titre professionnel visé

Développeur web et web mobile

MODALITÉ D'ACCÈS :

- ☒ 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.

DOSSIER PROFESSIONNEL ^(DP)

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

Activité type 1 : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

p. 5

- CP1 : Maquetter une application (*Exercice en formation "Réaliser un cahier des charges"*)

p. 5

- CP2 : Réaliser une interface utilisateur web statique et adaptable

p. 8

- CP3 : Développer une interface utilisateur web dynamique

p. 13

- CP4 : Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

P. 16

Activité type 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

p. 17

- CP1 : Créer une base de données

p. 17

- CP2 : Développer les composants d'accès aux données

p. 21

- CP3 : Développer la partie back-end d'une application web ou web mobile

p. 25

- CP4 : Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

P. 29

Titres, diplômes, CQP, attestations de formation (*facultatif*)

p. 30

Déclaration sur l'honneur

p. 31

Documents illustrant la pratique professionnelle (*facultatif*)

p. 32

Annexes (*Si le RC le prévoit*)

p. 33

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

CP1 - Maquetter une application (élaboration du cahier des charges)

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

Élaboration d'un cahier des charges pour l'application web de Martin Photographe

Le processus d'élaboration d'un cahier des charges pour l'application web de Martin Photographe a suivi plusieurs étapes clés pour répondre à ses besoins spécifiques. Tout d'abord, nous avons pris en compte le contexte de l'entreprise de Martin, sa spécialité photographique, son expérience professionnelle, ses objectifs commerciaux et son public cible.

Ensuite, nous avons procédé à l'identification précise des besoins de Martin. Pour ce faire, nous avons eu des échanges approfondis avec lui afin de comprendre ses exigences en matière de transmission de son travail à ses clients via une application web. Nous avons pris en compte différentes fonctionnalités souhaitées, telles que le partage sécurisé des images, la possibilité pour les clients de sélectionner et de télécharger des photos, ainsi que la gestion des droits d'accès.

Une fois les besoins identifiés, nous avons proposé des solutions adaptées à Martin. Nous avons élaboré un plan détaillé pour la création de l'application web, en mettant l'accent sur les fonctionnalités essentielles, une interface utilisateur conviviale et une sécurité des données optimale. Nous avons pris en considération les aspects techniques et les contraintes liées au développement de l'application.

Pour structurer l'application, nous avons travaillé sur l'enchaînement des écrans, également connu sous le nom de sitemap. Cela nous a permis de définir la navigation et la structure des différentes pages de l'application de manière claire et logique. Le sitemap nous a également aidé à visualiser comment les utilisateurs interagiraient avec l'application et à organiser les fonctionnalités de manière cohérente.

Une fois que la structure a été établie, nous avons procédé à la partition du travail en tâches simples, en utilisant le sitemap comme guide. Cela nous a permis de découper le développement de l'application en étapes claires et concises, facilitant ainsi la gestion du projet et garantissant une progression efficace.

L'analyse fonctionnelle, également appelée use case, a joué un rôle essentiel dans le processus. Nous avons réalisé une étude approfondie des interactions entre les utilisateurs et l'application, afin de définir les cas d'utilisation spécifiques. Cela nous a permis de mettre en évidence les principales fonctionnalités de l'application et de garantir qu'elles répondaient aux besoins de Martin et de ses clients.

Le zoning, une autre étape importante, a consisté à créer des zones claires dans l'interface utilisateur de l'application. Nous avons défini l'emplacement des différents éléments graphiques, tels que les boutons, les menus et les sections, afin d'optimiser l'expérience utilisateur et de rendre l'application plus intuitive et facile à utiliser.

DOSSIER PROFESSIONNEL ^(DP)

Enfin, nous avons créé des maquettes du site, représentant les différentes pages de l'application avec leur aspect visuel. Cela a permis à Martin de visualiser concrètement l'apparence finale de l'application, ainsi que les interactions possibles avec celle-ci.

En suivant ces étapes détaillées, nous avons pu élaborer un cahier des charges complet pour l'application web de Martin Photographe. L'application proposée répondra précisément à ses besoins spécifiques en matière de transmission de son travail à ses clients. Elle offrira une expérience utilisateur conviviale, une gestion sécurisée des images et une interface intuitive pour faciliter le partage et la sélection des photos.

(Annexe 1)

2. Précisez les moyens utilisés :

Pour réaliser efficacement les différentes étapes de l'élaboration d'un cahier des charges, plusieurs outils peuvent être utilisés pour faciliter la collaboration et la gestion du projet. Voici quelques-uns des outils couramment utilisés :

LucidChart : LucidChart est une plateforme en ligne qui permet de créer des diagrammes et des schémas visuels. Il peut être utilisé pour concevoir le sitemap de l'application web, en représentant graphiquement la structure et l'enchaînement des différentes pages. Cela facilite la compréhension de la navigation et de l'organisation de l'application.

Figma : Figma est un outil de conception d'interfaces utilisateur (UI) et d'expérience utilisateur (UX) basé sur le cloud. Il offre une interface collaborative qui permet à plusieurs membres d'une équipe de travailler simultanément sur la conception de l'application. Figma peut être utilisé pour créer les maquettes du site, en donnant une représentation visuelle précise de l'apparence finale de l'application.

Google Docs : Google Docs est un outil de traitement de texte en ligne qui facilite la collaboration et le partage de documents. Il peut être utilisé pour rédiger le cahier des charges lui-même, en permettant à plusieurs membres de l'équipe de travailler simultanément sur le document. Google Docs offre également des fonctionnalités de commentaires et de suivi des modifications, ce qui facilite la révision et la mise à jour du document.

Trello : Trello est un outil de gestion de projet basé sur des tableaux virtuels. Il permet de créer des listes de tâches et de les organiser en fonction de leur état d'avancement. Trello peut être utilisé pour partitionner le travail en tâches simples, en créant des cartes pour chaque tâche à accomplir. Les membres de l'équipe peuvent facilement collaborer en ajoutant des commentaires, des pièces jointes et en déplaçant les cartes d'une liste à l'autre au fur et à mesure de l'avancement du projet.

En utilisant ces outils, vous pouvez faciliter la collaboration, la visualisation et la gestion du processus d'élaboration du cahier des charges de l'application web de Martin Photographe. Chacun de ces outils offre des fonctionnalités spécifiques qui peuvent contribuer à optimiser le workflow et à garantir la réussite du projet.

DOSSIER PROFESSIONNEL ^(DP)

3. Avec qui avez-vous travaillé ?

Dans ce projet d'élaboration du cahier des charges pour l'application web de Martin Photographe, j'ai travaillé en autonomie. J'ai utilisé mes compétences de rédaction et mes connaissances en développement web pour mener à bien les différentes étapes du processus.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Via Formation

Chantier, atelier, service ▶ réaliser en formation

Période d'exercice ▶ Du : 20/05/2023 au : 21/05/2023

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

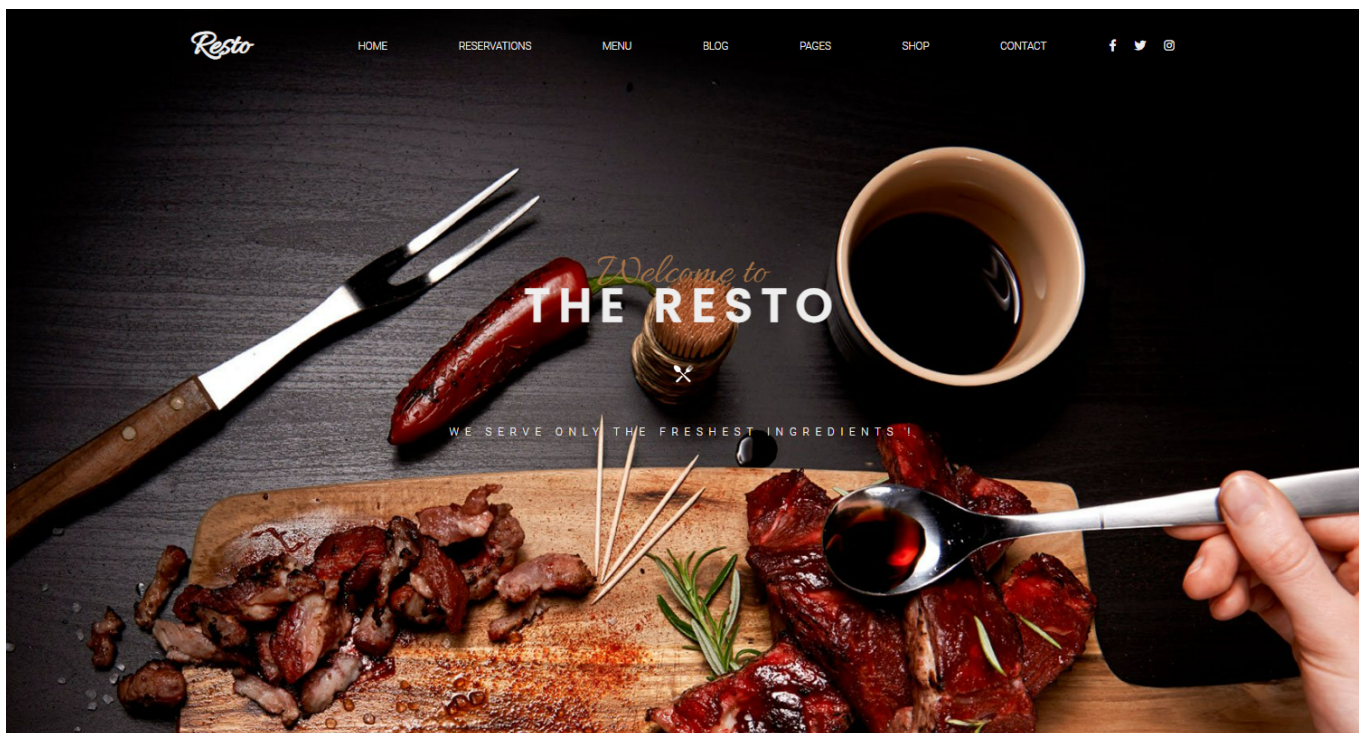
CP2 - Réaliser une interface utilisateur web statique et adaptable (site le resto)

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

En me basant sur les connaissances acquises au cours de mes études, j'ai créé plusieurs sites web statiques pour me familiariser avec les technologies HTML, CSS et SASS. Je souhaite vous présenter un projet spécifique : un site web pour un restaurant, conçu entièrement en mode statique et optimisé pour les appareils mobiles.

L'objectif de ce projet était de le rendre aussi simple que possible. Pour y parvenir, j'ai opté pour l'utilisation de HTML et de SASS, sans avoir recours à aucun framework. Néanmoins, j'ai intégré la bibliothèque d'icônes FontAwesome. En ce qui concerne JavaScript, j'ai décidé de ne pas l'utiliser pour ce projet statique.

Pour réaliser ce projet, j'ai décidé d'adopter la stratégie du mobile-first. En effet, du fait de l'utilisation majoritaire et grandissante des mobiles pour naviguer sur le web, il semble judicieux d'adopter cette approche pour réaliser ce projet



1. Structure des pages

Pour structurer les pages du projet, pour favoriser l'accessibilité et le SEO, j'ai choisi d'utiliser une bonne sémantique HTML.

Voici par exemple la structure générale de la page d'accueil:

```
index.html > html > head
1  <!DOCTYPE html>
2  <html lang="fr">
3  > <head> ...
21 </head>
22 <body>
23 > <header> ...
51 </header>
52 <main>
53 > <section> ...
73 </section>
74 > <section class="section-menu"> ...
193 </section>
194 > <section> ...
248 </section>
249 > <section> ...
314 </section>
315 > <section class="section-about"> ...
338 </section>
339 </main>
340 > <footer> ...
370 </footer>
371 </body>
372 </html>
373
```

On remarque ici que le corps de la page est divisé en trois grandes parties : la partie header, la partie main et le footer.

La partie header contient tout ce qui a attrait à la partie supérieure de la page, notamment la navigation.

La partie main quant à elle, contient l'ensemble des informations principales de la page. On remarque qu'ici cette partie contient cinq sections.

Et la partie footer inclut des liens de navigation essentiels, des informations de contact et des mentions légales.

```
<section>
  <div class="reservation">
    <div class="grid">
      <div class="left-reservation"> ...
    </div>
    <div class="right-reservation"> ...
  </div>
</div>
</section>
```

Cette section de la page, dédiée à la réservation, est structurée à l'aide d'une balise 'div' principale. À l'intérieur, elle contient une autre balise 'div' dotée de la classe 'grid'. Cette dernière se divise en deux balises 'div' : l'une pour la partie gauche et l'autre pour la partie droite de la section réservation.

```
<header>
  <div class="navbar">
    <div class="logo">
      <h1 class="font2">Resto</h1>
    </div>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">Reservations</a></li>
      <li><a href="#">Menu</a></li>
      <li><a href="#">blog</a></li>
      <li><a href="#">pages</a></li>
      <li><a href="#">shop</a></li>
      <li><a href="#">contact</a></li>
    </ul>
    <div class="icons">
      <i class="fab fa-facebook-f"></i>
      <i class="fab fa-twitter"></i>
      <i class="fab fa-instagram"></i>
    </div>
  </div>
  <div class="hero">
    <div class="hero-text">
      <h2 class="font2">Welcome to</h2>
      <h1>the Resto</h1>
      
      <p>we serve only the freshest ingredients !</p>
    </div>
  </div>
</header>
```

Le header du site web comporte deux éléments clés : une barre de navigation pour orienter facilement l'utilisateur à travers les différentes pages, et une 'hero banner' qui occupe une place centrale sur la page d'accueil.

La navigation du site est structurée en trois sections distinctes. La première partie utilise une balise 'div' pour encapsuler le logo du site. La deuxième section met en œuvre une balise 'ul', où chaque balise 'li' contient un lien hypertexte vers une des pages accessibles du site. La troisième et dernière section est spécifiquement dédiée aux icônes des réseaux sociaux.

2. responsive et media queries

Pour suivre la demande d'avoir un style responsive, j'ai utilisé des media queries.

```
.gird {  
  display: grid;  
  grid-template-columns: 30% 70%;  
  max-width: 1300px;  
  margin: 80px auto 150px;  
  
  @media screen and (max-width: 1100px) {  
    grid-template-columns: 100%;  
    .form {  
      display: grid;  
      grid-template-columns: 1fr 1fr 1fr;  
      padding: 0 60px;  
      gap: 30px;  
  
      @media screen and (max-width: 768px) {  
        grid-template-columns: 1fr;  
        padding: 0;  
      }  
    }  
  }  
}
```

Ici la media query utilise un breakpoint. Ce breakpoint s'applique aux écrans ayant une largeur maximum de 1100px (desktop).

Le SASS change notamment le nombre de colonne du display:grid en n'en conservant plus qu'une avec grid-template-columns :100% ;

2. Précisez les moyens utilisés :

Pour réaliser ce projet, j'ai utilisé VSCode avec Prettier pour formater le code pour vérifier l'accessibilité et les bonnes pratiques de SEO, j'ai utilisé l'outil Lighthouse de la console de développement de mon navigateur (Chrome). J'ai également utilisé la console pour simuler les media queries.

Je me suis référé au site MDN (Mozilla Developer Network) pour mieux utiliser les media queries ayant attrait aux préférences de thème et de contraste de l'utilisateur

3. Avec qui avez-vous travaillé ?

J'ai travaillé en autonomie sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Via Formation

DOSSIER PROFESSIONNEL ^(DP)

Chantier, atelier, service ▶

Projet personnel réalisé en cours de formation.

Période d'exercice ▶ Du : 01/06/2023 au : 12/06/2023

5. Informations complémentaires *(facultatif)*

Activité-type 1

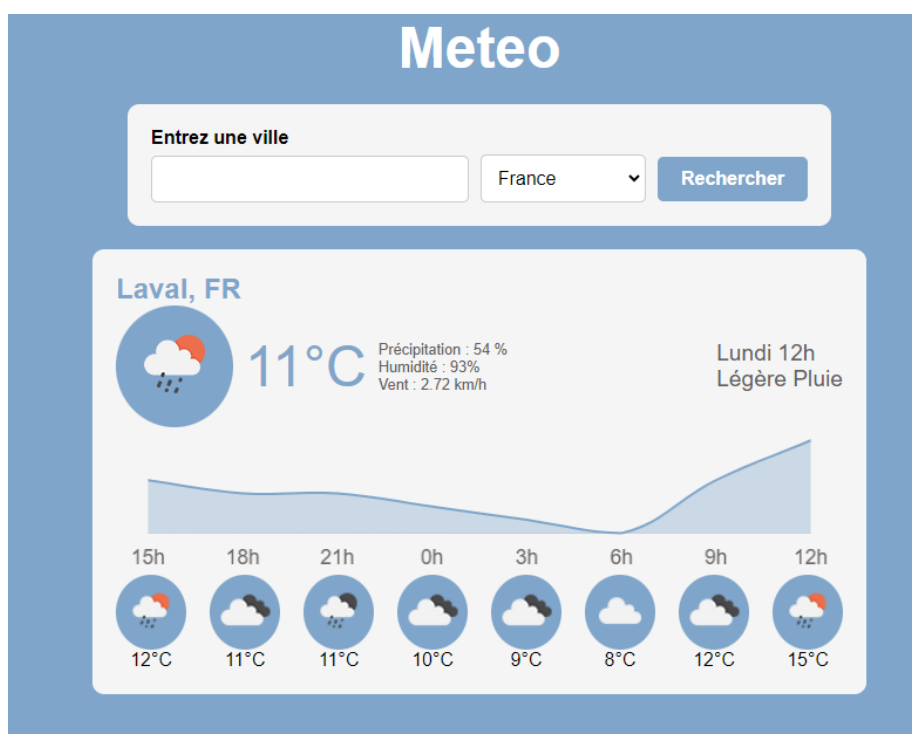
Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

CP3 - Développer une interface utilisateur web dynamique (App Meteo)

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

Dans le cadre de l'obtention de mon titre professionnel, j'ai développé une application météorologique. Cette application offre la possibilité de consulter les conditions météorologiques dans la ville et le pays de votre choix. De plus, elle actualise les prévisions toutes les trois heures pour une période de 24 heures. Pour une visualisation plus intuitive des données, j'ai intégré la bibliothèque Chart.js, qui permet d'afficher une courbe des températures sur cette même période.

J'ai utilisé l'API de OpenWeatherMap pour récupérer les données météorologiques.



DOSSIER PROFESSIONNEL (DP)

1. Le Fetch

```
function getResults(ville, country) {  
  fetch(  
    `${api.base}forecast?q=${ville},${country}&lang=fr&  
    units=metric&appid=${api.key}&cnt=9`  
  )  
    .then((weather) => {  
      return weather.json();  
    })  
    .then((weather) => {  
      if (weather.cod === "404") {  
        alert("Ville non trouvée");  
      } else {  
        displayResults(weather);  
      }  
    })  
  });  
}
```

La fonction **getResults** prend en paramètre la saisie de l'utilisateur pour effectuer une requête à l'API météorologique via la méthode **fetch**. Une fois la réponse obtenue, elle la convertit en format JSON pour un traitement plus aisé. Si la ville saisie n'est pas trouvée dans la base de données de l'API, une alerte est générée pour informer l'utilisateur. Dans le cas contraire, la fonction **displayResults** est appelée avec les données reçues en paramètre afin d'afficher les informations météorologiques correspondantes.

2. Précisez les moyens utilisés :

Pour mener à bien ce projet, j'ai employé les langages HTML, CSS et JavaScript. En complément, j'ai fait appel à la bibliothèque Chart.js et à sa documentation associée pour créer la courbe de températures.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Via Formation

Chantier, atelier, service ▶ *Projet personnel réalisé en cours de formation*

Période d'exercice ▶ Du : 15/06/2023 au : 20/06/2023

5. Informations complémentaires (facultatif)

Lien Github : <https://github.com/GioGio-Armani/MeteoApi>

DOSSIER PROFESSIONNEL ^(DP)

DOSSIER PROFESSIONNEL ^(DP)

Activité-type 1 Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

CP4 - Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce (exemple à mettre ici)

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

2. Précisez les moyens utilisés :

3. Avec qui avez-vous travaillé ?

4. Contexte

Nom de l'entreprise, organisme ou association - Via Formation

Chantier, atelier, service - Cliquez ici pour taper du texte.

Période d'exercice - Du : Cliquez ici au : Cliquez ici

5. Informations complémentaires (facultatif)

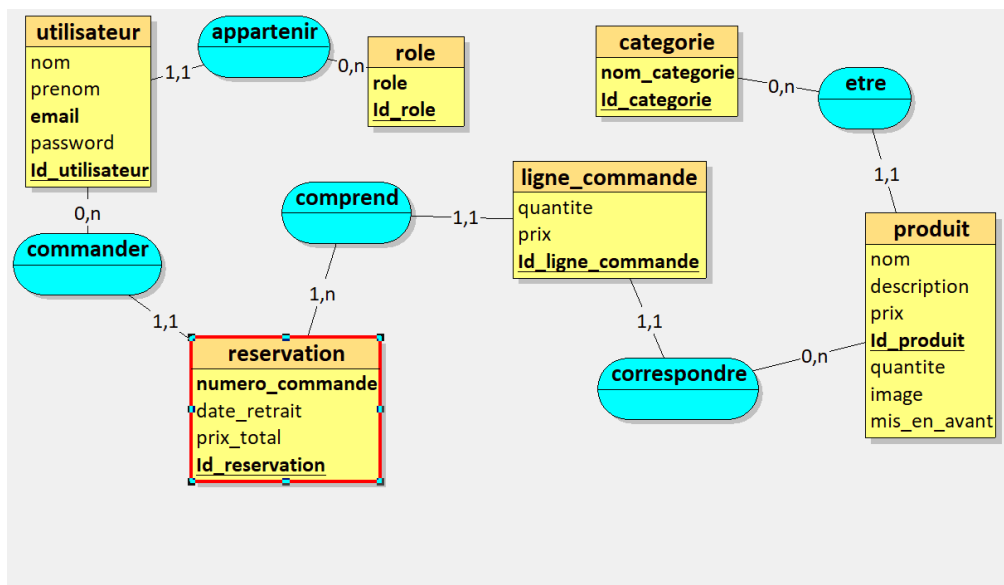
Activité-type 2

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

CP1 - Créer une base de données

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

Dans le cadre de ma formation, un exercice pratique axé sur les bases de données nous a été assigné. L'objectif était de concevoir un système de réservation en ligne pour la boutique Eden Zéro Déchet, située en centre-ville. Cette boutique écologique vend des produits locaux sans emballage en quantité limitée, ce qui entraîne fréquemment des ruptures de stock. Cela provoque des déplacements inutiles pour les clients, allant à l'encontre de la philosophie écologique de l'entreprise. Pour commencer, nous devons créer le Modèle Conceptuel de Données (MCD), le Modèle Logique de Données (MLD), ainsi que le script de la base de données.



On peut remarquer sur ce MCD les différentes cardinalités entre chaque tables, par exemple:

- Un utilisateur peut avoir 0 ou plusieurs (0N) réservation et il peut avoir au minimum et au maximum un seul rôle (1,1)
- Une réservation peut comprendre au minimum une ou plusieurs lignes de commande (1,n)

2. Le MLD

Après avoir défini la représentation visuelle de ma base de données avec le MCD, j'ai créé un MLD (modèle logique des données) pour représenter les relations entre les tables.

3. Le MPD

Puis une fois le MLD réalisé, je me suis servi de Looping pour réaliser le MPD (modèle physique des données).

```
CREATE TABLE role(
    Id_role INT AUTO_INCREMENT,
    role VARCHAR(255) NOT NULL,
    PRIMARY KEY(Id_role),
    UNIQUE(role)
);

CREATE TABLE categorie(
    Id_categorie INT AUTO_INCREMENT,
    nom_categorie VARCHAR(255) NOT NULL,
    PRIMARY KEY(Id_categorie),
    UNIQUE(nom_categorie)
);

CREATE TABLE UTILISATEUR(
    Id_utilisateur INT AUTO_INCREMENT,
    nom VARCHAR(255) NOT NULL,
    prenom VARCHAR(255) ,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    Id_role INT NOT NULL,
    PRIMARY KEY(Id_utilisateur),
    UNIQUE(email),
    FOREIGN KEY(Id_role) REFERENCES role(Id_role)
);

CREATE TABLE produit(
    Id_produit INT AUTO_INCREMENT,
    nom VARCHAR(255) NOT NULL,
    description TEXT,
    prix DECIMAL(15,2) ,
    quantite DECIMAL(15,2) NOT NULL,
    image VARCHAR(255) ,
    mis_en_avant BOOLEAN NOT NULL,
    Id_categorie INT NOT NULL,
    PRIMARY KEY(Id_produit),
    FOREIGN KEY(Id_categorie) REFERENCES categorie(Id_categorie)
);
```

```
CREATE TABLE reservation(  
  Id_reservation INT AUTO_INCREMENT,  
  numero_commande INT NOT NULL,  
  date_retrait DATE NOT NULL,  
  prix_total DECIMAL(15,2) NOT NULL,  
  Id_utilisateur INT NOT NULL,  
  PRIMARY KEY(Id_reservation),  
  UNIQUE(numero_commande),  
  FOREIGN KEY(Id_utilisateur) REFERENCES UTILISATEUR(Id_utilisateur)  
);  
  
CREATE TABLE ligne_commande(  
  Id_ligne_commande INT AUTO_INCREMENT,  
  quantite DECIMAL(15,2) NOT NULL,  
  prix DECIMAL(15,2) NOT NULL,  
  Id_produit INT NOT NULL,  
  Id_reservation INT NOT NULL,  
  PRIMARY KEY(Id_ligne_commande),  
  FOREIGN KEY(Id_produit) REFERENCES produit(Id_produit),  
  FOREIGN KEY(Id_reservation) REFERENCES reservation(Id_reservation)  
);
```

2. Précisez les moyens utilisés :

Pour la conception du Modèle Conceptuel de Données (MCD), j'ai utilisé le logiciel Looping. Ce même logiciel m'a également permis de générer le Modèle Physique de Données (MPD)

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Via Formation*

Chantier, atelier, service ▶ *Projet personnel réalisé en cours de formation*

Période d'exercice ▶ Du : 22/06/2023 au : 15/07/2023

DOSSIER PROFESSIONNEL ^(DP)

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

CP2 - Développer les composants d'accès aux données

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

Dans le cadre d'un projet visant à créer un mini réseau social, j'ai mis en place l'architecture en utilisant NodeJS et Express. J'ai opté pour l'utilisation de Mongoose pour effectuer les requêtes vers ma base de données MongoDB.

1.Initialisation de la connexion

Pour initialiser la connexion avec MongoDB j'ai créé un fichier db.js pour gérer la connexion.

```
1  const mongoose = require("mongoose");
2
3  // connexion à la base de données
4
5  const connectDB = async () => {
6    try {
7      mongoose.set("strictQuery", false);
8      await mongoose
9        .connect(process.env.MONGO_URI)
10       .then(() => console.log("Mongo connecté"));
11    } catch (err) {
12      console.log(err);
13      process.exit(1);
14    }
15  };
16
17  module.exports = connectDB;
```

2. Les modèles

Dans le cadre de la création et de la synchronisation des tables de ma base de données, j'ai pris l'initiative de créer un fichier `post.model.js` pour chaque entité de ma base de données. Cela signifie que pour chaque type d'objet ou de données que je souhaite stocker dans ma base de données, j'ai créé un fichier séparé avec l'extension `.js` pour définir sa structure et ses propriétés.

Par exemple :

```
1  const mongoose = require("mongoose");
2
3  const postSchema = new mongoose.Schema(
4    {
5      message: {
6        type: String,
7        required: true,
8      },
9      author: {
10       type: String,
11       required: true,
12     },
13     likes: {
14       type: [String],
15     },
16   },
17   {
18     timestamps: true,
19   }
20 );
21
22 module.exports = mongoose.model("post", postSchema);
23
```

3. Les controller

Puis, pour chaque entité, j'ai créé un fichier `controller.js` qui va exécuter les différentes requêtes CRUD (Créer, Lire, Mettre à jour, Supprimer) sur la base de données en utilisant le contenu du corps (body) de la requête. Par exemple le fichier `post.controller.js` :

DOSSIER PROFESSIONNEL (DP)

```
1  const PostModel = require("../models/post.model");
2
3  module.exports.getPosts = async (req, res) => {
4    const posts = await PostModel.find({});
5    res.status(200).json(posts);
6  };
7
8  module.exports.setPost = async (req, res) => {
9    if (!req.body.message) {
10     return res.status(400).json({
11       message: "Le contenu ne peut pas être vide !",
12     });
13   }
14
15   const post = await PostModel.create({
16     message: req.body.message,
17     author: req.body.author,
18   });
19   res.status(200).json(post);
20 };
```

2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai opté pour l'utilisation de la base de données MongoDB en combinant la puissance de la bibliothèque Mongoose. Cette approche m'a permis d'établir une connexion sécurisée à la base de données tout en facilitant l'envoi de requêtes de manière efficace.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Via Formation*

Chantier, atelier, service ▶ *Projet personnel réalisé en cours de formation*

Période d'exercice ▶ Du : *15/07/2023* au : *01/08/2023*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL ^(DP)

Activité-type 2

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

CP3 - Développer la partie back-end d'une application web ou web mobile

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

Pour m'exercer, j'ai souhaité réaliser un projet relativement simple : un petit blog. En ce qui concerne la partie back-end de cette application, j'ai également opté pour Express. En ce qui concerne la partie API de mon application, j'ai commencé par créer des routeurs.

Le cheminement des requêtes au niveau des routes est le suivant : la requête arrive d'abord au routeur principal, puis elle est dirigée vers le routeur spécifique à l'entité concernée, traverse différents middlewares, et enfin atteint le controller.

1.Initialisation du serveur

```
const express = require("express");
const { engine } = require("express-handlebars");
const dotenv = require("dotenv").config();

const connectDB = require("../api/config/bdd");
const router = require("../api/router");

const app = express();
const port = process.env.PORT;
connectDB();

app.engine("hbs", engine({ extname: "hbs" }));
app.set("view engine", "hbs");

app.use(express.json());
app.use(express.urlencoded({ extended: true }));

app.use("/", router);

app.listen(port, () => {
  console.log(`app lancé sur http://localhost:${port}`);
});
```

Ici on déclare notre serveur.

On remarque l'utilisation de Handlebars pour la gestion des views.

On remarque également l'utilisation de notre router principal. Ce routeur va se charger de gérer les requêtes effectuées

2.Routeur principal

```
L7
L8 app.use("/", router);
L9
```

Dans ce petit projet, j'ai décidé de regrouper toutes les routes de l'API dans un fichier principal. Je le déclare dans api/router.js. À l'intérieur de ce fichier, je vais déclarer les points d'accès.

3.Router d'entité

```
// contact
router
  .route("/contact")
  .get(contactController.get)
  .post(contactController.post);

// articles
router
  .route("/newarticle")
  .get(articleController.getNewArticle)
  .post(articleController.postNewArticle);
router.route("/article/:id").get(articleController.getArticle);

module.exports = router;
```

Dans ce fichier, je déclare les points d'accès de l'URL demandée, avec les méthodes associées telles que GET, POST, PATCH, DELETE, etc.

En ce qui concerne l'organisation du routeur de l'API, prenons l'exemple de la récupération d'un lien. Pour effectuer cette action, nous effectuons une requête GET à l'adresse suivante :

[https://mon-domaine.fr/\[alias-du-lien\]](https://mon-domaine.fr/[alias-du-lien]).

4.Controller

Une fois la requête effectuée, c'est le controller qui va prendre le relais et traiter la partie "logique" de notre API, il va traiter les données de la requête puis interagir avec la couche des données de notre API et enfin renvoyer des données en retour. En voici un exemple avec le controller de création d'un message de contact, contenu dans le fichier `api/controllers/contact.controller.js`.

```
const contactModel = require("../models/contactModel");

module.exports = {
  get: (req, res) => {
    const error = req.query.error;
    res.render("contact", { error });
  },

  post: async (req, res) => {
    if (!req.body.message) {
      return res.status(400).redirect("/contact?error=Message manquant");
    }
    await contactModel.create({
      email: req.body.mail,
      name: req.body.name,
      message: req.body.message,
    });
    console.log(req.body);
    res.redirect("/contact");
  },
};
```

Dans ce cas, nous stockons les informations dans la table 'contact', qui contient les champs email, nom et message.

Dans le contrôleur, lorsqu'une requête GET est effectuée, il renvoie la vue 'contact'. En revanche, lorsqu'une méthode POST est utilisée, l'API vérifie si les champs sont correctement remplis. Si ce n'est pas le cas, elle redirige avec un message d'erreur en tant que paramètre. Cependant, si tous les champs sont correctement remplis, le contrôleur crée une entrée dans la table 'contact' pour enregistrer ce message

2. Précisez les moyens utilisés :

Pour construire ce projet, j'ai fait usage des bibliothèques suivantes :

- Express, en m'appuyant sur sa documentation.
- Express-handlebars pour la gestion des vues.
- Mongoose pour établir la communication avec la base de données.

3. Avec qui avez-vous travaillé ?

DOSSIER PROFESSIONNEL ^(DP)

J'ai travaillé seul sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *Via Formation*

Chantier, atelier, service ▶ *Projet personnel réalisé en cours de formation*

Période d'exercice ▶ Du : *15/08/2023* au : *25/08/2023*

5. Informations complémentaires *(facultatif)*

DOSSIER PROFESSIONNEL ^(DP)

Activité-type 2

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

CP4 - *Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce (exemple à mettre ici)*

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

2. Précisez les moyens utilisés :

3. Avec qui avez-vous travaillé ?

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cliquez ici pour taper du texte.

Chantier, atelier, service ▶ Cliquez ici pour taper du texte.

Période d'exercice ▶ Du : Cliquez ici au : Cliquez ici

5. Informations complémentaires (facultatif)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
BEP CUISINE	CFA des 3 villes	2012
BAC PRO CUISINE	Le mans	2013

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] **Giovan HAMADAINÉ**,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à **LAVAL** le **24/10/2023**
pour faire valoir ce que de droit.

Signature :

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)

Annexe 1 : Cahier des charges Activité-type1 - CP1