

# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ DENIVET  
*Nom d'usage* ▶ DENIVET  
*Prénom* ▶ Julia  
*Adresse* ▶ 7 rue Colbert - 13001 Marseille

## Titre professionnel visé

Concepteur(trice) Développeur(se) Informatique

### 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.

*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>Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>5</b>
- La Boutique de Luc React Native	p. p.	5
- Le Tic Tac Toe	p p.	11
<b>Concevoir et développer la persistance des données en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>14</b>
- Expense Management SystemI	p. p.	14
- La Boutique de Luc API Restful	p. p.	18
<b>Concevoir et développer une application multicouches répartie en intégrant les recommandations de sécurité</b>	<b>p.</b>	<b>21</b>
- Le Port de Marseille	p. p.	21
- La Boutique de Luc React Native	p. p.	27
- Le Déploiement avec Hopps Group	p p.	31
<b>Titres, diplômes, CQP, attestations de formation</b> <i>(facultatif)</i>	<b>p.</b>	<b>35</b>
<b>Déclaration sur l'honneur</b>	<b>p.</b>	<b>36</b>
<b>Documents illustrant la pratique professionnelle</b> <i>(facultatif)</i>	<b>p.</b>	<b>37</b>
<b>Annexes</b> <i>(Si le RC le prévoit)</i>	<b>p.</b>	<b>38</b>

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - La Boutique de Luc

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

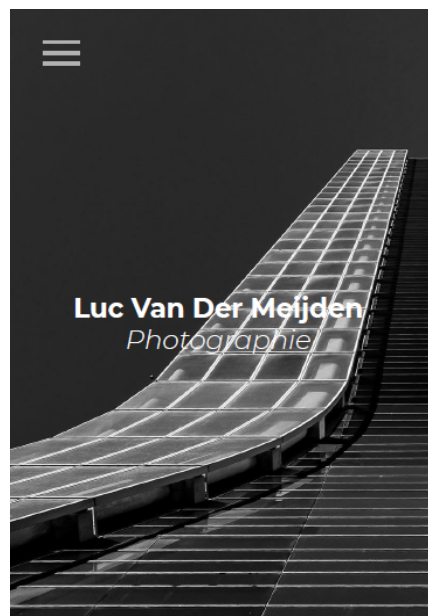
La Boutique de Luc est une application mobile qui a pour but de permettre à Luc de gagner beaucoup plus de visibilité et de pouvoir vendre les photos qu'il a fait. Après plusieurs réunions d'équipe, nous nous sommes tous mis d'accord sur l'arborescence du site.

L'arborescence du site est faite comme ceci :

- une page d'accueil avec un visuel sur les favoris
- une navigation qui s'ouvre avec un slide vers la droite et vers la gauche pour fermer. Permet donc de changer facilement de page.
- une page de connexion et une page d'inscription
- une boutique séparés en plusieurs catégories
- un panier
- une page de modification de profil avec en plus un visuel sur l'historique des commandes
- une page administration accessible uniquement si on a le bon id droit.

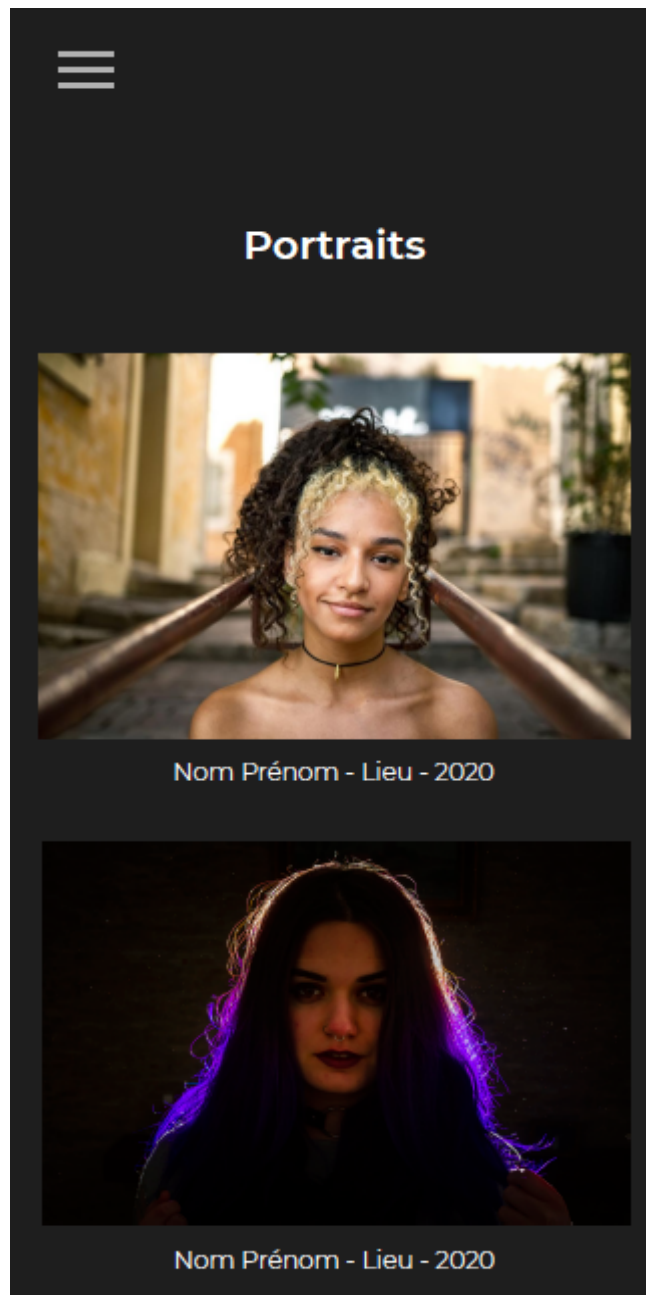
Lorsque l'arborescence a été mise en place, on a pu commencer à réfléchir à la maquette du site, à son identité. Nous étions tous d'accord pour partir sur une thème minimaliste noir et blanc car la présence de photos puisque c'est le but de notre projet cela va déjà mettre un certain design donc le thème minimaliste était, je pense le plus approprié pour cette application mobile.

Voici quelques screenshots de notre maquette avec les pages accueil et boutique (maquette complète en annexe) :



page d'accueil

Pour la maquette, nous avons utilisé le logiciel sketch qui est un éditeur de graphiques vectoriels. Une différence essentielle entre Sketch et d'autres éditeurs de graphiques vectoriels est que Sketch n'inclut pas de fonctionnalités de conception d'impression. Sketch est principalement utilisé pour la conception d'UI et d'UX d'applications mobiles et web.



Concernant la partie front-end, elle a été développée en React Native. React Native est un framework Javascript permettant de développer des applications mobiles pour Android, IOS et UWP (*Universal Windows Platform*). C'est un framework open source créé par facebook. Ce framework permet aux développeurs d'utiliser React avec les fonctionnalités native de ces plateformes.

La différence avec React est que React Native utilise principalement des balises spécifiques. Lorsqu'on commence le render d'un composant on ouvre une balise <View>. C'est la balise fondamentale pour commencer à développer une interface utilisateur. La view est un conteneur qui peut prendre en charge la mise en page (flex, style et une certaine gestion tactile). La view va mapper directement à l'équivalent de la vue Native sur la plateforme sur laquelle il s'exécute. View est un composant qui peut être imbriqué dans d'autres vues et il peut avoir de 0 à n enfants de tout type.

```
return (  
  <View style={styles.loginContainer}>  
    <Text>Sign Up</Text>  
    <TextInput  
      name="mail"  
      placeholder="Email Address"  
      style={styles.textInput}  
      onChangeText={this.handleForm.bind(this)}  
      value={this.state.values.mail}  
      keyboardType="email-address"  
    />  
    <TextInput  
      name="password"  
      placeholder="Password"  
      style={styles.textInput}  
      onChangeText={this.handleForm.bind(this)}  
      value={this.state.values.password} You,  
    />  
    <Button  
      onPress={handleSubmit}  
      title="LOGIN"  
    />  
  </View>  
)
```

```
import { StyleSheet, Text, View, TextInput, Button } from "react-native";
```

Prenons l'exemple du formulaire de connexion de boutique. Le formulaire a deux inputs : un pour le mail et un pour le mot de passe. Pour utiliser un input en React Native, il faut importer ses balises. Ici,

j'utilise une balise Text Input donc je pense donc à bien importer le TextInput de React Native.

Regardons maintenant les paramètres pris en compte par la balise TextInput :

- name = le nom de ma balise (exemple : mail)
- placeholder = permet de renseigner un texte dans un champ de formulaire. C'est une valeur qui s'efface dès que l'utilisateur active le champ de formulaire
- style = là où on va attribuer les classes css
- onChangeText = événements pour lire l'entrée de l'utilisateur.
- value = la récupération de la valeur de l'input insérée par l'utilisateur.
- keyboardType = le type de clavier souhaité (exemple : email keyboard)

Pour finir, le formulaire comporte aussi une balise Button qui doit aussi être importée de react-native.

Elle prendra en paramètres deux éléments :

- onPress = Méthode à appeler lorsque l'utilisateur presse le bouton.
- Title = le nom du bouton (exemple : Login)

La partie API de l'application a été faite avec CodeIgniter, un framework PHP. On va parler d'une entité assez représentative du site : les produits. Pour l'API, nous avons utilisé une API RESTful, utilisant le CRUD pour toutes les entités.

```
public function index()
{
    try
    {
        $data=$this->Models_Produits->getProduits();
    }
    catch (Exception $e)
    {
        echo 'Exception reçue : ', $e->getMessage(), "\n";
    }
}
```

Products Controller

```
public function getProduits()
{
    try
    {
        if(isset($_GET['id']))
        {
            $query = $this->db->query("SELECT * FROM produit WHERE id='".$_GET['id']."'");
            $data = $query->row_array();
            echo json_encode($data);
        }
    }
    catch (Exception $e)
    {
        http_response_code('401');
        echo json_encode(array( "status" => false, "message" => $e->getMessage()));exit;
    }
}
```

Products Model

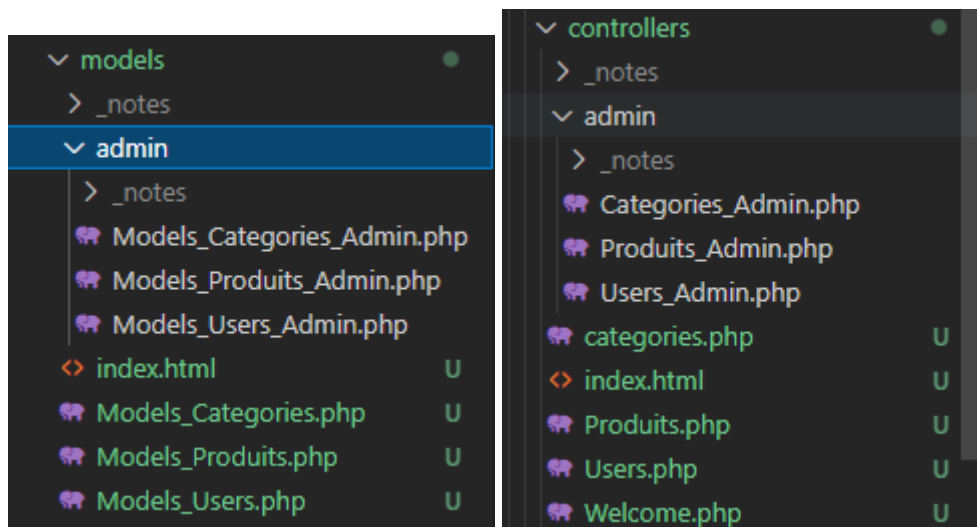
Dans notre application, nous avons préféré utiliser des modèles qui contiennent les logiques métiers.



## DOSSIER PROFESSIONNEL (DP)

Dans le controller, on fait appel à la méthode `GetProducts()` du modèle pour récupérer les résultats de la requête `"SELECT * FROM produit WHERE id='". $_GET['id'] .'"` et on stocke ça dans la variable `$data` qui sera ensuite envoyé à notre View : React Native.

La boutique de Luc contient trois entités principales : les produits, les catégories et les utilisateurs.



Pour ensuite récupérer la liste des produits, il faudra utiliser le bundle Axios de React pour faire appel à l'api et à la méthode `GetProducts()`.

### 2. Précisez les moyens utilisés :

#### Langages utilisés :

- PHP
- SQL
- Javascript
- 

#### Frameworks utilisés :

- CodeIgniter
- ReactNative
- Expo
- 

#### IDE utilisés :

- Visual Studio Code

#### Pour la Maquette :

- Sketch

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec mes camarades de cursus : Luc Van Der Meijden et Justine Lesur.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ LaPlateforme

Chantier, atelier, service ▶ Web Developer

Période d'exercice ▶ Du : 14/12/2020 au : 30/07/2021

## 5. Informations complémentaires *(facultatif)*

## Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

---

## Exemple n°2 - Le Tic Tac Toe

---

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

Le projet Tic Tac Toe est un projet développé en python. Le Tic Tac Toe est un jeu d'alignement à deux joueurs, une variante simplifiée du jeu du Morpion (jeu qui se joue sur une grille qui n'a pas de limites). On trouve une version de ce jeu dans l'empire romain.

Déroulement du jeu sur papier :

- une feuilles blanches et des crayons
- Tracer une grille carrée de 3 lignes et 3 colonnes

Les deux joueurs doivent remplir, à tour de rôle, une case de la grille avec le symbole qui leur est attribué, généralement X ou O. Celui qui aligne, le premier, 3 symboles identiques, a gagné. L'alignement peut être, horizontal, vertical ou en diagonale. L'issue du jeu peut être la victoire de l'un des deux joueurs ou une partie nulle.

Le choix du langage informatique pour développer ce jeu est python. Python est un langage de programmation le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science . Aujourd'hui python est l'un des langages les plus utilisés dans le domaine du développement de logiciels, de gestion d'infrastructure et d'analyse de données. Il s'agit d'un élément moteur de l'explosion du Big Data.

Il s'agit d'un langage de programmation interprété, qui ne nécessite donc pas d'être compilé pour fonctionner. Un programme " interpréteur " permet d'exécuter le code Python sur n'importe quel ordinateur. Ceci permet de voir rapidement les résultats d'un changement dans le code. En revanche, ceci rend ce langage plus lent qu'un langage compilé comme le C.

Les principaux avantages de python sont :

- facile à apprendre et à utiliser
- il fonctionne sur tous les principaux systèmes d'exploitation
- créer des logiciels de qualité professionnelle

```
Player X turn. Which box? : 7

  X |   | O
  ---|---|
  X |   | O
  ---|---|
  X |   |

Player X has won the game!!

-----
                        SCOREBOARD
-----
      Juli          1
      julia         0
-----

Turn to choose for julia
Enter 1 for X
Enter 2 for O
Enter 3 to Quit
█
```

Voici l'aperçu du jeu dans le terminal, on retrouve bien la grille de 3 par 3 et on retrouve aussi les X et O qui représentent les pions. Dans cette partie, c'est le joueur Juli qui a gagné car elle a fait une des combinaisons gagnante qui est le [1;4;7].

Pour vérifier le gain ou non de la personne toutes les combinaisons gagnantes possible sont enregistrés dans une variable :

```
# All possible winning combinations
soln = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [1, 4, 7], [2, 5, 8], [3, 6, 9], [1, 5, 9], [3, 5, 7]]
```

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

Au début de la partie, chacun des joueurs renseigne son prénom avec des inputs et choisissent leur pions X ou O. A chaque fin de partie, le score est gardé en mémoire dans le scoreboard.

Lorsque la manche se finit, la personne qui l'a gagné rapporte 1 point. Si les joueurs décident d'arrêter ils peuvent demander de quitter la partie et le scoreboard final s'affichera.

## 2. Précisez les moyens utilisés :

Langages utilisés :

- Python

IDE utilisés :

- Visual Studio Code

## 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé seule.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ LaPlateforme\_

Chantier, atelier, service ▶ Web Developer

Période d'exercice ▶ Du : 01/04/2021 au : 02/04/2020

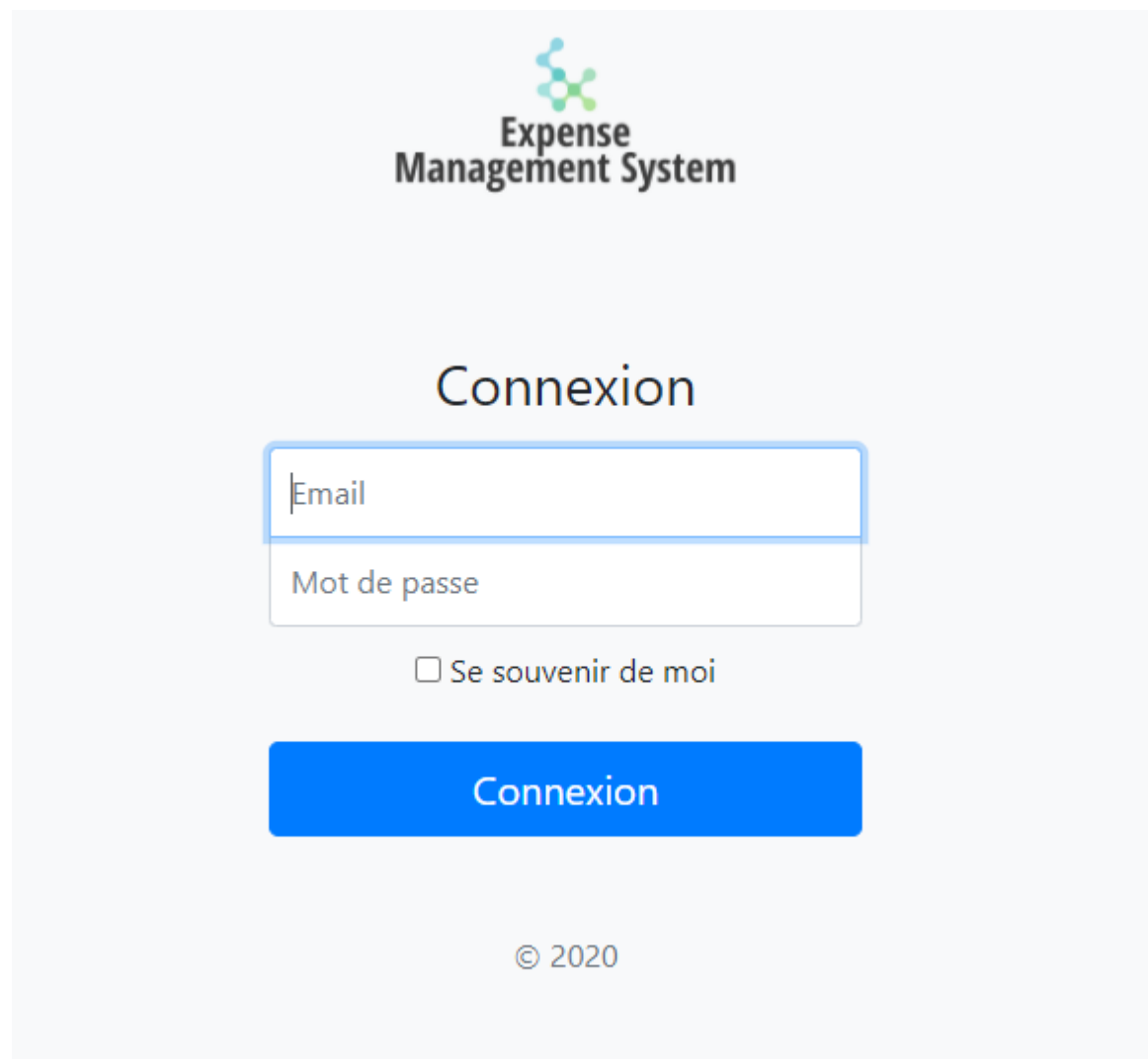
## 5. Informations complémentaires (facultatif)

## Activité-type 2

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 - Expense Management System

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



The screenshot shows the login interface of the 'Expense Management System'. At the top, there is a logo consisting of a stylized green and blue network icon above the text 'Expense Management System'. Below the logo, the word 'Connexion' is centered in a large, black, sans-serif font. Underneath, there are two input fields: the first is labeled 'Email' and the second is labeled 'Mot de passe'. Below these fields is a checkbox with the text 'Se souvenir de moi'. A large blue button with the text 'Connexion' in white is positioned below the checkbox. At the bottom of the page, there is a copyright notice '© 2020'.

Le projet ems (Expense Management System) est une application de notes de frais permettant aux salariés de pouvoir saisir une demande de remboursement pour une note de frais. Pour cela, il devra uploader un justificatif (photo, PDF) et saisir le montant de la note. Il doit renseigner son nom et prénom et s'il le souhaite il peut ajouter un commentaire accompagné de la note de frais.

Voici le chemin salariés, ce qu'il peut faire sur Expense Management System :

- Le salarié a mangé au restaurant lors de son voyage d'affaire à Singapour
- il photographie la facture et l'upload sur ems
- il écrit son nom prénom
- il recopie le montant de la note et éventuellement il ajoute un commentaire.
- il envoie sa demande

Du côté RH :

- Une personne du service RH se connecte à l'outil via un accès spécial
- La liste de demande de remboursement est visible
- Pour chaque demande, le service RH peut voir le salarié concerné, le montant et le justificatif
- La note de frais peut être acceptée ou refusée.

L'application a été développée avec deux Framework :

- Symfony pour la partie back-end
- React pour la partie front-end

```
getUsers() {  
  axios.get(  
    'http://localhost:8000/api/users',  
    { headers: { "Content-Type": "application/json" } }  
  ).then(users => {  
    this.setState({ users: users.data, loading: false })  
  })  
}
```

GetUsers() React

```
public function getUsers() {  
    $users = $this->getDoctrine()->getRepository(User::class)->findAll();  
  
    $response = new Response();  
  
    $response->headers->set('Content-Type', 'application/json');  
    $response->headers->set('Access-Control-Allow-Origin', '*');  
  
    $json = json_encode($users);  
    $response->setContent($json);  
  
    return $response;  
}
```

GetUsers() Symfony

Pour cet exemple, nous allons plus nous concentrer sur l'échange de données entre le front et le back. Pour React, l'utilisation des outils ajax pour communiquer avec l'api pour la récupération de data. Pour cela, on va utiliser Axios un bundle de react qui est un client HTTP léger basé sur le service \$http. Lorsqu'on fait appel à Axios on a besoin de npm pour l'installer en utilisant la commande suivant :

```
npm i axios
```

Après l'installation, il faut importer Axios dans le composant où on souhaite l'utiliser. Axios a quatres types d'actions possibles : le get, le post et le put. Lorsque l'on choisit la méthode on doit renseigner le lien qui cible la méthode que l'on veut appeler. Ici, on cible la méthode index de la classe User qui est la récupération des utilisateurs.

Dans le controller User de Symfony, nous avons créé la fonction getUsers(). D'aller récupérer toutes les informations de l'entité users en utilisant le findAll() dans l'entité User. Ces résultats sont ensuite convertis en Json et envoyés en réponse. Cette réponse va être ensuite récupérée par Axios pour les afficher sur la page.

#	Prénom - Nom	Service
1	julia denivet	
2	zzfdsc edfsc	
3	Camille Denivet	pole web
4	Julia Denivet	pole web
5	Julie Denivet	pole back office

Liste des utilisateurs du côté RH



# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

Julia Denivet

julia.denivet@colisprive.com

Julia

Denivet

pole web

Salarié

Modifier

Supprimer

Pop-up pour la modification ou la suppression du user.

## 2. Précisez les moyens utilisés :

### Langages utilisés :

- PHP
- JavaScript
- HTML/CSS

### Framework utilisés :

- React
- Symfony

### IDE utilisés :

- Visual Studio Code

### Test API :

- Postman

## 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec Etienne Januel et Olivier Crozet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ LaPlateforme\_

Chantier, atelier, service ▶ Web Developer

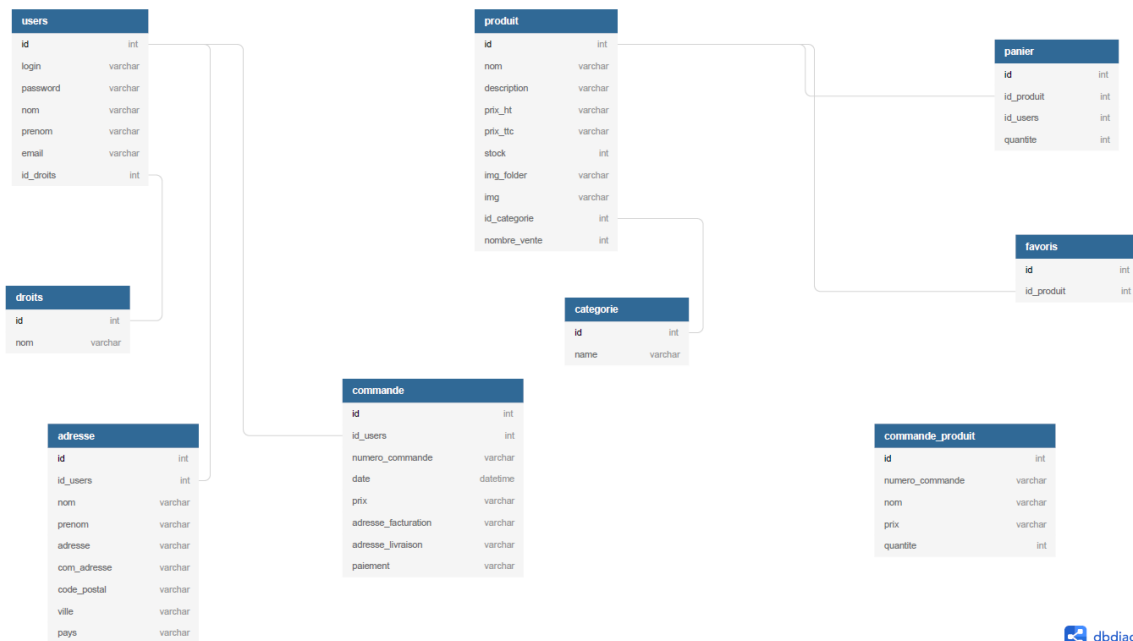
Période d'exercice ▶ Du : 19/10/2020 au : 16/11/2020

## 5. Informations complémentaires (facultatif)

## Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 2- La Boutique de Luc API Restful



dbdiagram.io

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index	A_I	Commentaires	Virtualité	Déplacer une colonne
	INT		Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>			
	INT		Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>			
	INT		Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>			
	INT		Aucun(e)			<input type="checkbox"/>	---	<input type="checkbox"/>			

Structure

Commentaires de table :

Interclassement :

Moteur de stockage :

```

public function create($data)
{
    $data['password'] = password_hash($data['password'], PASSWORD_DEFAULT);
    echo json_encode($data);
    $this->db->insert('users', $data);
}
    
```

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

Lors de la conception de l'API RESTful, nous avons conçu la base de données. Tout d'abord, nous l'avons fait sur papier en même temps que la conception des pages de la boutique.

Pour faire la maquette, nous avons utilisé dbdiagram.io est un outil simple et gratuit pour dessiner des diagrammes ER en écrivant simplement du code. Il est conçu pour les développeurs et les analystes de données.

C'est ici qu'on va créer nos tables, on définit d'abord le nom de la table ainsi que son initiale, on ouvre ensuite une accolade et on va y renseigner les éléments qu'on veut mettre dans la table ainsi que leur type. Dbdiagram va ensuite récupérer ces informations et faire le diagramme de la base de données avec toutes les informations qu'on lui a données.

Dès que cette étape est faite, nous pouvons créer la base de données sur phpmyadmin. Je crée mes tables une par une.

Voici l'interface permettant la création d'une table, on y renseigne le nom, le type, la taille/valeur, interclassement et AI (Auto incrémentation toujours sur l'id). Par exemple, pour une table user si je mets firstname je vais lui assigner un type varchar, une taille 255 et un interclassement en utf-8 pour éviter les problèmes de caractères spéciaux.

La base de données enfin créée on peut commencer à coder l'API. Pour cela, j'ai utilisé le framework PHP Codeigniter.

Dans le model User, nous faisons appel à la base de données en utilisant le langage mysql.

Ici, pour la méthode create, nous utilisons le langage SQL pour insérer une donnée dans la table user. Dans mon cas, j'envoie un tableau de données avec le prénom, le nom, l'email et le mot de passe. Avant l'envoi des données, je récupère le mot de passe pour le hasher/crypter. Par la suite je convertis mes données en JSON et je les envoie à ma base de données avec `$this->db->insert('la table ou je veux envoyer mes données' $mesdonnees)`.

La connexion à la base de données a été faite grâce à la fonction `construct__`. Pour connecter Codeigniter avec la base de données il faut modifier le fichier database.php avec les informations de notre base de données.

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 2. Précisez les moyens utilisés :

### Langages utilisés :

- PHP
- SQL
- Javascript

### Frameworks utilisés :

- CodeIgniter
- ReactNative with Expo

### IDE utilisés :

- Visual Studio Code

### Application de gestion de base de données :

- phpMyAdmin

### Maquette de la base de données :

- DbDiagram.io

## 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec le pôle web dans lequel je suis ainsi que l'équipe back-office.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ La Plateforme

Chantier, atelier, service ▶ Web Developer

Période d'exercice ▶ Du : 14/12/2020 au : 30/07/2021

## 5. Informations complémentaires (facultatif)

## Activité-type 3

Concevoir et développer une application multicouches répartie en intégrant les recommandations de sécurité

Exemple n° 1 - Le Port de Marseille

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

Le port de Marseille est un projet de gestion. Dans cette partie, nous allons parler de tout ce qui se passe avant le développement d'une application : la conception.

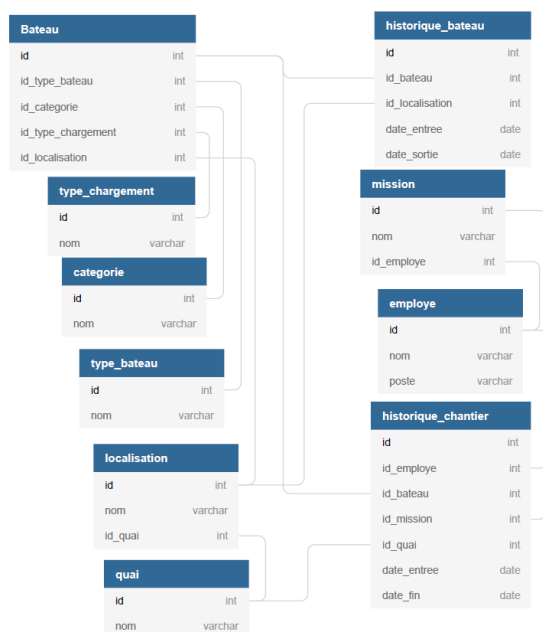
#### La première étape de la conception : les maquettes

Lorsque l'on commence la conception d'un site, il faut toujours commencer par la maquette. Savoir quelles données sont nécessaires pour le site. Réfléchir à quelle base de données utilisés, quelles tables créer.

Pour le port de Marseille, la maquette de la base de données c'est fait en trois étapes :

- le MCD : est le modèle conceptuel de données, représentation graphique qui permet de comprendre comment les éléments sont liés entre eux (entité 1 rectangle 1 entité, les propriétés, les relations entre les entités, les cardinalités).
- le MLD : est un modèle logique de données permet de décrire la structure utilisée sans faire référence à un langage de programmation.
- le MPD : est le modèle physique de données. Cette étape permet de construire la structure finale de la base de données avec les différents liens entre les éléments qui la composent: c'est ici que le vocabulaire change : les entités deviennent des tables, les propriétés se transforment en champs, les propriétés se trouvant au milieu d'un relation crée une nouvelle table, les identifiants se transforment en clé et les relations et les cardinalités se transforment en champs parfois soulignés : on crée des "clés étrangères" reliés à une "clé primaire" dans un autre table.

Ce sont les trois étapes les plus importantes lors de la conception de la base de données. Ce sont des étapes que l'on fait rapidement lorsque l'on fait le diagramme de base de données.



Modèle physique de données du Port de Marseille

La maquette de la base de données est finie, passons maintenant à la maquette du site. Nous avons fait cette maquette en trois étapes :

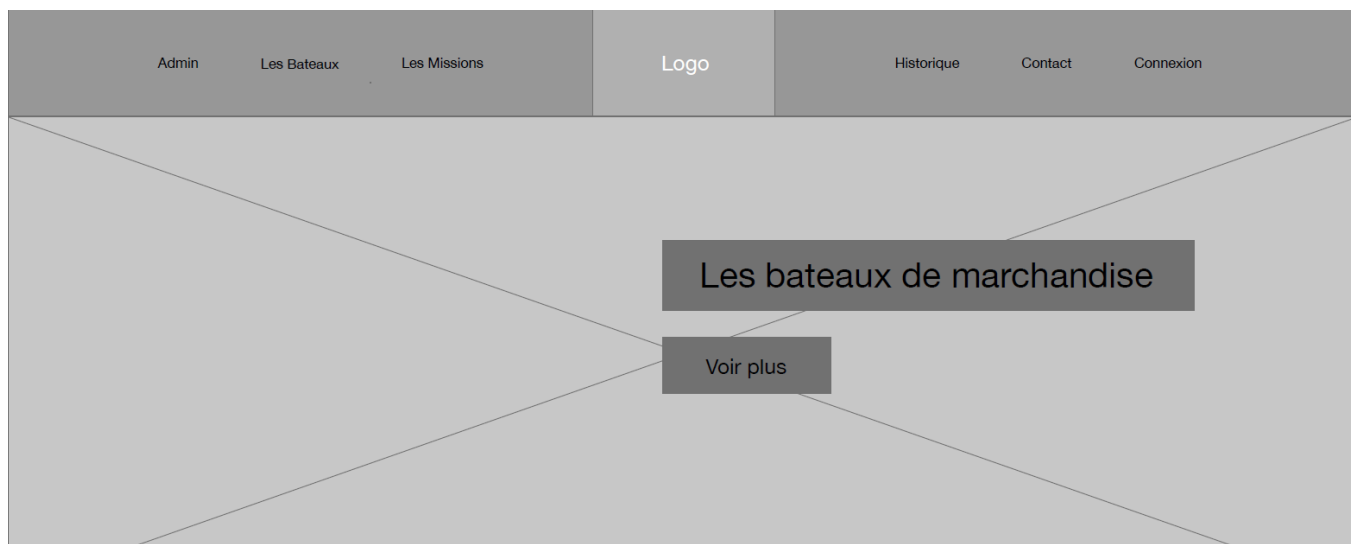
- Le zoning : c'est une schématisation grossière de ce que sera la future page web. On utilise des blocs pour déterminer où se trouvent les contenus et fonctionnalités.
- maquette ou wireframe : il s'agit des détails des zonings avec l'intégration du contenu dans les blocs.
- Prototype : Cette méthode consiste à élaborer des prototypes de l'interface finale d'un site web ou d'une application mobile.

Logo

Menu

Bateau

## Le zoning du Port de Marseille



## La Maquette du Port de Marseille



## Les Bateaux



### Les bateaux de Marchandise

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat



### Les bateaux de Marchandise

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

## Prototype du port de Marseille

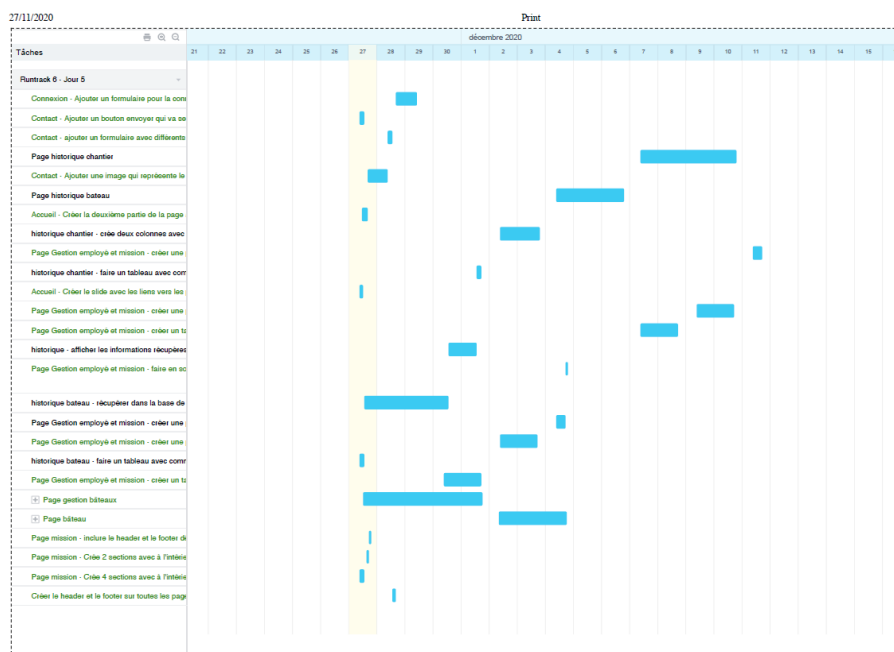
L'étape suivante est le cahier des charges, il est divisé en deux parties : une partie technique et une partie fonctionnelle. Pour résumer, la partie technique contient la page de connexion, la partie admin et la page d'accueil et la partie fonctionnelle contient toutes les pages du site, cette partie va plus servir à quoi servent les pages et ce qu'on y voit.

Par exemple, la page de connexion :

Dans la partie technique, La page de connexion convient à l'authentification pour permettre aux employés d'avoir accès à la gestion des bateaux, missions, du personnel ainsi que l'historique des bateaux et chantiers. De base, un utilisateur administrateur existe dans la base de données et sert à l'addition d'autres comptes pour les autres employés si nécessaire. Les mots de passe sont cryptés avec l'algorithme de hachage Blowfish.

Dans la partie fonctionnelle, Cette page permet à un utilisateur de se connecter au site via un formulaire, contenant un identifiant et un mot de passe. Ce formulaire va permettre à l'utilisateur d'avoir accès à toutes les données présentes dans la partie Admin : tant qu'un utilisateur n'est pas connecté, il ne peut pas accéder aux fonctions de l'administrateur.

Pour finir, lorsqu'on a finalement défini les tâches on doit définir le temps qu'elles vont prendre. Pour cela, on va faire ce qu'on appelle un diagramme de Gantt. Le diagramme de Gantt permet de visualiser dans le temps les diverses tâches composant un projet. Il s'agit d'une représentation d'un graphe connexe, valué et orienté, qui permet de représenter graphiquement l'avancement du projet.





# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 2. Précisez les moyens utilisés :

### Outils utilisés :

- Adobe XD (Pour la maquette)
- Google Slide (Pour l'arborescence )
- DbDiagram.io (pour la maquette de la base de données)
- Google Docs (Pour le cahier des charges )
- trello (pour la répartition des tâches)
- le Diagramme de Gantt (Pour la répartition des tâches d'avoir une temporalité de la réalisation des tâches du projet, de connaître leur date de début et de fin pour estimer le temps que va prendre la réalisation du projet.)

## 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec Luc Van Der Meijden, Justine Lesur et Marceau Maubert

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ LaPlateforme\_

Chantier, atelier, service ▶ Web Developer

Période d'exercice ▶ Du : 23/11/2020 au : 28/11/2020

## 5. Informations complémentaires (facultatif)

## Activité-type 3 Concevoir et développer une application multicouches répartie en intégrant les recommandations de sécurité

Exemple n° 2 - La Boutique de Luc React Native

---

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

Ce projet a été développé pendant mon cursus à la Plateforme\_. Ce projet a été développé avec deux frameworks : React Native et CodeIgniter.

Dans cette partie, nous allons uniquement nous concentrer sur l'API qui a été faite avec CodeIgniter. CodeIgniter est un framework PHP utilisant le modèle de développement MVC. Il utilise plus précisément Hierarchical Model View Controller la séparation claire entre l'acquisition de données (le M de Model), la présentation de ces données (le V de View) et la gestion des demandes des utilisateurs (le C de Controller), le modèle HMVC y ajoute une composante hiérarchique pour encore plus de modularité.

#### Le Model :

Les modèles de données définissent la structure et la manière d'accéder aux données. C'est là que l'on code l'enregistrement, la lecture, la mise à jour et la suppression des données. Elle contient les logiques métiers.

#### La View :

Les vues sont responsables de la présentation des données. Dans le cadre d'une application web les vues servent souvent à générer des pages web (HTML) mais aussi à présenter de manière différentes les mêmes données.

#### Le Controller :

Les Controller font le lien entre les modèles, les vues et les requêtes de l'utilisateur. C'est le routeur du framework qui va, dans un premier temps, faire le lien entre l'url et la classe/méthode de contrôleur à utiliser. C'est ensuite de ce contrôleur que les données nécessaires vont être manipulées via le modèle adapté, puis présentées en utilisant une vue.

Dans notre exemple, nous allons étudier la partie Admin de notre Boutique, plus particulièrement la partie Users.

```
public function getUsers()  
{  
    //$received_Token = $this->input->request_headers('Authorization');  
    try  
    {  
        /*if(isset($_GET['id']))  
        {  
            $jwtData = $this->objOfJwt->DecodeToken($received_Token['Token']);  
            if($jwtData['id_droits'] == "1337")  
            {*/  
                $query = $this->db->query("SELECT * FROM users WHERE id='".$_GET['id']."'");  
                $data = $query->row_array();  
                echo json_encode($data);  
            /*}  
            else  
            {  
                echo $jwtData['id_droits'];  
                echo "Vous n'avez pas le droit d'accéder à cette page";  
            }  
        }*/  
    }  
    catch (Exception $e)  
    {  
        http_response_code('401');  
        echo json_encode(array( "status" => false, "message" => $e->getMessage()));exit;  
    }  
}
```

Dans le modèle, on retrouve la logique métier. Ici, on cherche à récupérer les informations de l'utilisateur et à les afficher pour permettre à l'administrateur de modifier ou supprimer un utilisateur.

Ici, on sélectionne tous de la table Users où l'id correspond à l'id reçu depuis le front. Lorsque l'id est trouvé on récupère ensuite toutes les informations liées à cette Id.

```
public function index()
{
    try
    {
        $data=$this->Models_Users_Admin->getUsers();
    }
    catch (Exception $e)
    {
        echo 'Exception reçue : ', $e->getMessage(), "\n";
    }
}
```

Dans le contrôleur, on va créer une variable \$data où on va faire appel à la fonction getUsers() du Model pour récupérer les résultats qui ont été récupérés dans la base de données.

Par la suite, dans React nous allons utiliser Axios pour faire appel à cette fonction pour afficher les utilisateurs.

## 2. Précisez les moyens utilisés :

### Langages utilisés :

- PHP
- SQL
- Javascript

### Frameworks utilisés :

- CodeIgniter
- ReactNative with Expo

### IDE utilisés :

- Visual Studio Code

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec mes camarades de cursus : Luc Van Der Meijden et Justine Lesur.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La Plateforme*

Chantier, atelier, service ▶ *Web Developer*

Période d'exercice ▶ Du : *14/12/2020* au : *30/07/2021*

## Activité-type 3

Concevoir et développer une application multicouches répartie en intégrant les recommandations de sécurité

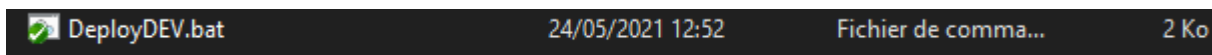
Exemple n° 3 - Le Déploiement et Test API avec Hopps Group

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

Pendant mon alternance j'ai pu travailler sur le projet ExtranetLogIT. Ce site est une application pour les clients de HOPPS. Ce site va permettre aux clients de voir les ventes de leur colis, les articles les plus vendus etc. avec des graphiques pour pouvoir observer une évolution ou non de leurs services.

Pour ce projet, j'ai pu faire le déploiement en environnement de développement. Pour faire ça, j'ai été accompagné de mon tuteur.

Pour déployer un site, il faut ouvrir un fichier nommé deploy.BAT. Dans mon cas à moi, j'ai utilisé ce fichier là :



Pour que le déploiement marche, il faut absolument être sur le même réseau que Hopps. En général, on se connecte à la VM pour être sûr d'être sur le même réseau. On doit mettre un lien sur la machine vers les fichiers où on veut livrer. On choisit ensuite la version du patch majeur, mineur etc. On se connecte au serveur (cela est fait automatiquement).

On a aussi un dossier backup avec la dernière version du site comme ça si il y a un problème on peut toujours récupérer cette version.

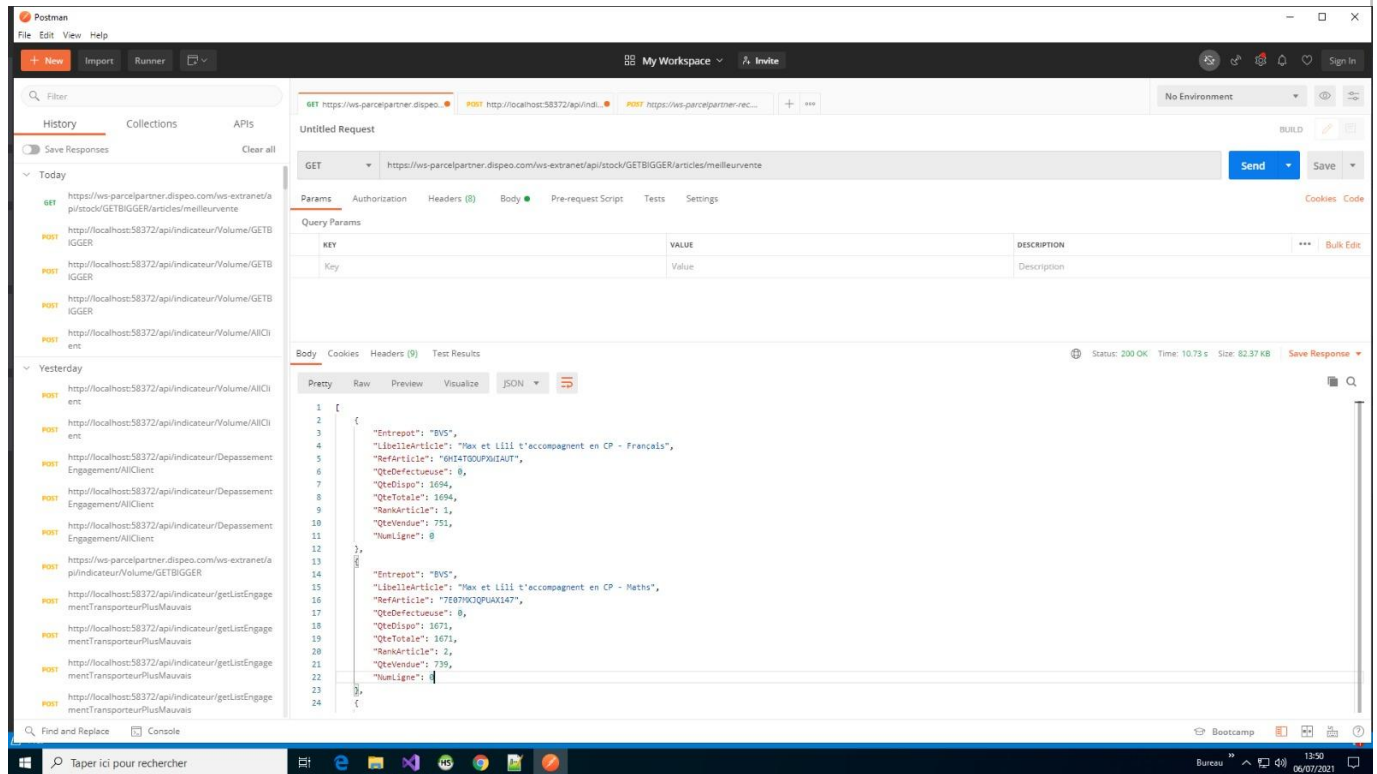
A chaque deploy, la dernière version du site est mise dans le backup ensuite on fait un ng build et on envoi les fichiers sur le serveur de développement.

```
Es-tu connecte au reseau HOPPS ?
Appuyez sur une touche pour continuer...
Quelle version ?
1- Majeure
2- Mineure
3- Patch
0- Sans changement de version
Choisir la version : 0
Connexion au serveur
La commande s'est terminée correctement.

Sauvegarde des donnees
Remplacer \\R04WSGITEXTRA-1.DISPEO.ADSI\e$\HOPPS\SITE-HOPPS\rec-extranet-backup\web.config (Oui/Non/Tous)? _
```

# DOSSIER PROFESSIONNEL (DP)

Pour le projet Extranet, j'ai pu travailler sur la partie API. J'ai pu travailler sur le controller article avec les meilleures ventes. Pour tester mon API, j'ai utilisé Postman et j'ai lancé Visual Studio.



Tout d'abord on choisit l'action de l'API : dans notre cas, il s'agit d'un GET car on veut afficher les meilleures ventes.

On vient ensuite insérer le lien de l'api : "`api/stock/GETBIGGER/articles/meilleurvevente`" = GETBIGGER est un paramètre il correspond au p\_client. Il existe encore d'autres clients comme JAPHY et LAMARQUEENMOINS.

Vu que l'on effectue un GET nous n'avons pas besoin d'utiliser le body pour ajouter des paramètres. On déclenche ensuite la requête en appuyant sur le bouton Send.

La requête a bien marché puisque l'on récupère bien des résultat :

```
[
  {
    "Entrepot": "BVS",
    "LibelleArticle": "Max et Lili t'accompagnent en CP - Français",
    "RefArticle": "6HI4TGOUPIXIAUT",
    "QteDefectueuse": 0,
    "QteDispo": 1694,
    "QteTotale": 1694,
    "RankArticle": 1,
    "QteVendue": 751,
    "NumLigne": 0
  },
]
```

Pour classer les meilleurs ventes nous nous basons sur la quantité vendue = Qte Vendue

## 2. Précisez les moyens utilisés :

### Langages utilisés :

- C#
- Javascript

### Framework utilisés :

- Doc NET
- Angular
- Bootstrap

### IDE utilisés :

- Visual Studio pour C#
- Visual Studio Code pour Angular



## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

### 3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec le pôle web dans lequel je suis ainsi que l'équipe back-office.

### 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *HoppsGroup COLIS PRIVE .*

Chantier, atelier, service ▶ *Web Developer*

Période d'exercice ▶ Du : *21/10/2020* au : *30/07/2021*

### 5. Informations complémentaires *(facultatif)*

---

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

---

### Titres, diplômes, CQP, attestations de formation

*(facultatif)*

Intitulé	Autorité ou organisme	Date
Baccalauréat Sciences et Technologies de Laboratoire	Lycée Paul Langevin	2018
Développeur web et web mobile	La Plateforme_	2020

### Déclaration sur l'honneur

---

Je soussigné(e) Julia Denivet ,  
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 **Marseille**

le **06/07/2021**

pour faire valoir ce que de droit.

Signature :

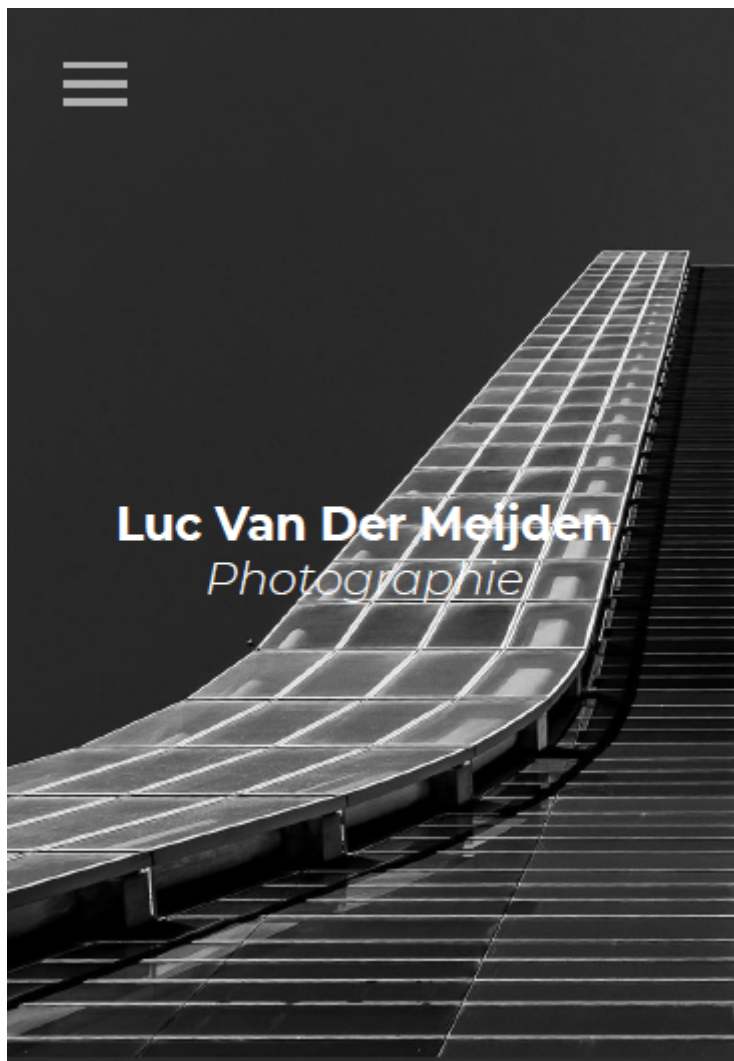
## Documents illustrant la pratique professionnelle

*(facultatif)*

Intitulé
Cliquez ici pour taper du texte.

## ANNEXES

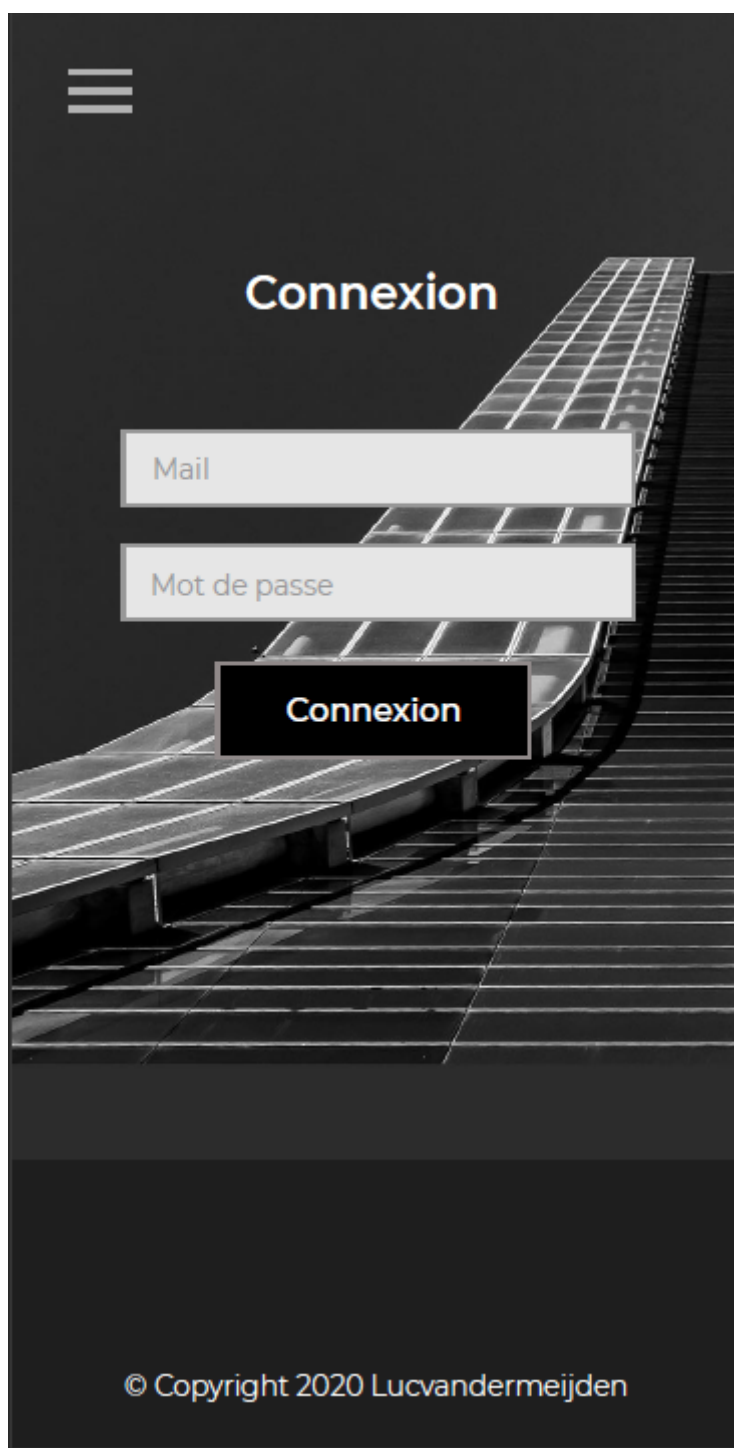
*(Si le RC le prévoit)*



## Les favoris

---







A registration form is displayed over a background image of solar panels. The form consists of the following elements:

- A hamburger menu icon (three horizontal lines) in the top left corner.
- The title "Inscription" centered at the top of the form area.
- A text input field labeled "Mail".
- A text input field labeled "Mot de passe".
- A text input field labeled "Confirmer le mot de passe".
- A black button with the white text "Inscription" at the bottom.






## Portraits




Nom Prénom - Lieu - 2020




Nom Prénom - Lieu - 2020



## Panier



Photoshooting de Laetia  
30 x 50 cm - 1 exemplaire  
**89,99 €**



Photoshooting de Manon  
30 x 50 cm - 1 exemplaire  
**99,99 €**

**Total - 189.98 €**

Commander

© Copyright 2020 Lucvandermeijden

---

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>






---

---

## Administrateur

*Gestion catégorie  
et produits*

Tous les Produits

	IMG_5931.jpg	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	IMG_5932.jpg	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	IMG_5933.jpg	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	IMG_5934.jpg	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	IMG_5935.jpg	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Tous les Albums


Ajouter un album

Photoshooting d'Agnès	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Photoshooting de Manon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Photoshooting de Jess	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Toutes les Catégories

Ajouter une catégorie

Tout	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Paysages	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Architecture	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Portrait	<input checked="" type="checkbox"/>	<input type="checkbox"/>



## Administrateur

*Gestion des utilisateurs*

Tous les utilisateurs	
Luc Van der Meijden	<input checked="" type="checkbox"/>
Julia Denivet	<input checked="" type="checkbox"/>
Justine Lesur	<input checked="" type="checkbox"/>
Pascal Assens	<input checked="" type="checkbox"/>
Terry Cristinelli	<input checked="" type="checkbox"/>
Gregory Fauvel	<input checked="" type="checkbox"/>
Olivier Crozet	<input checked="" type="checkbox"/>

© Copyright 2020 Lucvandermeijden