

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance*

▶ NATSIOS

*Nom d'usage*

▶ Entrez votre nom d'usage ici.

*Prénom*

▶ Sotirios

*Adresse*

▶ 1 Chemin de la Capuche

## 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

<b>Intitulé de l'activité-type n° 1 Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité.</b>	<b>p.</b>	<b>5</b>
▶ CP 1 Création de wireframes et de prototypes (Maquetter une application)	p.	5
▶ CP 2 Interface avec des données de démonstration (Réaliser une interface statique et adaptable)	p.	11
▶ CP 3 Données dynamiques (E-commerce)	p.	15
<b>Intitulé de l'activité-type n° 2 Développer la partie back-end d'un e-shop</b>	<b>p.</b>	<b>19</b>
▶ CP 4 Diagramme pour la base de données .....	p.	19
▶ CP 5 Création de la base de données .....	p.	22
▶ CP 6 Création des tables et des relations.....	p.	23
<b>Intitulé de l'activité-type n° 3 Développer la partie back-end avec NoSQL</b>	<b>p.</b>	<b>25</b>
▶ CP 7 Développement d'une application CRUD NoSQL .....	p.	25
<b>Titres, diplômes, CQP, attestations de formation</b> <i>(facultatif)</i>	<b>p.</b>	
<b>Déclaration sur l'honneur</b>	<b>p.</b>	
<b>Documents illustrant la pratique professionnelle</b> <i>(facultatif)</i>	<b>p.</b>	
<b>Annexes</b> <i>(Si le RC le prévoit)</i>	<b>p.</b>	

# **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** ► Création de wireframes et de prototypes. (Maquetter une application)

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

#### Conception et création de wireframes et de prototypes pour l'e-shop.

##### Wireframes et prototypes

Pour mon projet de stage, mon responsable m'a chargé de redéfinir et de moderniser la plateforme de commerce électronique de l'entreprise. Le site existant était construit à l'aide de WordPress, ce qui entraînait des performances lentes, avec des temps de chargement des pages prenant souvent plusieurs secondes, voire des arrêts occasionnels. De plus, mon responsable a exprimé des préoccupations concernant les coûts de maintenance élevés associés à la plateforme. Pour résoudre ces problèmes, on m'a également demandé de faire passer la boutique en ligne à une pile technologique plus efficace et rentable.

Après avoir discuté des exigences du projet avec mon responsable, j'ai proposé d'adopter une pile technologique composée de Directus pour le backend et de Next.js pour le frontend. Ayant déjà travaillé sur plusieurs projets en utilisant Next.js et étant conscient des avis positifs concernant Directus, j'étais convaincu que cette combinaison répondrait aux préoccupations de performance et de coût de maintenance. Après avoir évalué ma proposition, nous avons décidé de poursuivre cette approche technologique.

Mon responsable a souligné l'importance de créer un design intuitif pour la navigation des utilisateurs, ainsi qu'une interface conviviale pour les tâches administratives. Directus offre un système de gestion très simple et efficace pour la base de données et les données qui y sont stockées. Après avoir démontré un exemple des capacités de Directus, mon responsable a été convaincu que ce serait un excellent choix pour les exigences de notre projet.

##### 1. Wireframes

Une fois d'accord sur la pile technologique souhaitée, j'ai immédiatement commencé à concevoir la version modernisée du site web. La première étape de ce processus consistait à créer des wireframes, qui servent de guide de base pour le développement et la conception ultérieurs des prototypes. Étant donné que les principes de design moderne privilégient une approche "mobile-first", j'ai commencé le processus de création des wireframes avec la vue mobile (petits écrans). Pour cette tâche, j'ai utilisé la plateforme Draw.io, un outil avec lequel je suis bien familiarisé.

Wireframe de la page d'accueil pour les mobiles :

Dans ce wireframe, nous observons une séparation claire de la vue en trois sections distinctes : la section supérieure/en-tête, la section principale/contenu et la section inférieure/pied de page.

La section d'en-tête est conçue pour informer les visiteurs et les clients des liens disponibles via le menu, de toute information supplémentaire spécifiée par le responsable, ainsi que des catégories de la boutique en ligne.

La section principale présente deux carrousels de produits : un pour les produits suggérés et un autre pour les nouveaux articles ajoutés à la boutique.

La section de pied de page fonctionne comme un sous-menu, fournissant des liens supplémentaires vers des pages juridiques, ainsi que des informations concernant les méthodes de paiement et les droits d'auteur du site.

Les sections d'en-tête et de pied de page restent cohérentes sur l'ensemble du site web, avec des ajustements mineurs apportés à la section d'en-tête pour s'adapter à la taille de l'écran et au contenu affiché sur chaque page.

Wireframe de la page d'accueil pour la vue desktop :

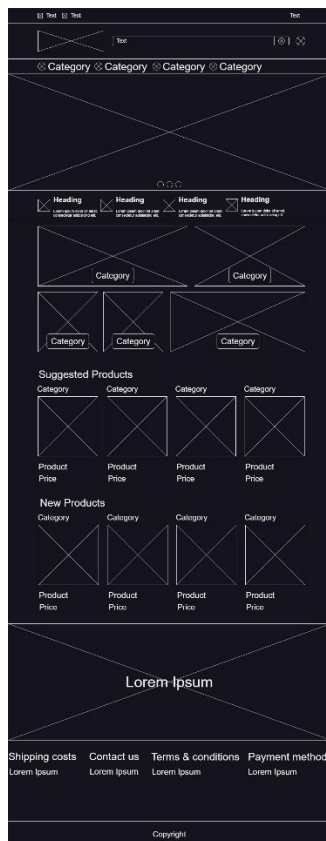
Similaire au design de la vue mobile, la vue desktop est également divisée en trois sections : en-tête, principale et pied de page. La principale différence dans le design desktop est la taille des images. Puisque les écrans plus grands peuvent accueillir des graphiques plus grands, les images sont agrandies sans perte de qualité. De plus, un sous-menu de style grille pour les catégories est intégré dans la section principale. Les sections d'en-tête et de pied de page restent cohérentes sur l'ensemble du site web, avec des ajustements mineurs apportés à la section d'en-tête pour s'adapter au contenu affiché sur chaque page.

Prototypes :

Pour la création des prototypes, j'ai utilisé Figma, car il offre un niveau gratuit et fournit une suite complète d'outils de conception. J'ai développé des prototypes fonctionnels pour les vues mobile et desktop, reproduisant fidèlement les effets et le design du produit final. Tout au long de ce processus, j'ai strictement suivi les instructions de mon responsable et les conceptions des wireframes pour m'assurer que les prototypes répondaient aux exigences spécifiées.

Exemples de wireframes et de prototypes :

# DOSSIER PROFESSIONNEL (DP)



Header & Menu

Hero Section

Info banner

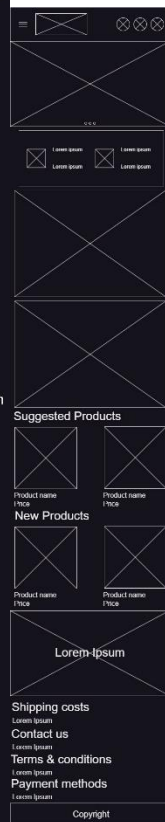
Categories

Products Section

Parallax info

Footer

Copyright



Header & Menu

Hero Section

Info banner

Categories

Products Section

Parallax info

Footer

Copyright

Home - Cart

+30 2314 011 186

petsquare.store@gmail.com

Ο λογαριασμός μου

Αναζήτηση προϊόντος

Αρχική

Σκύλος

Γάτα

Πτηνά

Τρωκτικά

Ψάρια

Χονδρική

Καλάθι αγορών

Περιεχόμενο καλαθιού

ⓧ Άδειασμα καλαθιού

Προϊόν	Τιμή	Ποσότητα	Σύνολο
Premil Fish	25.00 €	1	25.00 €

Ενημέρωση καλαθιού

Σύνολο καλαθιού

Υποσύνολο

25.00 €

Αποστολή

Έως 21 κιλά: 4.00 €

Αποστολή σε Κεντρική Μακεδονία

Αλλαγή διεύθυνσης

Σύνολο

29.00 €

Ολοκλήρωση παραγγελίας

Μεταφορικά

Για αγορές άνω των 30 ευρώ, δωρεάν μεταφορικά. Κάτω των 30 ευρώ η χρέωση είναι 4 ευρώ. Παράδοση 1-4 εργάσιμες ημέρες.

Για παραγγελίες με βάρος άνω των 21 κιλών, η χρέωση γίνεται κατόπιν συνεννόησης. Παραγγελίες άνω των 21 κιλών αποστέλλονται σε 2 ή και περισσότερα δέματα.

Επικοινωνία

petsquare.store@gmail.com

2314-011-186

Φιλιππουπόλεως 50, Αμπελόκηποι Θεσσαλονίκης, 56123

Όροι και προϋποθέσεις

Όροι και προϋποθέσεις

Πολιτική Απορρήτου

Πολιτική επιστροφών & ακυρώσεων

Συναλλαγές με κάρτα

Τραπεζικοί λογαριασμοί

Πειραιώς: GR64 0172 2130 0052 1309 0814 992

Eurobank: GR68 0260 0600 0000 4028 1211 938





Petsquare 2024 - All rights reserved ©

Page 8

DOSSIER PROFESSIONNEL-Version du 11/09/2017



iPhone SE - Cart

Καλάθι αγορών

Περιοχόμενο καλάθιού

✕

Άδειασμα καλάθιού

Προϊόν:

Premil Fish

Τιμή:

25.00 €

Ποσότητα:

1

Σύνολο:

25.00 €

Ενημέρωση καλάθιού

Σύνολο καλάθιού

Υποσύνολο:


25.00 €

Αποστολή:

Έως 21 κιλά: 4.00 €

Αποστολή σε Κεντρική Μακεδονία

Αλλαγή διεύθυνσης



Τελικό σύνολο:

29.00 €

Ολοκλήρωση παραγγελίας

Μεταφορικά

Για αγορές άνω των 30 ευρώ, δωρεάν μεταφορικά. Κάτω των 30 ευρώ η χρέωση είναι 4 ευρώ. Παράδοση 1-4 εργάσιμες ημέρες.

Για παραγγελίες με βάρος άνω των 21 κιλών, η χρέωση γίνεται κατόπιν συνεννόησης. Παραγγελίες άνω των 21 κιλών αποστέλλονται σε 2 ή και περισσότερα δέματα.

Επικοινωνία

✉ [petsquare.store@gmail.com](mailto:petsquare.store@gmail.com)

☎ 2314-011-186

📍 Φιλιππουπόλεως 50, Αμπελόκηποι  
Θεσσαλονίκης, 56123






Όροι και προϋποθέσεις

Όροι και προϋποθέσεις

Πολιτική Απορρήτου

Πολιτική επιστροφών & ακυρώσεων

Συναλλαγές με κάρτα

🏦 Τραπεζικοί λογαριασμοί

Πειραιώς: GR64 0172 2130 0052 1309 0814 992

Eurobank: GR68 0260 0600 0000 4020 1211 938

Petsquare 2024 - All rights reserved ®

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé Draw.io et figma

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Petsquare GR*

Chantier, atelier, service ► Projet professionnel réalisé en cours de stage.

Période d'exercice ► Du : *04/11/2024* au : *13/12/2024*

## 5. Informations complémentaires (facultatif)

Le dépôt de ce projet est défini comme privé car il contient des données sensibles concernant l'entreprise.

## Activité-type 1

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

CP 2 ▶

Interface avec des données de démonstration (Réaliser une interface statique et adaptable)

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

Conception du frontend avec des données codées en dur / de démonstration :

Après avoir terminé les wireframes et les prototypes, j'ai procédé au développement du frontend de l'application. Voici des exemples de code provenant du composant Header du projet Next.js.

Au début du fichier, j'importe toutes les bibliothèques et utilitaires nécessaires à ce composant. Puisque le composant utilise des hooks React, il doit être rendu côté client. Par conséquent, la première ligne de code définit 'use client'. Les imports restants incluent les hooks mentionnés, l'objet React de la bibliothèque React, des icônes de lucide-react, le composant Image de Next.js, et mon composant SearchComponent personnalisé.

Dans la définition du composant, j'initialise deux variables d'état pour stocker des informations temporaires basées sur les interactions de l'utilisateur. Pour optimiser les performances, j'utilise le hook useEffect pour m'assurer que React ne réaffiche pas inutilement la page lorsque les valeurs de ces variables changent, ce qui sinon provoquerait la réinitialisation des variables. J'inclus les variables dans le tableau de dépendances du hook useEffect pour suivre les changements apportés à ces variables spécifiques.

Lors du rendu du composant, je crée un élément <div> d'enveloppement pour l'ensemble du composant, qui utilise display: flex avec une direction de colonne et s'étend sur toute la largeur de la fenêtre. Ensuite, je définis un élément pour contenir la partie supérieure du menu. Cet élément utilise également display: flex, mais avec une direction de ligne pour ses éléments enfants. Le contenu est justifié avec space-between, garantissant un espacement uniforme entre les éléments, et les éléments sont centrés à l'aide de align-items: center.

Lorsque la largeur de l'écran atteint le point de rupture pour les grands écrans (@media (min-width: 1024px)), la direction passe à la colonne, empilant les éléments verticalement. Par défaut, pour les écrans avec une largeur maximale inférieure à 1024px, le menu de haut niveau est masqué. Au point de rupture 'lg', l'élément revient à display: flex, la modification principale étant sa hauteur, qui est définie à 50px.

Le thème principal du composant est l'utilisation de display: flex, qui gère efficacement l'alignement des éléments dans une seule dimension, soit horizontalement, soit verticalement.

La gestion des points de rupture est effectuée à l'aide des valeurs standardisées fournies par Tailwind CSS. Il est important de noter que Tailwind CSS adopte une approche "mobile-first", ce qui signifie que ses requêtes média sont structurées comme @media (min-width: ...px). Par conséquent, le processus de conception commence avec les écrans plus petits et s'améliore progressivement pour les écrans plus grands.

Un autre aspect notable est le menu mobile, qui contraste avec le menu principal. Sur les écrans plus petits, le menu principal est masqué. À sa place, une icône de hamburger est affichée. Lors de l'interaction, cette icône déclenche le menu mobile, qui glisse dans la vue depuis la gauche avec une transition fluide. Le menu mobile contient les mêmes catégories que le menu principal. L'élément d'enveloppement du menu mobile empêche le

## DOSSIER PROFESSIONNEL (DP)

défilement vertical du site pour éviter toute confusion de l'utilisateur.

Sur les écrans plus grands, le menu mobile et l'icône de hamburger sont masqués. À la place, le menu principal est affiché, présentant des effets fluides lors des interactions avec la souris, telles que le survol, le clic et la sortie de la zone.

Exemples de code :

## DOSSIER PROFESSIONNEL (DP)

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé VSCode, Next.js et React

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Petsquare GR*

Chantier, atelier, service ► Projet professionnel réalisé en cours de stage.

Période d'exercice ► Du : *16/12/2024* au : *17/01/2025*

## 5. Informations complémentaires (facultatif)

Le dépôt de ce projet est défini comme privé car il contient des données sensibles concernant l'entreprise.

## Activité-type 1 Développer la partie front-end d'un e-shop

CP 3 ► Données dynamiques (E-commerce)

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

Pour permettre à mon frontend (Next.js) de récupérer des données depuis le backend (Directus), j'ai apporté plusieurs ajustements pour accommoder l'API de Directus. Tout d'abord, j'ai ajouté les informations nécessaires concernant mon frontend au fichier `docker-compose.yml`. Ensuite, j'ai installé le package npm Directus SDK en utilisant la commande `npm install @directus/sdk`.

Après l'installation, j'ai créé un nouveau fichier nommé `directus.ts` dans le dossier des bibliothèques de mon projet frontend. Dans ce fichier, j'ai importé les méthodes nécessaires et défini une interface pour les catégories, car il s'agissait des premières données intégrées au site. J'ai également créé une interface de schéma pour décrire la structure de la base de données, en me concentrant sur la collection des catégories, car c'était la seule table utilisée.

Ensuite, j'ai instancié Directus et développé des fonctions d'aide pour récupérer les données nécessaires depuis la base de données. Enfin, j'ai exporté l'instance de Directus pour une utilisation dans l'ensemble du projet.

Pour afficher les catégories dans mon menu, j'ai importé la fonction d'aide `getCategories` et l'interface `Category` depuis le fichier `directus.ts` dans mon composant `Header`. J'ai ensuite créé une fonction pour récupérer les données en utilisant la fonction `getCategories` et je l'ai invoquée dans un hook `useEffect` pour m'assurer que la requête est exécutée une seule fois. J'ai défini une nouvelle variable pour stocker les données retournées et créé une table de hachage pour mes icônes, me permettant d'afficher l'icône correcte pour chaque catégorie correspondante.

Enfin, dans la section JSX où les éléments du menu sont rendus, j'ai utilisé une fonction de mappage pour itérer sur les catégories et retourner leur contenu.

Exemples de code :

```

{ /* Main menu */}
<div className="hidden lg:flex w-full bg-[#D93A39] text-white justify-between items-center px-12
h-[50px] text-xl">
  <div
    className="cursor-pointer flex items-center group relative"
    key="home-main-menu-item"
  >
    <HomeIcon className="mr-2" />
    <p>Home</p>
    <span className="absolute -bottom-1 left-1/2 h-px w-0 bg-white transition-all duration-300
group-hover:left-0 group-hover:w-full"></span>
  </div>
  {categories.map((category) => (
    <div
      key={category.id}
      className="cursor-pointer flex items-center group relative"
    >
      { /* Dynamically select the icon */}
      {iconMap[category.name] || <Building className="mr-2" />}
      <p>{category.name}</p>
      <span className="absolute -bottom-1 left-1/2 h-px w-0 bg-white transition-all duration-300
group-hover:left-0 group-hover:w-full"></span>
    </div>
  ))}
</div>

```



```
import { createDirectus, rest, readItems } from "@directus/sdk";

// Category interface
export interface Category {
  id: string;
  name: string;
  slug: string;
  parent_category?: string;
  description?: string;
  icon?: string;
  view_type: string;
}

// Collections schema
interface Schema {
  categories: Category[];
}

// Directus instance
const client = createDirectus<Schema>(
  process.env.NEXT_PUBLIC_DIRECTUS_URL!
).with(rest());

// Helper function to get public URL for assets
export const getDirectusImageUrl = (imageId: string): string => {
  if (!imageId) return "";
  return `${process.env.NEXT_PUBLIC_DIRECTUS_URL}/assets/${imageId}`;
};

// Categories helper functions
export const getCategories = async () => {
  try {
    const response = await client.request(
      readItems("categories", {
        sort: ["name"],
      })
    );
    return response || [];
  } catch (error) {
    console.error("Error fetching categories:", error);
    return [];
  }
};

export default client;
```

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé VSCode, Next.js et React

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Petsquare GR*

Chantier, atelier, service ► Projet professionnel réalisé en cours de stage.

Période d'exercice ► Du : *17/02/2025* au : *11/03/2025*

## 5. Informations complémentaires (facultatif)

Le dépôt de ce projet est défini comme privé car il contient des données sensibles concernant l'entreprise.

## Activité-type 2 Développer la partie back-end d'un e-shop

CP 4 ► Diagramme pour la base de données

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

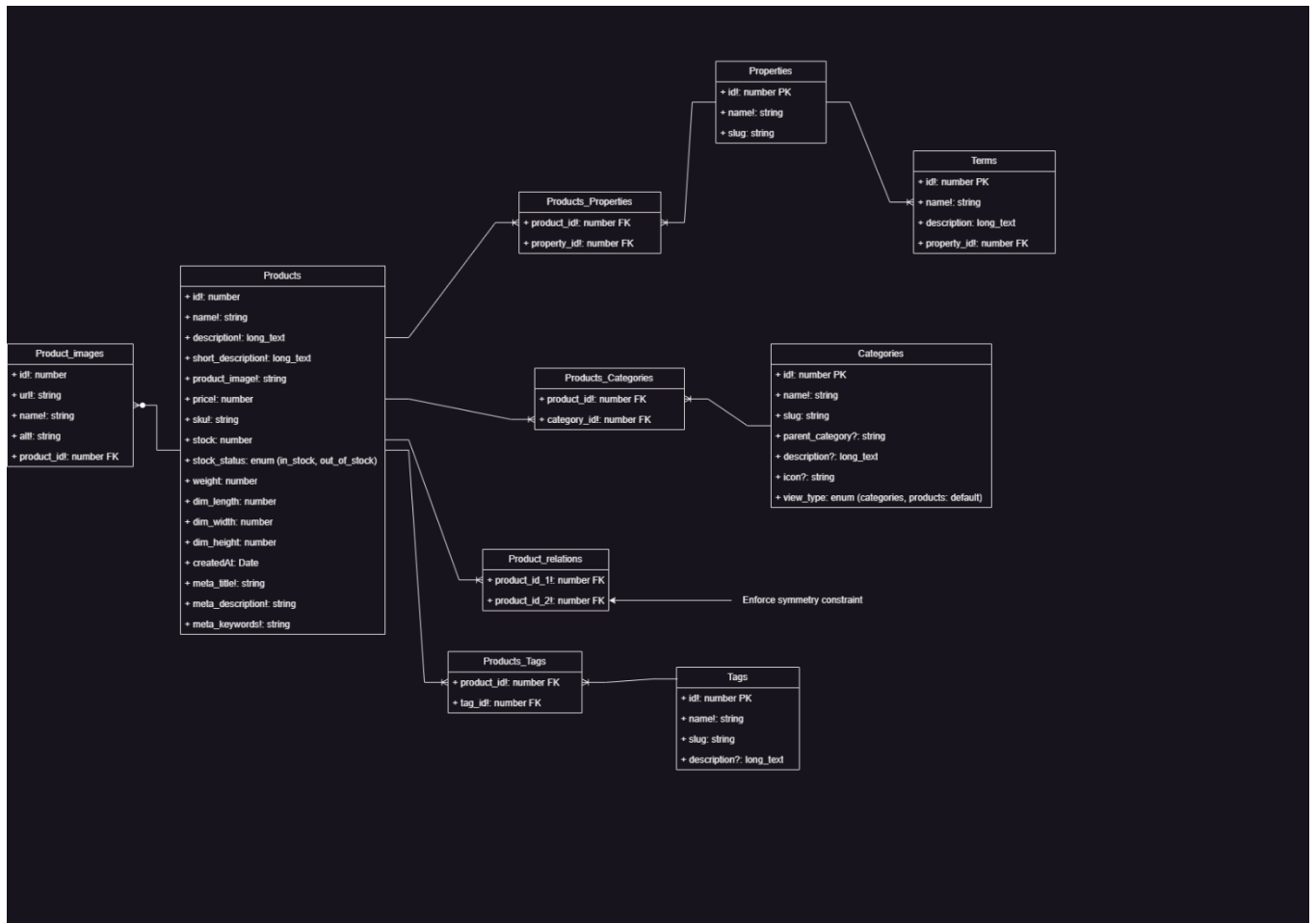
J'ai créé le diagramme de la base de données en utilisant Draw.io. La base de données comprend plusieurs tables pertinentes pour le commerce électronique, notamment une table de produits, une table de catégories, une table d'étiquettes et des tables de connexion associées. Il est important de noter que les relations plusieurs-à-plusieurs nécessitent des tables de connexion, tandis que les relations un-à-plusieurs ne nécessitent que des clés étrangères.

Pour les produits, une galerie d'images de produits est nécessaire. J'ai donc créé une table supplémentaire connectée à la table des produits. Cette table contient les URL des images, les noms, le texte alternatif pour le produit correspondant et la clé étrangère reliant chaque image au bon produit.

La table des produits comprend des champs standard tels que le nom, l'ID, la description, le prix et le SKU. Elle contient également des champs pour le numéro de stock et le statut du stock pour une meilleure gestion des inventaires, le poids et les dimensions pour la gestion de l'expédition, ainsi que des métadonnées pour aider aux recherches sur le site et sur les plateformes, telles que Google Search.

Les tables d'étiquettes et de catégories sont utilisées pour une meilleure gestion des produits et pour faciliter les recherches de produits.

# DOSSIER PROFESSIONNEL (DP)



## 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé Draw.io

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Petsquare GR*

Chantier, atelier, service ► Projet professionnel réalisé en cours de stage.

Période d'exercice ► Du : 20/01/2025 au : 27/01/2025

## 5. Informations complémentaires (facultatif)

Le dépôt de ce projet est défini comme privé car il contient des données sensibles concernant l'entreprise.

## Activité-type 2 Développer la partie back-end d'un e-shop

CP 5 ► Création de la base de données

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

Pour le backend, j'ai choisi Directus, un système backend bien connu et convivial basé sur Node.js. J'ai suivi le guide d'auto-hébergement fourni dans la documentation de Directus pour l'installation.

J'ai commencé par télécharger Docker, car le projet devait être conteneurisé pour fonctionner. Le processus d'installation était simple, ne nécessitant que quelques clics pour être complété.

Après avoir installé Docker, j'ai créé un dossier Directus pour mon projet, qui comprenait trois sous-dossiers : base de données, téléchargements et extensions. Dans le dossier principal du projet, j'ai créé un fichier nommé docker-compose.yml contenant les instructions nécessaires pour Docker. Enfin, j'ai exécuté le conteneur Docker en naviguant vers le dossier contenant mes fichiers et en exécutant la commande docker compose up.

Une fois le téléchargement terminé, j'ai pu accéder à mon backend via <http://localhost:8055> ou <http://127.0.0.1:8055>.

### 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé Directus et Docker

### 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

### 4. Contexte

Nom de l'entreprise, organisme ou association ► *Petsquare GR*

Chantier, atelier, service ► Projet professionnel réalisé en cours de stage.

Période d'exercice ► Du : 27/01/2025 au : 03/02/2025

## 5. Informations complémentaires (facultatif)

Le dépôt de ce projet est défini comme privé car il contient des données sensibles concernant l'entreprise.

## Activité-type 2 Développer la partie back-end d'un e-shop

CP 6 ► Création des tables et des relations

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

Après avoir configuré le backend Directus dans Docker, j'ai procédé à la création des tables de la base de données (appelées collections dans Directus) et à la définition de leurs relations. Pour lancer Directus, le conteneur Docker doit être en cours d'exécution. J'utilise généralement l'interface de Docker pour démarrer mon conteneur, bien que d'autres méthodes soient disponibles comme mentionné précédemment.

Une fois le conteneur Docker en cours d'exécution, j'accède à l'interface de Directus en me rendant sur l'URL locale et en entrant mes identifiants. Après une connexion réussie, je crée une nouvelle table en me rendant dans la section "Paramètres" en bas du menu de gauche, en sélectionnant "Modèle de données" et en cliquant sur l'icône "+" pour créer une nouvelle collection.

Dans la section suivante, je spécifie un nom pour la collection, je détermine si elle doit être traitée comme un objet unique et je définis la clé primaire. Ensuite, j'ajoute tous les champs supplémentaires nécessaires en accédant à l'onglet "Champs optionnels" et en cliquant sur l'icône de validation pour confirmer mes sélections.

Ensuite, je sélectionne ma nouvelle collection et je clique sur le bouton "Créer un champ" pour ajouter tous les champs nécessaires. Cette étape est cruciale car c'est ici que les relations entre les tables sont définies. Je choisis le type de champ (par exemple, un champ de saisie simple), je remplis les informations requises et j'enregistre les modifications.

Ma nouvelle collection est maintenant prête à être utilisée. Si des modifications supplémentaires sont nécessaires, je peux répéter le processus, modifier les champs existants ou ajouter des informations complémentaires telles qu'un tri automatique basé sur un champ spécifique.

Pour ajouter du contenu à ma nouvelle collection, je me rends dans l'onglet "Contenu" du menu de gauche, je

# DOSSIER PROFESSIONNEL (DP)

sélectionne ma collection et je clique sur le bouton "Créer un élément". Ensuite, je définis le statut de l'élément, je remplis les champs et j'enregistre l'entrée en cliquant sur le bouton de sauvegarde.

Ce flux de travail était typique pendant le développement de mon projet de stage.

## 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé Directus et Docker

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *Petsquare GR*

Chantier, atelier, service ► Projet professionnel réalisé en cours de stage.

Période d'exercice ► Du : *03/02/2025* au : *17/02/2025*

## 5. Informations complémentaires (facultatif)

Le dépôt de ce projet est défini comme privé car il contient des données sensibles concernant l'entreprise.



## Activité-type 3

CP 7 ▶

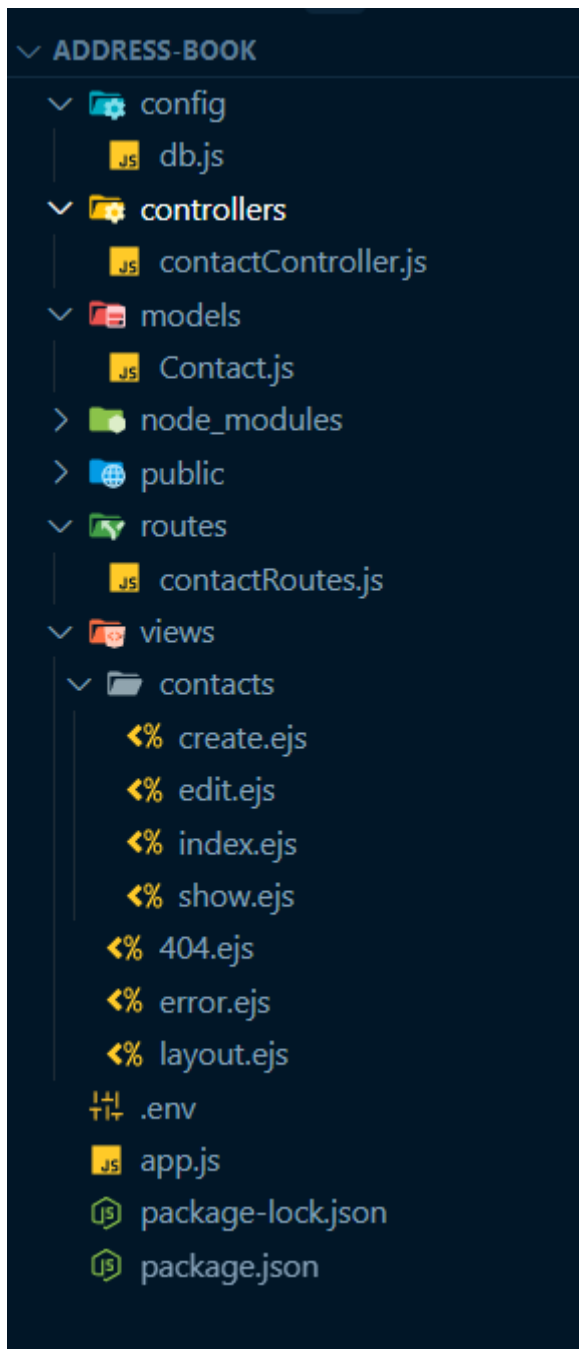
Développer la partie back-end avec NoSQL

Développement d'une application CRUD NoSQL

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

Lors de notre formation, nous avons appris à créer des bases de données NoSQL en utilisant MongoDB.

J'ai développé un petit projet de carnet d'adresses en utilisant Node.js, Express.js et Mongoose pour démontrer une application "CRUD" utilisant NoSQL. Pour ce projet, j'ai suivi une architecture MVC.



J'ai commencé par initialiser un nouveau projet Node.js avec `npm init -y`. Ensuite, j'ai créé les dossiers nécessaires : `config` pour la configuration de la base de données, `controllers`, `models`, `routes`, `views` et un dossier `public` pour la partie vue. J'ai également créé des sous-dossiers pour les contacts à l'intérieur du dossier `views`, un dossier `css` et un dossier `js` pour le dossier `public`, puis j'ai créé les fichiers nécessaires, en commençant par `.env`.

J'ai défini deux variables, `PORT` et `MONGO_URI`, nécessaires pour la configuration de la base de données.

Ensuite, dans le dossier config, j'ai créé un fichier db.js contenant le code pour exécuter Mongoose et démarrer la connexion à MongoDB.

```
const mongoose = require("mongoose");

const connectDB = async () => {
  try {
    const conn = await mongoose.connect(
      process.env.MONGO_URI || "mongodb://localhost:27017/address_book",
      {
        useNewUrlParser: true,
        useUnifiedTopology: true,
      }
    );
    console.log(`MongoDB Connected: ${conn.connection.host}`);
  } catch (err) {
    console.error(`Error: ${err.message}`);
    process.exit(1);
  }
};

module.exports = connectDB;
```

Dans le dossier models, j'ai créé un modèle Contact.js qui définit la structure de la collection.

```
const mongoose = require("mongoose");

const ContactSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
    trim: true,
  },
  email: {
    type: String,
    required: false,
    trim: true,
  },
  phone: {
    type: String,
    required: false,
    trim: true,
  },
  createdAt: {
    type: Date,
    default: Date.now,
  },
});

module.exports = mongoose.model("Contact", ContactSchema);
```

Dans le dossier controllers, j'ai ajouté le contrôleur `contactController.js` qui gère les opérations CRUD.

```
// @desc    Get all contacts
// @route   GET /
exports.getContacts = async (req, res) => {
  try {
    const contacts = await Contact.find().sort({ createdAt: -1 });
    res.render("contacts/index", {
      title: "Address Book",
      contacts,
    });
  } catch (err) {
    console.error(err);
    res.status(500).render("error", {
      title: "Error",
      message: "Server Error",
    });
  }
};
```

Par exemple, ma fonction `getContacts` récupère toutes les données de la collection `Contact`, les trie par le champ `createdAt` en ordre décroissant, puis rend le résultat à la vue `index` (détaillée plus tard). En cas d'erreur, celle-ci est consignée dans la console.

J'ai ensuite créé le fichier `contactRoutes.js` dans le dossier `routes` avec les méthodes de requête appropriées (`get`, `post`, `put`, `delete`).

```
const express = require("express");
const router = express.Router();
const contactController = require("../controllers/contactController");
router.get("/", contactController.getContacts);
router.get("/contacts/new", contactController.getAddContactForm);
router.post("/contacts", contactController.addContact);
router.get("/contacts/:id", contactController.getContact);
router.get("/contacts/:id/edit", contactController.getEditContactForm);
router.put("/contacts/:id", contactController.updateContact);
router.delete("/contacts/:id", contactController.deleteContact);

module.exports = router;
```

Dans le dossier views, j'ai créé une page 404, une page d'erreur et défini la mise en page générale. Dans le sous-dossier contacts, j'ai ajouté les fichiers create, edit, index et show. Le contenu de la page change en fonction du fichier chargé lors de la visite d'une vue différente.

```
<h2>Contacts</h2>
<% if (contacts.length === 0) { %>
<p>No contacts found. <a href="/contacts/new">Add a contact</a>.</p>
<% } else { %>
<div class="contacts-list">
  <% contacts.forEach(contact => { %>
    <div class="contact-card">
      <h3><a href="/contacts/<%= contact._id %>"><%= contact.name %></a></h3>
      <% if (contact.email) { %>
      <p><strong>Email:</strong> <%= contact.email %></p>
      <% } %> <% if (contact.phone) { %>
      <p><strong>Phone:</strong> <%= contact.phone %></p>
      <% } %>
      <div class="contact-actions">
        <a href="/contacts/<%= contact._id %>/edit" class="edit-btn">Edit</a>
        <form action="/contacts/<%= contact._id %>?_method=DELETE" method="POST">
          <button
            type="submit"
            class="delete-btn"
            onclick="return confirm('Are you sure you want to delete this contact?')"
          >
            Delete
          </button>
        </form>
      </div>
    </div>
    <% }) %>
  </div>
<% } %>
```

Par exemple, dans l'index : si aucun résultat n'est retourné par la base de données, le premier élément p est affiché. Sinon, les éléments sont listés selon la structure définie.

Enfin, dans mon fichier app.js, qui est le point d'entrée de l'application, j'importe toutes les bibliothèques nécessaires, les utilise dans le code principal, implémente les routes et démarre le serveur en écoutant le port défini dans le fichier .env.

Pour exécuter l'application, j'ai installé MongoDB, Node.js, puis dans le dossier principal du projet, j'ai exécuté la commande npm i (ou npm install) pour installer toutes les bibliothèques Node requises. Ensuite, j'ai lancé l'application avec npm start et visité l'URL <http://localhost:3000/>.

# DOSSIER PROFESSIONNEL (DP)

## Address Book

[Home](#) [Add Contact](#)

### Contacts

No contacts found. [Add a contact](#).

© 2025 Address Book

## Address Book

[Home](#) [Add Contact](#)

### Add New Contact

Name \*

Email

Phone

Add Contact

Cancel

## Address Book

[Home](#) [Add Contact](#)

### Contacts

Test testington

Email: testington@test.com

Phone: 1234567890

Edit

Delete

## Address Book

[Home](#) [Add Contact](#)

### Edit Contact

Name \*

Test testington

Email

testington@test.com

Phone

1234567890

Update Contact

Cancel

## 2. Précisez les moyens utilisés :

Dans le cadre de ce projet, j'ai utilisé Node.js, Express.js, VSCode, MongoDB, mongoose

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour ce projet



# DOSSIER PROFESSIONNEL (DP)

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *M2i Scribtel*

Chantier, atelier, service ► Projet personnel réalisé en cours de la formation.

Période d'exercice ► Du : *12/03/2025* au : *14/02/2025*

## 5. Informations complémentaires (facultatif)

## 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.

---

## DOSSIER PROFESSIONNEL (DP)

---

---

### Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Sotirios NATSIOS ,  
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 Grenoble le 15/03/2025

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)*