



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

<i>Nom de naissance</i>	▸ Courrin
<i>Nom d'usage</i>	▸ Courrin
<i>Prénom</i>	▸ Maeva
<i>Adresse</i>	▸ 331 chemin du bon pin 06210 Mandelieu

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.

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



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

DOSSIER PROFESSIONNEL (DP)

Sommaire

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. p. 5

- CP 1 Maquetter une application. p. 5
- CP 2 Réaliser une interface statique et adaptable. p. 8
- CP 3 Développer une interface utilisateur web dynamique. p. 11
- CP 4 Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce. p.

Intitulé de l'activité-type n° 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité. p.

- CP 5 Créer une base de données. p. 16
- CP 6 Développer les composants d'accès aux données. p. 21
- CP 7 Développer la partie back-end d'une application web ou web mobile. p. 24
- CP 8 Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce . p.

Titres, diplômes, CQP, attestations de formation (facultatif) p. 29

Déclaration sur l'honneur p. 30

Documents illustrant la pratique professionnelle (facultatif) p.

Annexes (Si le RC le prévoit) p.

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

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

CP 1 - Maquetter une application.

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

J'ai réalisé les wireframes et user stories ci-dessous dans le cadre d'un projet personnel.

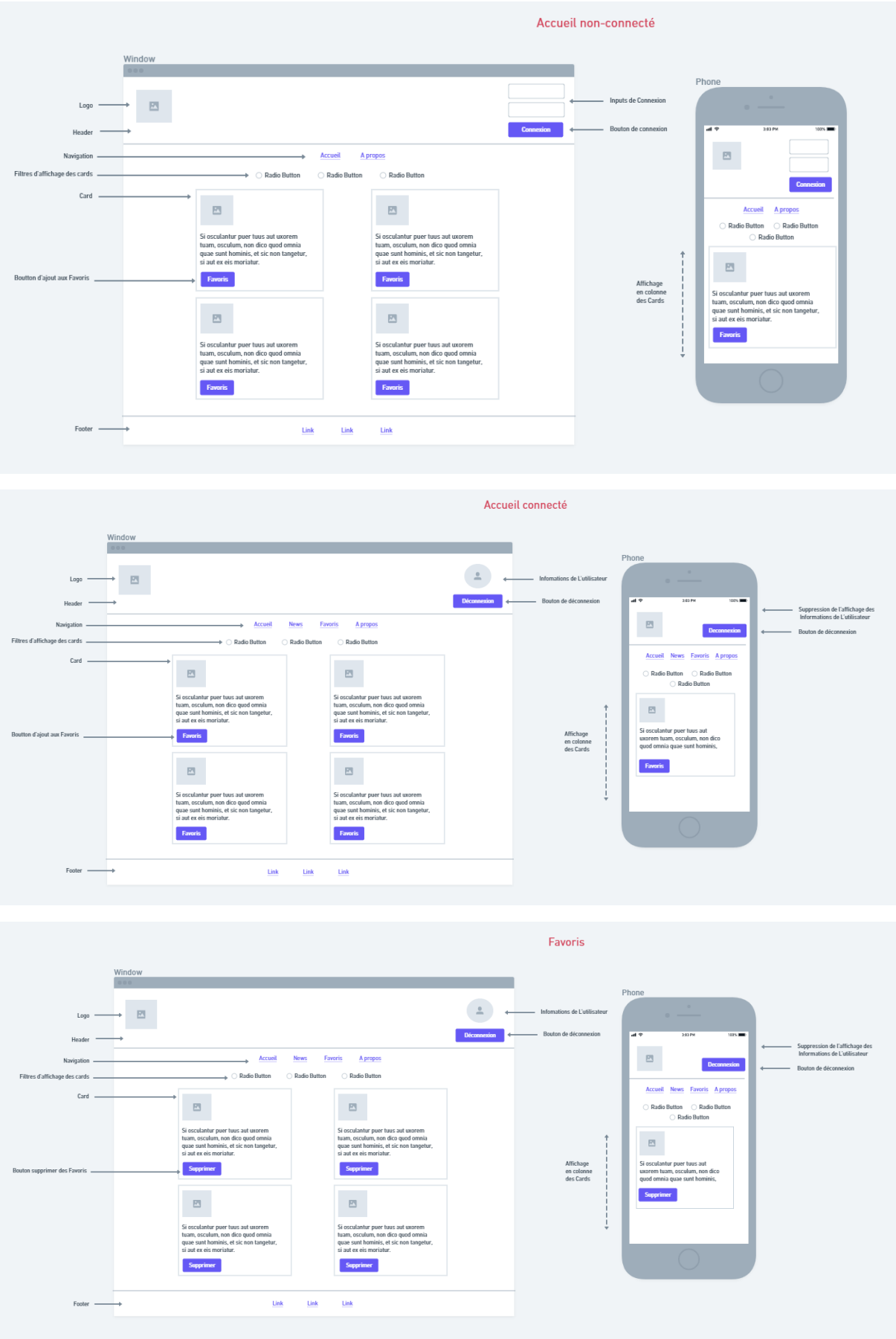
Une user story est une explication non formelle, générale d'une fonctionnalité logicielle, elle représente un objectif final exprimé du point de vue de l'utilisateur.

Un wireframes ou maquette fonctionnelle est un schéma utilisé lors de la conception d'une interface utilisateur pour définir les zones et composants qu'elle doit contenir.

User Stories :

En tant que ...	Je veux ...	Afin de ...
Visiteur	Consulter les articles	M'informer sur les produits/articles
Visiteur	Pouvoir me connecter	Profiter des services
Utilisateur connecté	Trier les articles	Simplifier ma recherche
Utilisateur connecté	Ajouter un article aux favoris	Organiser ma recherche
Utilisateur connecté	Modifier les articles des favoris	Organiser ma recherche
Utilisateur connecté	Créer un nouvelle article	Proposer un nouvel article
Utilisateur connecté	Pouvoir me déconnecter	Sécurisé mes informations

Wireframes :



DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour réaliser les wireframes, j'ai utilisé le site <https://whimsical.com/> et pour la réalisation des user-stories j'ai utilisé un fichier excel.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ces documents.

4. Contexte

Nom de l'entreprise, organisme ou association - *O'Clock*

Chantier, atelier, service - *Projet personnel.*

Période d'exercice - Du : *27/02/2021* au : *09/08/2021*

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.

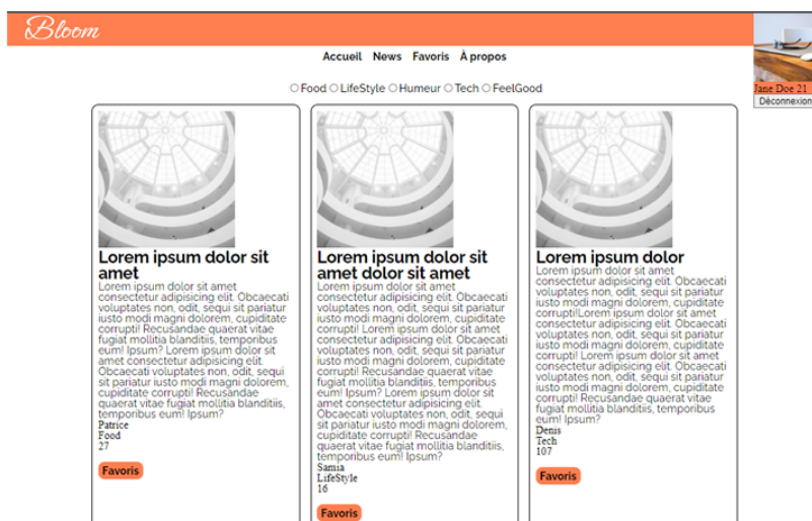
CP 2 - Réaliser une interface web statique et adaptable.

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

Un site web réactif est un site web dont la conception vise, grâce à différents principes et techniques, à offrir une consultation confortable sur des écrans de tailles très différentes.

J'ai choisi de coder le site en mobile-first puis d'adapter vers une version desktop.

Page d'accueil



```
26 @media screen and (max-width: 400px) {
27   .login_unlogged{
28     position: fixed;
29     top: 0;
30   }
31   .login_label{
32     display: none;
33   }
34   .login_input {
35     width: 12ch;
36   }
37   .login_logged{
38     position: fixed;
39     top: 0;
40     height: 0%;
41     min-height: 0vh;
42   }
43   .login_logged_info{
44     display: none;
45   }
```

Pour le site j'ai souhaité que dans la version mobile, dans le but de rationaliser l'espace disponible, le composant chargé de l'affichage des informations de l'utilisateur soit réduit à son minimum.

Dans la version mobile, seule reste le bouton de déconnexion.

La media queries ci-contre s'applique aux appareils dotés d'un écran dont la valeur d'affichage maximum est de 400px.

Dans ce cas, on applique un display none sur les labels accompagnants les inputs.

DOSSIER PROFESSIONNEL (DP)

```
59
60 @media screen and (max-width: 1050px) {
61     .content_cards{
62         justify-content: center;
63         padding-right: 4em;
64     }
65     .card{
66         max-width: 40%;
67     }
68     .card-container p{
69         font-size: 1.5em;
70     }
71 }
72
73 @media screen and (max-width: 800px) {
74     .content_cards{
75         flex-direction: column;
76         align-items: center;
77     }
78     .card{
79         max-width: 100%;
80     }
81 }
82
83 @media screen and (max-width: 400px) {
84     .card-container p{
85         font-size: 1em;
86     }
87 }
```

Pour gérer l'affichage de mes cartes là aussi j'utilise les media queries.

Pour une valeur d'affichage maximum de 1050px j'ajuste le padding, la width des cards et la font size à l'aide de valeur relative, pour garder une bonne lisibilité .

Pour une valeur d'affichage maximum de 800px je souhaite changer le sens de l'affichage des cards pour être en adéquation avec le support.

Ici j'applique un flex-direction column, puis un align items center, là aussi j'ajuste la valeur de la width.

Enfin, pour une valeur d'affichage maximum de 400px, la media queries précédente s'appliquant toujours je me contente d'ajuster la font-size du texte .

2. Précisez les moyens utilisés :

Afin de rendre le site responsive, j'ai utilisé des media queries qui permette d'appliquer du CSS de façon conditionnelle en fonction du type d'écran, de la largeur, de la hauteur, de la zone d'affichage.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association - *O'Clock*

Chantier, atelier, service - *Projet personnel*

Période d'exercice - Du : *27/02/2021* au : *09/08/2021*

5. Informations complémentaires (facultatif)


```

JS index.js ...App U    JS index.js ...Content U    JS Card.js U ●
src > components > Content > JS Card.js > ...
1  import React from 'react';
2
3
4  const Card = ({ post, favoris, setFavoris }) => {
5
6      const addPost = (post) => {
7          setFavoris({...favoris, [post.id]: post});
8      };
9
10     return (
11         <li className="card">
12             <img src={post.photo} alt="illustration" />
13             <div className="card-container">
14                 <h4>{post.title}</h4>
15                 <ul className="card-info">
16                     <p>{post.content}</p>
17                     <li>{post.author}</li>
18                     <li>{post.categorie}</li>
19                     <li>{post.likes}</li>
20                 </ul>
21             </div>
22             <button
23                 className="card-btn-favoris"
24                 type="button"
25                 value={post}
26                 onClick={() => addPost(post)}
27             >
28                 Favoris
29             </button>
30         </li>
31     );
32 };

```

Pour cela dans mon composant Card je créé une fonction addPost.

Cette fonction va être déclenchée au clic sur le bouton "Favoris".

En paramètre de cette fonction se trouve l'objet "post".

Au clic sur le bouton on va ajouter l'objet "post", à un state "favoris" grâce à la fonction setFavoris.

```

18
19  const App = () => {
20      const [isLoggedIn, setIsLoggedIn] = useState(false);
21
22      const [favoris, setFavoris] = useState([""]);
23
24      useEffect(() => {
25          return () => {
26              setFavoris();
27          }
28      }, []);
29  };

```

J'ai créé le state "favoris" dans mon composant App.

J'utilise les props de react pour faire remonter l'état de mon state jusqu'au composant App.

Dans mon composant App l'utilisation du hook useEffect avec en callback la fonction setFavoris permet d'avoir un re-rendu du composant à chaque changement de ce state.

DOSSIER PROFESSIONNEL (DP)

```
JS index.js ...App U • JS index.js ...Favorites U X JS index.js ...Content U JS Card.js U JS ...
src > components > Favorites > JS index.js > ...
1  import React from 'react';
2
3  import './index.css'
4
5  const Favorites = ({ favoris, setFavoris }) => {
6
7      const deletePost = (e) => {
8          const id= e.target.value;
9          const newFav = Object.values(favoris).filter((fav) => fav.id !== id);
10         setFavoris(newFav);
11     };
12
13     return (
14         <ul className="favorites__content">
15             {Object.keys(favoris).slice(1).map((favori, i) => (
16                 <ul className="favorites" key={i}>
17                     <img src={favoris[favori].photo} alt="illustration"/>
18                     <div className="favorites-container">
19                         <h4>{favoris[favori].title}</h4>
20                         <ul className="favorites-info">
21                             <p>{favoris[favori].content}</p>
22                             <li>{favoris[favori].author}</li>
23                             <li>{favoris[favori].categorie}</li>
24                             <li>{favoris[favori].likes}</li>
25                         </ul>
26                         <button
27                             className="card-btn-favoris"
28                             type="button"
29                             value={favoris[favori].id}
30                             onClick={(e) => deletePost(e)}
31                         >
32                             Supprimer
33                     </button>
34                 </div>

```

Grâce au props de React je peux transmettre mon state dans mon composant favori.
J'utilise la fonction map() pour créer un affichage de toutes les valeurs du state.

2. Précisez les moyens utilisés :

J'ai utilisé les Hook de react pour créer des interactions côté client.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *O'Clock*

Chantier, atelier, service ▶ *Projet personnel*

Période d'exercice ▶ Du : *27/02/2021* au : *09/08/2021*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

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

CP 4 - Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.

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)

Activité-type 2

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

CP 5 - Créer une base de données.

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

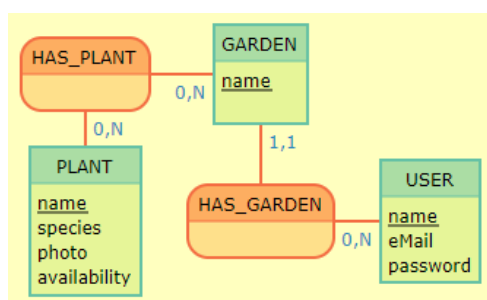
J'ai créé cette base de données dans le cadre d'un projet personnel, de type exercice pratique.

1-MCD (modèle conceptuel de données)

Dans un premier temps j'ai réalisé le MCD de l'application. Le MCD représente la structure des données et ses relations (associations) :

J'ai listé 3 différentes entités (USER, GARDEN, PLANT) ainsi que les attributs pour chacune.

J'ai établi les relations entre les différentes entités (un utilisateur possède un jardin qui contient des plantes) ainsi que les cardinalités :



- Un utilisateur peut avoir au minimum zéro jardin et au maximum plusieurs jardins (Relation de type 0,N)
- Un jardin peut appartenir au minimum et au maximum à un utilisateur (Relation de type 1,1)
- Un jardin peut contenir au minimum aucunes plantes et au maximum toutes. (Relation de type 0,N)
- Une plante peut appartenir au minimum à aucun jardin et au maximum à plusieurs jardins (Relation de type 0,N)

2-MLD (modèle logique de données) :

Dans un second temps j'ai réalisé le MLD à partir du MCD.

Dans le MLD, on ajoute une clé primaire à chacune des tables qui sera représentée par un id, ainsi qu'une clé étrangère dans les relations de type 1:N. La clé étrangère sera toujours du côté du 1. (représentée par un #TABLE(id)) .

```
USER(id, name, eMail, password)
GARDEN(id, name, #USER(id))
HAS_PLANT(#GARDEN(id), #PLANT(id))
PLANT(id, name, species, photo, availability)
```

Dans notre MCD, nous avons une association N:N entre GARDEN et PLANT. De ce fait, nous devons créer une table de liaison contenant deux clés étrangères afin de relier les deux tables.

3-MPD (modèle physique de données) :

Ici, j'ai directement créé le fichier SQL qui sera importé dans la base de données lors de sa création. Ce fichier sert à structurer la base de données.

Pour commencer je crée une base de données vide avec CREATE DATABASE (ligne 2) puis j'ouvre cette BDD (ligne 3), avec la commande \c .

Je supprime les tables si elles existent avec un DROP TABLE (ligne 5 - 8)

Je crée chacune des tables en me servant du MLD.

Les id doivent être uniques et incrémentés, pour cela on indique "INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY".

Pour chaque champ, j'indique le type (INT pour des nombres, TEXT pour du texte...) et j'indique NOT NULL lorsque le champ doit obligatoirement contenir une donnée.

```
1 CREATE DATABASE plants;
2 \c plants;
3
4 DROP TABLE IF EXISTS user;
5 DROP TABLE IF EXISTS garden;
6 DROP TABLE IF EXISTS plant;
7 DROP TABLE IF EXISTS has_plant;
8
9
10 CREATE TABLE "user" (
11     id int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
12     firstName TEXT NOT NULL,
13     lastName TEXT NOT NULL,
14     email EMAIL NOT NULL UNIQUE,
15     password text NOT NULL
16 );
17
18 CREATE TABLE garden (
19     id int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
20     name TEXT NOT NULL,
21     user_id int NOT NULL REFERENCES "user"(id)
22 );
23
24 CREATE TABLE has_plant (
25     garden_id int NOT NULL REFERENCES garden(id),
26     plant_id int NOT NULL REFERENCES plant(id)
27 );
28
29 CREATE TABLE plant (
30     id int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
31     name TEXT NOT NULL,
32     species TEXT NOT NULL,
33     photo TEXT NOT NULL,
34     availability BOOLEAN DEFAULT false
35 );
36
37
```

2. Précisez les moyens utilisés :

Afin de créer cette base de données, j'ai utilisé la méthode MERISE qui est une méthode d'analyse et de conception de base de données. Elle est basée sur le principe de la séparation des données et des traitements et utilise des modèles répartis sur trois niveaux : conceptuel, logique et physique. J'ai utilisé le SGBD (Système de Gestion de Base de Données) PostgreSQL pour cet exercice.

J'ai utilisé le site <http://mocodo.wingi.net/> pour créer le MCD de l'application.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *O'Clock.*

Chantier, atelier, service ▶ *Projet personnel.*

Période d'exercice ▶ Du : *27/02/2021* au : *09/08/2021*

DOSSIER PROFESSIONNEL (DP)

5. Informations complémentaires (facultatif)

Dictionnaire de données:

Table USER			
Champ	Type	Spécificités	Description
id	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	IDENTIFIANT UNIQUE
firstName	TEXT	NOT NULL	Prénom de l'utilisateur
lastName	TEXT	NOT NULL	Nom de l'utilisateur
email	email	NOT NULL	Email de l'utilisateur
password	TEXT	HASH	Mot de passe de l'utilisateur
Table GARDEN			
Champ	Type	Spécificités	Description
id	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	IDENTIFIANT UNIQUE
name	TEXT	NOT NULL	Nom du jardin de l'utilisateur
user_id	INT	NOT NULL	clé étrangère liant le jardin à l'utilisateur
Table PLANT			
Champ	Type	Spécificités	Description
id	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	IDENTIFIANT UNIQUE
name	TEXT	NOT NULL	Nom commun de la plante
species	TEXT	NOT NULL	Especie de la plante
photo	TEXT	NOT NULL	Url vers l'image
availability	BOOLEAN	DEFAULT false	La plante est-elle disponible?
Table HAS_PLANT			
Champ	Type	Spécificités	Description
garden_id	INT	NOT NULL	IDENTIFIANT UNIQUE
plant_id	INT	NOT NULL	IDENTIFIANT UNIQUE

Le dictionnaire de données est créé à partir du MLD, il liste l'ensemble des tables, des champs qui les composent et des relations qui les lient.

Activité-type 2

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

CP 6 - Développer les composants d'accès aux données.

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

1-Connexion à la base de données avec PG

```
server > .env
1  PORT=5000
2
3  PGUSER='postgres'
4  PGHOST='localhost'
5  PGPASSWORD='maevac'
6  PGDATABASE='plants'
7  PGPORT=5432
8
```

Dans un premier temps je vais définir les variables d'environnements nécessaire à la connexion à la base de données, grâce au package dotenv.

Dans un second temps je crée le client de connexion grâce au module pg.

Je commence par extraire la class Pool de pg (ligne 1)

Je crée une nouvelle instance de Pool.

```
JS index.js server M JS index.js ...db A X
server > app > db > JS index.js > ...
1  const { Pool } = require('pg');
2
3  const pool = new Pool();
4
5  module.exports = pool
```

2-Création des Models

La couche Model représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données, de les convertir selon les besoins.

Chaque models dispose de ses propres accès CRUD à la BDD.

Pour permettre à notre code d'être plus lisible et plus facilement maintenable, j'applique le concept "SOC" pour séparation of concerns. et je crée un model par entité de la base de données.

```
JS index.js M JS user.js U X .env JS gardenRouter.js U
server > app > models > JS user.js > user > findOne
1  const db = require('../db/index');
2
3  class user {
4
5      constructor(obj={}){
6          for(const propName in obj){
7              this[propName] = obj[propName]
8          }
9      }
10
```

Ici dans le model user.js on commence par importer notre client de connexion à la BDD. (ligne 1)
On déclare la class User (ligne 16) et on prévoit un constructor (ligne 19 - 23).

exemple:

```
45     async createUser() {  
46         try{  
47             const {rows} = await db.query('INSERT INTO "user"(firstName, lastName, email, password)  
VALUES($1, $2, $3, $4) RETURNING id ', [ this.firstName , this.lastName, this.email,  
this.password ]);  
48             this.id = rows[0].id;  
49             return this;  
50         } catch (error) {  
51             console.log(error);  
52         }  
53     }  
54 }
```

Ici la méthode asynchrone “createUser” est chargée d'insérer dans la base de données, un nouvel utilisateur.

Une fonction asynchrone est une fonction qui s'exécute de façon asynchrone grâce à la boucle d'événement en utilisant une promesse (Promise) comme valeur de retour.

Pour ce faire j'utilise l'objet courant this, grâce auquel je vais pouvoir utiliser les données passées en paramètre depuis le controller.

Sécurité

- Injection SQL

Un injection SQL se produit lorsqu'une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête.

Afin d'empêcher ces attaques, il faut préparer les requêtes SQL.

Dans notre cas (ligne 47) je prépare la requête avec des “token” = \$1.

La requête est réalisée avant d'y insérer les paramètres et de l'exécuter, ce qui empêche un éventuel code inséré dans les paramètres d'être interprété



DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

L'application a été conçue selon le pattern MVC pour Model View Controller.

J'ai utilisé le package dotenv <https://www.npmjs.com/package/dotenv> pour créer mes variables d'environnements.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *O'clock.*

Chantier, atelier, service ▶ *projet personnel.*

Période d'exercice ▶ Du : *27/02/2021* au : *09/08/2021*

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.

CP 7 ➤ 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 :

Mise en place du routeur

Pour plus de clarté et simplifier la maintenance de notre code, on applique le concept de SOC (separation of concerns).

Je crée donc plusieurs instances de Router.

Pour chaque endpoints on va désigner la route, la méthode HTTP (get, post, put, delete) utilisée, et le controller requis.

```
6 const plantRouter = require('./app/router/plantRouter');
7 const gardenRouter = require('./app/router/gardenRouter');
8 const userController = require('./app/router/userRouter');
9
```

```
JS userRouter.js U
server > app > router > JS userRouter.js > ...
1 const { Router } = require("express");
2 const router = Router();
3
4 const userController = require('./controllers/userController');
5
6 //USER
7 router.get('/users', userController.findAll);
8 router.get('/users/:id(\\d+)', userController.findOne);
9
10 router.post('/register', userController.addUser);
11 router.post('/login', userController.findByLogin);
12 router.put('/user/:user_id/edit/password', userController.editPassword);
13 router.put('/user/:user_id/edit/email', userController.editEmail);
14 router.put('/user/:user_id/edit/username', userController.editUserName);
15 router.delete('/user/:user_id/deleteAccount', userController.deleteAccount);
16
17 module.exports = router;
```


Création des controllers

La couche Controller gère les requêtes des utilisateurs. Elle est responsable de retourner une réponse avec l'aide des couches Model et Vue

On commence par importer notre model user (ligne 1), puis on crée notre objet userController (ligne 7).

Ici je vais également utiliser les packages bcrypt et jwt pour garantir une authentification unique et sécurisée.

```
JS index.js M    JS user.js U    JS userController.js U • ⚙
server > app > controllers > JS userController.js > ...
1  const user = require('../models/user');
2  const bcrypt = require("bcrypt");
3  //const jwt = require('../utils/jwtGenerator');
4  const jwt = require('jsonwebtoken');
5
6
7  const userController = {
8    findAll: async( _, res) => {
9      const users = await user.findAll();
10     res.json(users);
11   },
12 }
```

exemple:

```
64
65   addUser : async(req, res) => {
66     try{
67       const hashedPassword = await bcrypt.hash(req.body.password, 10);
68
69       const data = { firstName: req.body.firstName, lastName: req.body.
        lastName, email: req.body.email, password: hashedPassword };
70
71       const userAdd = new user(data);
72       let newUser = await userAdd.createUser();
73     }
```

Ici la méthode addUser est chargée d'interroger le modèle avec les données récupérées depuis la vue grâce au request.body (ligne 69) , puis de renvoyer la réponse du modèle vers la vue.

Dans un premier temps on chiffre le mot de passe grâce bcrypt (ligne 67) pour augmenter la sécurité et rendre les attaques par force brute inutile.

Les données ainsi récupérées, on crée une nouvelle instance de user (ligne 71) avec les données provenant de la vue, puis j'interroge le modèle User avec la méthode createUser du modèle et la nouvelle instance. (ligne 72)

```
73
74     if(newUser){
75         const SECRET_KEY = process.env.JWTSecret;
76
77         const expireIn = '1h';
78         const token = jwt.sign({
79             user: newUser
80         },
81         SECRET_KEY,
82         {
83             expiresIn: expireIn
84         });
85         if(token){
86             return res.status(201).json({
87                 message: 'authentification successful',
88                 newUser,
89                 token
90             });
91         }else{
92             next();
93         };
94     }else if( !newUser) {
95         message = "Cet user existe déjà";
96         res.status(401).json(message);
97     }else{
98         next();
99     }
100 }catch(err){
101     res.status(500).json(err);
102 }
103 },
104
```

En cas de réponse du modèle, (ligne 74) un token unique est généré. (ligne 75 - 85) , et les données (le nouvel utilisateur, le token, et le message) sont renvoyées vers la vue au format Json.

Sécurité

Le bundle JWT Authentication Bundle permet d'utiliser les JSON Web Tokens afin de protéger les ressources de notre API REST.

Lorsque l'utilisateur s'authentifie correctement, un token lui est retourné. Ce token est ensuite transmis côté front afin de permettre à l'utilisateur de lire ou d'écrire les ressources protégées à chaque requête.

Grâce à ce token l'application a les moyens de garantir une authentification sécurisée et unique.

Pour prévenir les attaques par force brute, Hasher les mots de passe représente la meilleure méthode puisque non-réversible.

Cette méthode permet de créer une empreinte de la chaîne fournie (le mot de passe) et de stocker cette empreinte en base, cette méthode est donc la plus recommandée.

Cette opération est réalisée grâce au package bcrypt.

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Dans ce projet j'ai utilisé le framework Express ainsi que Node.js pour mon serveur.

J'ai également utilisé les librairies :

-Bcrypt : <https://www.npmjs.com/package/bcrypt>

-JWT : <https://www.npmjs.com/package/jsonwebtoken>

Pour tester les routes, et les méthodes j'ai utilisé Postman : <https://www.postman.com/>

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *O'Clock*

Chantier, atelier, service ▶ *projet personnel.*

Période d'exercice ▶ Du : *27/02/2021* au : *09/08/2021*

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.

CP 8 - *Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.*

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)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
BTS assistant de gestion	ESCCOM	2011

Déclaration sur l'honneur

Je soussigné(e) Maeva Courrin ,
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 Mandelieu le 02/11/2021

pour faire valoir ce que de droit.

Signature :

Maeva Courrin

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

ANNEXES

(Si le RC le prévoit)