



# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ EL MOUHINE  
*Nom d'usage* ▶ EL MOUHINE  
*Prénom* ▶ Fatima  
*Adresse* ▶ 127 chemin de st Joseph à ste Marthe 13014

## 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.

# DOSSIER PROFESSIONNEL (DP)

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

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

## Sommaire

### Exemples de pratique professionnelle

<b>Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité</b>	<b>p. 5</b>
- Medenpharmakine	p. 5
- <i>Le jeu du Snake</i>	p. 7
- <i>Medenpharmakine</i>	p. 11
<b>Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité</b>	<b>p. 15</b>
- Medenpharmakine	p. 15
<b>Concevoir et développer la persistance des données en intégrant les recommandations de sécurité</b>	<b>p. 21</b>
- Medenpharmakine	p. 21
<b>Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité</b>	<b>p. 25</b>
- Portfolio	p. 25
<b>Déclaration sur l'honneur</b>	<b>p. 29</b>
<b>Annexes (Si le RC le prévoit)</b>	<b>p. 30</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 - Medenpharmakine - Maquetter une application

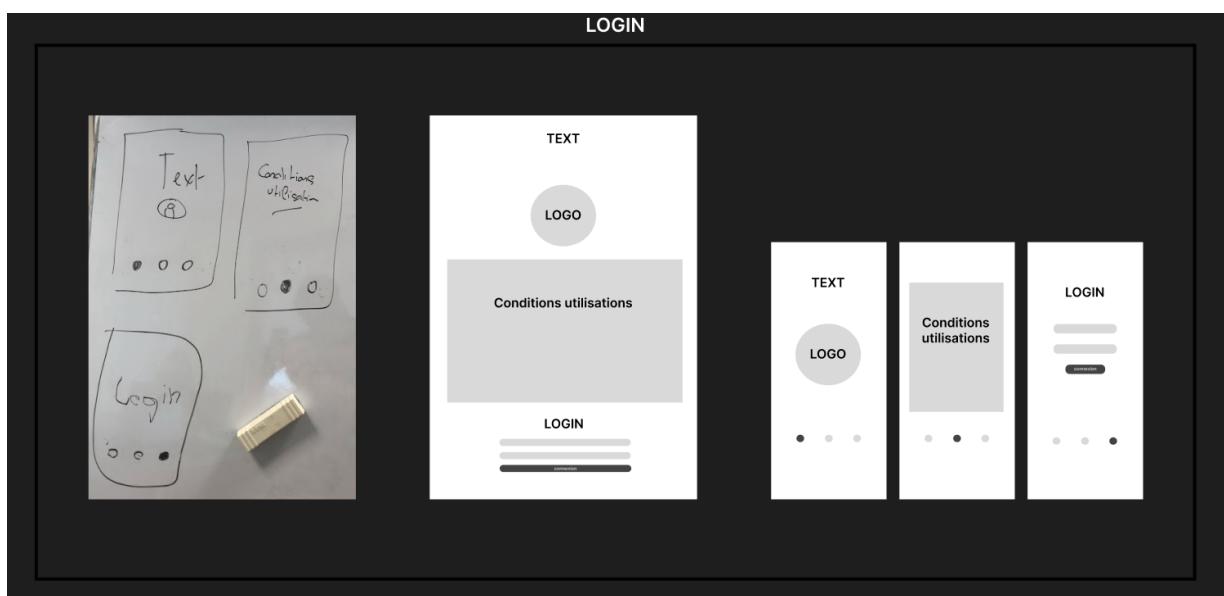
### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre d'une collaboration avec l'association le Sel de la vie, on a pu créer une application mobile accompagnée d'un panel admin.

Nous avons réalisé dans un premier temps des maquettes basse fidélité puis haute fidélité sur **Figma**.

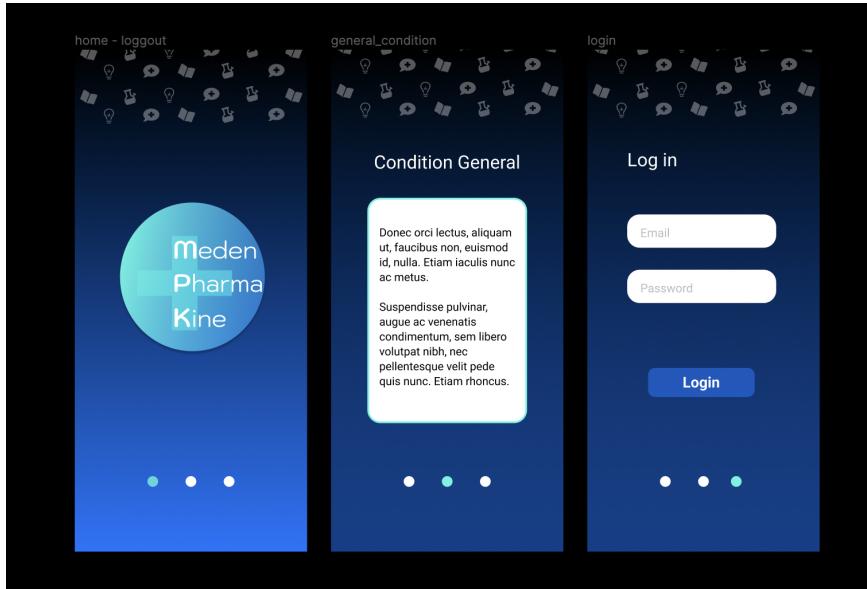
Nous avons également utilisé des outils de prototypage qui ont permis à l'association de se projeter plus facilement en termes de futures fonctionnalités.

#### Maquette basse fidélité :



# DOSSIER PROFESSIONNEL (DP)

## Maquette haute fidélité :



Cet exemple me permet de valider la compétence :

- **Maquetter une application**

## 2. Précisez les moyens utilisés :

Pour réaliser ces maquettes nous avons utilisé Figma

## 3. Avec qui avez-vous travaillé ?

J'ai collaboré avec **Lucien Zak** et **Ifanl Ibrahim** durant ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ➤ *L'association Le Sel De La Vie*

Chantier, atelier, service ➤ *Medenpharmakine*

Période d'exercice ➤ Du : *01/01/2023* au : *31/01/2023*

## 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 jeu du Snake - Développer une interface utilisateur de type desktop*

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

Dans le cadre de ma formation, j'ai développé le jeu du snake en Python.

Pour cela, j'ai utilisé la librairie Pygames, qui me permet de créer des éléments graphiques ainsi que d'écouter sur des événements bien précis comme sur celui des touches du clavier.

Ce projet a été codé en Orienté Objet pour pouvoir au mieux faire interagir mes éléments et une plus grande lisibilité du code.

J'ai commencé par initier le jeu avec la fonction init de Pygame et définir la surface sur laquelle on va jouer. J'ai rajouté une fonction draw\_grid pour dessiner le fond de mon interface afin d'être sur de la position de mon snake et de sa nourriture.

```
● ● ●  
pygame.init()  
  
nbr_col = 10  
nbr_row = 15  
cell_size = 50  
  
screen = pygame.display.set_mode(size=(nbr_col * cell_size, nbr_row * cell_size))  
  
# Fonction pour tracer la grille de mon interface  
def draw_grid():  
    for i in range(0,nbr_col):  
        for j in range(0, nbr_row):  
            rect = pygame.Rect(i * cell_size, j * cell_size, cell_size, cell_size)  
            pygame.draw.rect(screen, pygame.Color('black'), rect, 1)
```

Cette dernière fonction est ensuite appelée dans ma game loop:

```
while game_on:
    # Event pour quitter le jeu en cliquant sur la croix
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_on = False
            sys.exit()
    # Mettre à jour la position du snake et de la nourriture
    if event.type == screen_update:
        game.update()

    # Event pour les flèches du clavier
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            if game.snake.direction != 'RIGHT':
                game.snake.direction = 'LEFT'
        if event.key == pygame.K_RIGHT:
            if game.snake.direction != 'LEFT':
                game.snake.direction = 'RIGHT'
        if event.key == pygame.K_UP:
            if game.snake.direction != 'DOWN':
                game.snake.direction = 'TOP'
        if event.key == pygame.K_DOWN:
            if game.snake.direction != 'TOP':
                game.snake.direction = 'DOWN'

    # Dessiner la grille
    draw_grid()
```

J'ai également créé une classe pour retourner la position de mon bloc et qui possède comme attributs des coordonnées x et y

```
class Block:
    def __init__(self, x_pos, y_pos):
        self.x = x_pos
        self.y = y_pos
```

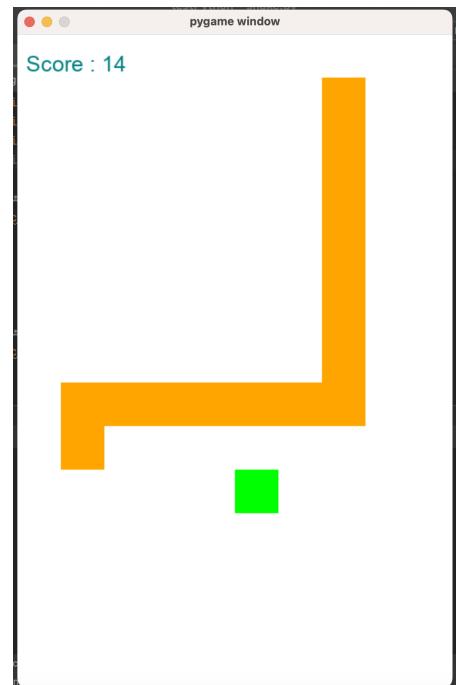
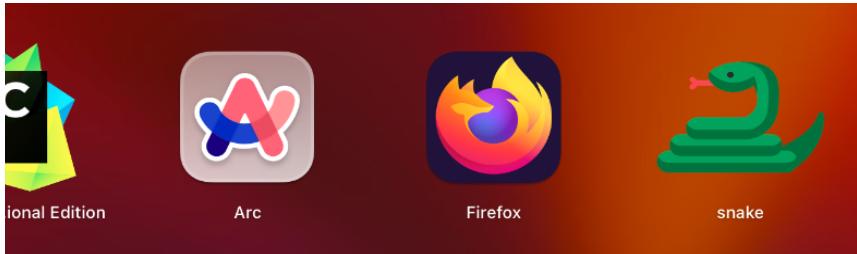
# DOSSIER PROFESSIONNEL (DP)

Je construis également le bloc de nourriture dans une **classe**. Dans le **constructeur**, j'applique aléatoirement une position en utilisant la fonction randint de la librairie **random**. Puis je stocke la position en utilisant une instance de la classe Block. Je crée également une fonction qui me permettra de dessiner la nourriture à la position précédente, retourner et appliquer à l'instance self.block.

```
class Food:
    def __init__(self):
        x = random.randint(0, nbr_col - 1)
        y = random.randint(0, nbr_row - 1)
        self.block = Block(x, y)

    def draw_food(self):
        rect = pygame.Rect(self.block.x * cell_size, self.block.y * cell_size, cell_size, cell_size)
        pygame.draw.rect(screen, pygame.Color('green'), rect)
```

Pour finir, j'ai créé un exécutable en un seul fichier et ne plus passer par la ligne de commande pour lancer mon script, j'ai utilisé la librairie **PyInstaller** et utilisé la commande (le dernier paramètre me sert à lui assigner une icon) : **pyinstaller snake.py --onefile --windowed --icon snake-icon.icns**



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

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- VScode
- Python comme langage
- Pip comme gestionnaire de package
- les librairies : pygame , random sys, pyinstaller

## 3. Avec qui avez-vous travaillé ?

J'ai fait ce projet seule

## 4. Contexte

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

Chantier, atelier, service ▶ *Snake*

Période d'exercice ▶ Du : *07/03/2023* au : *08/03/2023*

## 5. Informations complémentaires (*facultatif*)

## Activité-type 1 Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 3 - Medenpharmakine

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

Dans le cadre du projet pour l'association le Sel de La Vie, nous avons développé un panel Admin web en **NextJs**.

Ce projet valide les compétences :

- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur
- Développer la partie back-end d'une interface utilisateur

Cette interface contient un accès à un **CRUD** pour gérer les étudiants, les cours, les forums avec leurs messages, les thèmes et les QCMs.

On a choisi NextJs parce qu'il nous permet de prendre en charge le rendu côté serveur, la génération statique et le rendu côté client dans une seule application. Il intègre également un routeur, il nous suffit de créer un fichier dans le dossier **page**.

Si on veut aller sur la route "/users", il faut créer un fichier *users.js*. Et quand on veut faire une page élément, par exemple ici, on veut modérer les messages liés à un Forum en allant sur la route *messages/id\_topic*, il va falloir créer un dossier *messages* et dedans un fichier *[id].js*. Grâce à la méthode **getServerSideProps()**, on va pouvoir récupérer *id\_topic* dans l'url. Cette méthode récupère les données pour chaque requête.



```
export async function getServerSideProps(context) {
    return {
        props: {
            id: context.query.id_topic
        }
    }
}
```



```
const [messagesItem, setMessagesItem] = useState([]);

useEffect(() => {
  getTopicMessage(props.id_topic).then((data) => {
    setMessagesItem(data.Messages);
  });
}, []);
```

On va récupérer les messages liés au forum via un appel à l'API et l'afficher côté front. On le fait dans un **useEffect** parce qu'on récupère la donnée en asynchrone puis on la stocke dans un **state**. La fonction côté front API se présente comme tel :

```
import axiosInstance from '../config'

export const getTopicMessage = async (topicID) => {
  const { data } = await axiosInstance.get(`topics/${topicID}?include=Message`)
  return data
}
```

On a créé une instance axios sur une Url précisé dans le fichier de fichier config.js

```
import axios from "axios";

const axiosInstance = axios.create({
  baseURL: process.env.NEXT_PUBLIC_API_URL,
  responseType: "json",
});

axiosInstance.interceptors.request.use(
  function (request) {

    const token = localStorage.getItem("user");
    if(token){
      request.headers.Authorization= `Bearer ${token}`
    }
    return request;

  },
);
export default axiosInstance;
```

# DOSSIER PROFESSIONNEL (DP)

Le point d'entrée de l'API est stocké dans une variable d'environnement, pour optimiser le code et n'avoir qu'un seul endroit où changer les paramètres. De ce fait, on a créé un fichier .gitignore pour éviter d'éventuelles failles de sécurité en le pushant sur github.

Le panel admin se présente comme tel avec différents onglets où l'on peut modérer :

- les étudiants (créer / modifier / supprimer leur compte)
- les cours (créer / modifier / supprimer le course)
- les forums (créer / modifier / supprimer les messages)
- les thèmes (créer / modifier / supprimer les thèmes)
- les QCM (créer / modifier / supprimer les questions)

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- VScode
- Next js
- Node js
- sequelize
- les librairies react utilisées : MUI, Jwt-decode, React-to-print, axios

## 3. Avec qui avez-vous travaillé ?

J'ai collaboré avec **Lucien Zak** et **Ifanl Ibrahim** durant ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *L'association Le Sel De La Vie*

Chantier, atelier, service ▶ *Medenpharmakine*

Période d'exercice ▶ Du : *01/01/2023* au : *31/01/2023*

## 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

## Liste des étudiants

ID	Prénom	Nom	E-mail	Rôle	Photo de profil	Date de création	Date de modification	Modifier	Supprimer
1	admin	admin	admin@admin.com	ROLE_ADMIN ROLE_STUDENT		25/01/2023	13/02/2023 10:21:49	<button>MODIFIER</button>	<button>SUPPRIMER</button>
2	user	user	user@user.com	ROLE_TUTOR		25/01/2023	25/01/2023 15:52:02	<button>MODIFIER</button>	<button>SUPPRIMER</button>
3	user2	user2	user2@user2.com	ROLE_STUDENT		25/01/2023	26/01/2023 11:39:38	<button>MODIFIER</button>	<button>SUPPRIMER</button>

## Ajouter un utilisateur

Prénom	<input type="text"/>
Nom	<input type="text"/>
E-mail	<input type="text"/>
Mot de passe	<input type="password"/>
Confirmer le mot de passe	<input type="password"/>
Rôle	<input type="text"/>

ENREGISTRER ANNULER

## Activité-type 2 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

*Exemple n° 1 - Medenpharmakine*

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

Après avoir pris en note dans le cahier des charges les besoins de l'association (voir exemple 1), il a été décidé de réaliser une application en **React Native** qui serait composée :

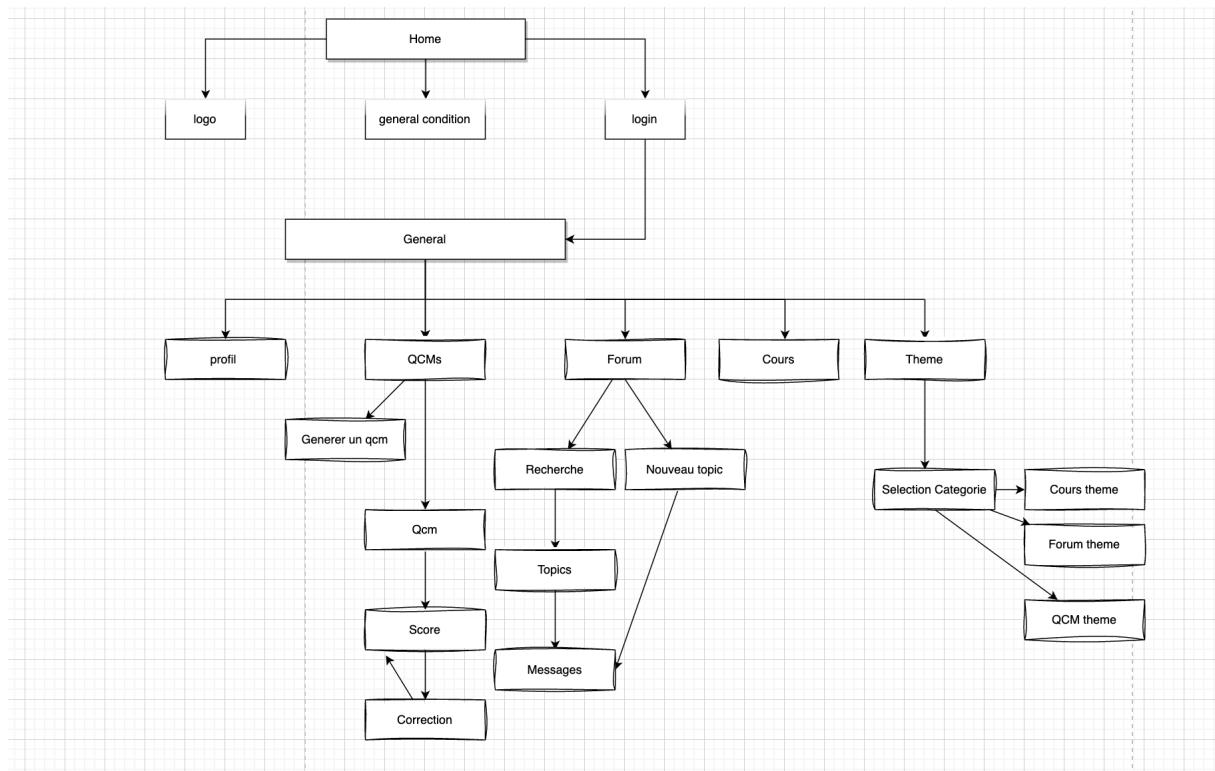
- d'un espace de connexion
- d'un espace profil
- d'un espace forum
- d'un espace cours
- d'un espace qcm
- d'un espace statistique

Ce projet me permet de valider les compétences :

- **Développer une application mobile**
- **Développer des composants métier**
- **Concevoir une application**
- **Construire une application organisée en couches**

Nous avons architecturé l'application mobile comme représenté dans le schéma suivant :

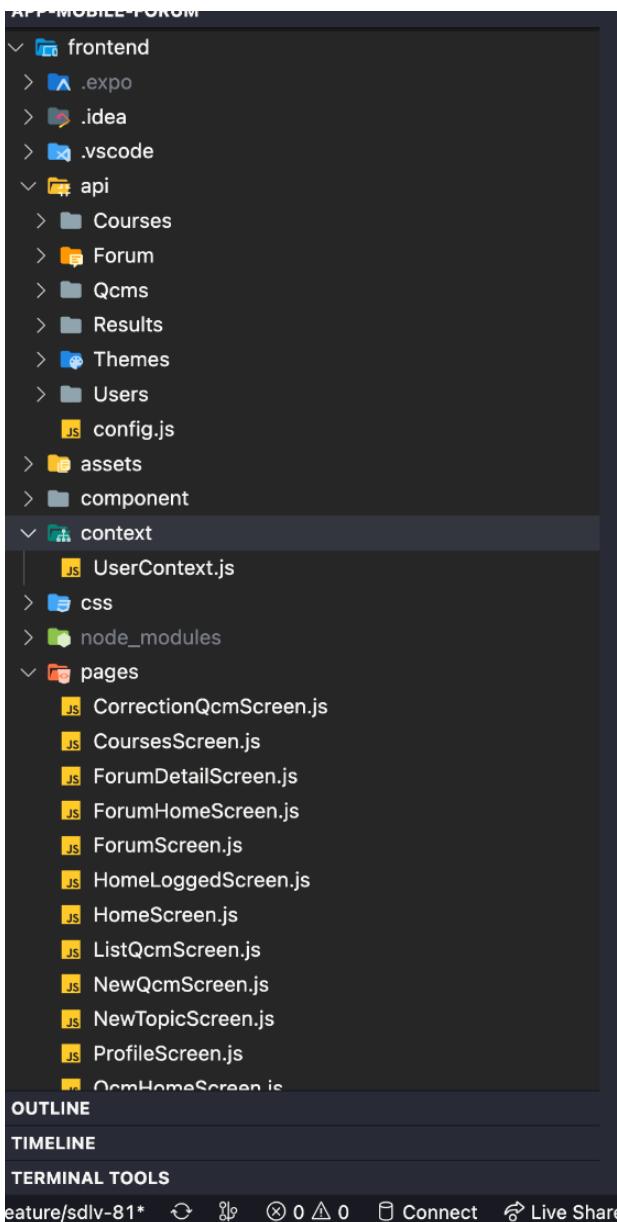
# DOSSIER PROFESSIONNEL (DP)



On ne peut pas accéder au reste de l'application si l'on n'est pas connecté.

Nous avons utilisé **Expo**, qui est à la fois un **framework** et une plateforme qui simplifient la création et le déploiement d'applications mobiles avec **React Native**. Expo embarque de nombreux outils utiles et des librairies natives pour React Native. Il gère aussi la mise à jour de ces librairies.

# DOSSIER PROFESSIONNEL (DP)



Côté Frontend, nous avons créé différents dossiers afin de mieux nous organiser . Dans le dossier **api**, on aura tout ce qui est appelé api qui communique avec la base de données. Dans le fichier config, on a mis en place une instance **Axios** qui nous permet :

- de personnaliser les paramètres pour toutes les requêtes effectuées à partir de cette instance,
- d'éviter la répétition de code et de rendre notre code plus lisible.

```
import axios from 'axios'
import * as SecureStore from 'expo-secure-store';
import API_IP from '../env.js'

const axiosInstance = axios.create({
  baseURL: `${API_IP}/api/`, responseType: 'json',
})

axiosInstance.interceptors.request.use(function (request) {
  SecureStore.getItemAsync('token').then((token) => {
    request.headers.Authorization = `Bearer ${token}`
  })
  return request
}, function (error) {
  console.log(error);
  return Promise.reject(error)
})

export default axiosInstance
}
```

Par le biais du **Secure Store** d'expo, on a pu stocker le token d'authentification et ainsi le passer en entête de nos requêtes. Les requêtes sont faites avec la librairie Axios comme expliqué dans l'exemple précédent. On a également mis en place **React Context** qui nous a permis de rendre les données de l'utilisateur accessibles à l'ensemble des composants de notre application.

```
export const UserContext = createContext();

const UserContextProvider = (props) => {
  const [token, setToken] = useState(null);
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const decodedToken = token ? jwtDecode(token) : null;
```

Dans un premier temps, on déclare un UserContext avec la fonction *createContext()* de react pour

pouvoir utiliser le context provider et faire passer les informations de notre utilisateur.

```
<UserContext.Provider  
  value={{userDetails, setUserDetails, userAuthentication, isLoggedIn, setIsLoggedIn, userLogout}}>  
>  
  {props.children}  
</UserContext.Provider>
```

On passe également en paramètre du provider :

- la variable isLoggedIn qui nous permet de savoir si l'utilisateur est connecté à partir du token d'authentification
- la fonction setUserDetails qui nous sert à mettre à jour le context depuis d'autres pages.
- userAuthentication qui permet à l'utilisateur de se connecter

## Exemple de composants métier

```
<LinearGradient  
  colors={[ "purple", "#02254F", "#2D84EA" ]}  
  style={styles.containerGradient}  
>  
  <Image source={require("../assets/logo_fond.png")} style={styles.bgTop} />  
  <View style={[ styles.profileImgContainer ]}>...  
  </View>  
  <Text style={styles.sectionTitle}>Mes derniers cours</Text>  
  <ScrollView style={{ display: "flex", flexDirection: "column" }}>  
    <View style={styles.sectionLatestCourse}>  
      {courses.length !== 0 &&  
       courses.map((course, i) => {  
         return <CourseCardComponent key={i} course={course} />;  
       })}  
    </View>  
  </ScrollView>  
</LinearGradient>
```

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- VScode
- React Native

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

- Node js
- sequelize
- Expo
- les librairies react utilisées : react native paper, Jwt-decode, axios,

## 3. Avec qui avez-vous travaillé ?

J'ai collaboré avec **Lucien Zak** et **Ifanl Ibrahim** durant ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ➤ *L'association Le Sel De La Vie*

Chantier, atelier, service ➤ *Medenpharmakine*

Période d'exercice ➤ Du : *01/01/2023* au : *31/01/2023*

## 5. Informations complémentaires (*facultatif*)

## Activité-type 3

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

*Exemple n° 1 - Medenpharmakine*

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

Pour ce projet, nous avons créé une base de données pour répondre à la demande du client.

Ce projet valide les compétences :

- **Concevoir une base de données**
- **Mettre en place une base de données**
- **Développer des composants dans le langage d'une base de données**

Sur cette base de données, on va pouvoir enregistrer les différentes informations liées :

- à l'utilisateur
- aux cours
- aux QCM et questions liées
- aux qcm répondus par l'utilisateur
- aux thèmes
- au forum et sujet associés
- aux messages des sujet

On a utilisé node Js ainsi que la librairie sequelize qui prend en charge MySQL et offre une prise en charge solide des transactions et des relations. Sequelize est un ORM ("Object Relational Mapping") qui sert à mettre à disposition des classes objet permettant de manipuler les bases de données relationnelles.

Nous nous sommes appuyés sur la méthode "**Merise**" pour concevoir notre base de données SQL.

On a commencé par réaliser sur l'outil **draw.io**, le **MCD** (modèle conceptuel de données) et enfin le **MLD** (modèle logique de données) comme on peut le voir en Annexe 1 et 2.

```
const { Sequelize, DataTypes } = require('sequelize')
const sequelize = new Sequelize('mysql::memory')

module.exports = (sequelize) => {
  const User = sequelize.define('User', {
    // Model attributes are defined here
    firstName: {
      type: DataTypes.STRING,
      allowNull: false
    },
    lastName: {
      type: DataTypes.STRING
      // allowNull defaults to true
    },
  },

```

Dans le dossier model, on retrouve un fichier pour chaque table comme par exemple pour ce fichier user.

On définit chaque champ et lui spécifie son type et s'il peut être nul en base de données.

```
const sequelize = new Sequelize({
  dialect: 'mysql',
  username: process.env.DB_USERNAME,
  password: process.env.DB_PASSWORD,
  host: process.env.DB_HOST,
  port: process.env.DB_PORT,
  database: process.env.DB_NAME,
});


```

Pour se connecter à la base de données, on a dû créer une instance Sequelize et on lui passe en paramètre les informations de connection qui sont dans notre .env

```
function association(sequelize){
  const {User, Qcm, UserQcm, Topic, Message, Question, sequelize.models;

  UserhasMany(Message, {foreignKey: 'id_user'});
  TopichasMany(Message, {foreignKey: 'id_topic'});
  Message.belongsTo(Topic, {foreignKey: 'id_topic'});

  Message.belongsTo(User, {foreignKey: 'id_user'});
  UserhasMany(UserQcm, {foreignKey: 'id_user'});

  QcmhasMany(UserQcm, {foreignKey: 'id_qcm'});
  UserQcm.belongsTo(Qcm, {foreignKey: 'id_qcm'});
}


```

Sequelize nous permet de faire des associations.

Il ya 4 types d'associations :

- **HasOne**
- **BelongsTo**
- **HasMany**
- **BelongsToMany**

L'étape MCD / MLD nous a aidé pour repérer plus facilement quel type de relation était le plus adapté en fonction

des cardinalités .

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

J'ai utilisé :

- VScode
- Node js
- sequelize
- MySQL
- draw.io

## 3. Avec qui avez-vous travaillé ?

J'ai collaboré avec **Lucien Zak** et **Ifanl Ibrahim** durant ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ *L'association Le Sel De La Vie*

Chantier, atelier, service ▶ *Medenpharmakine*

Période d'exercice ▶ Du : *01/01/2023* au : *31/01/2023*

## 5. Informations complémentaires (*facultatif*)

## Activité-type 2 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 2 - Portfolio

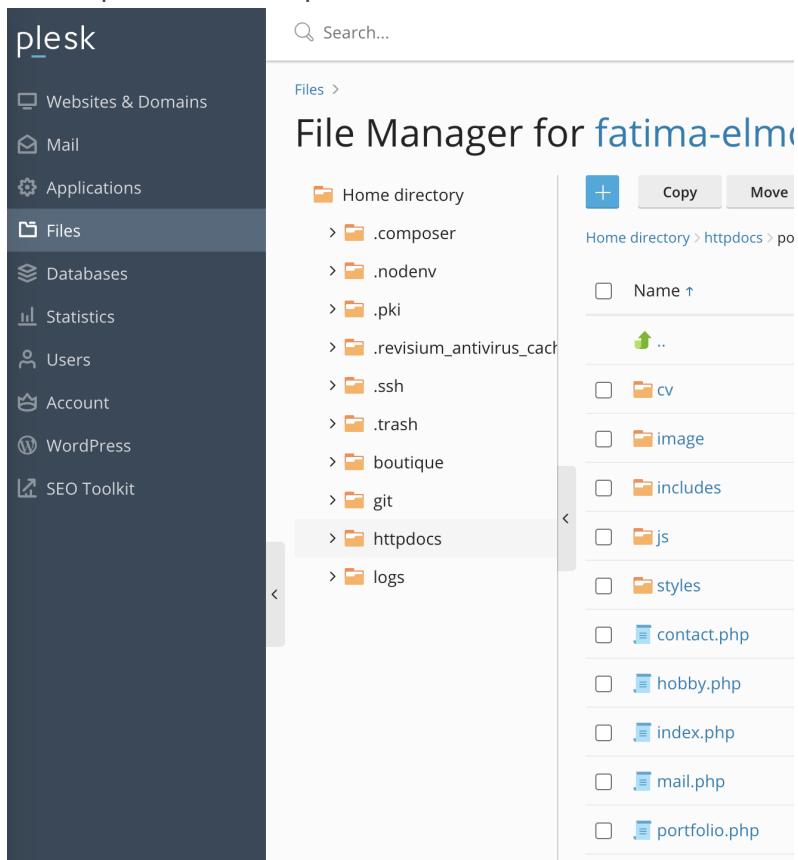
### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Dans le cadre de ma recherche d' alternance, j'ai pu réaliser un portfolio que j'ai pu déployé sur un hébergeur (plesk)

Cet exemple me permet de valider la compétence :

- Préparer et exécuter le déploiement d'une application

Je mets dans un premier temps mon dossier portfolio directement sur le serveur > dossier httpdocs.



# DOSSIER PROFESSIONNEL (DP)

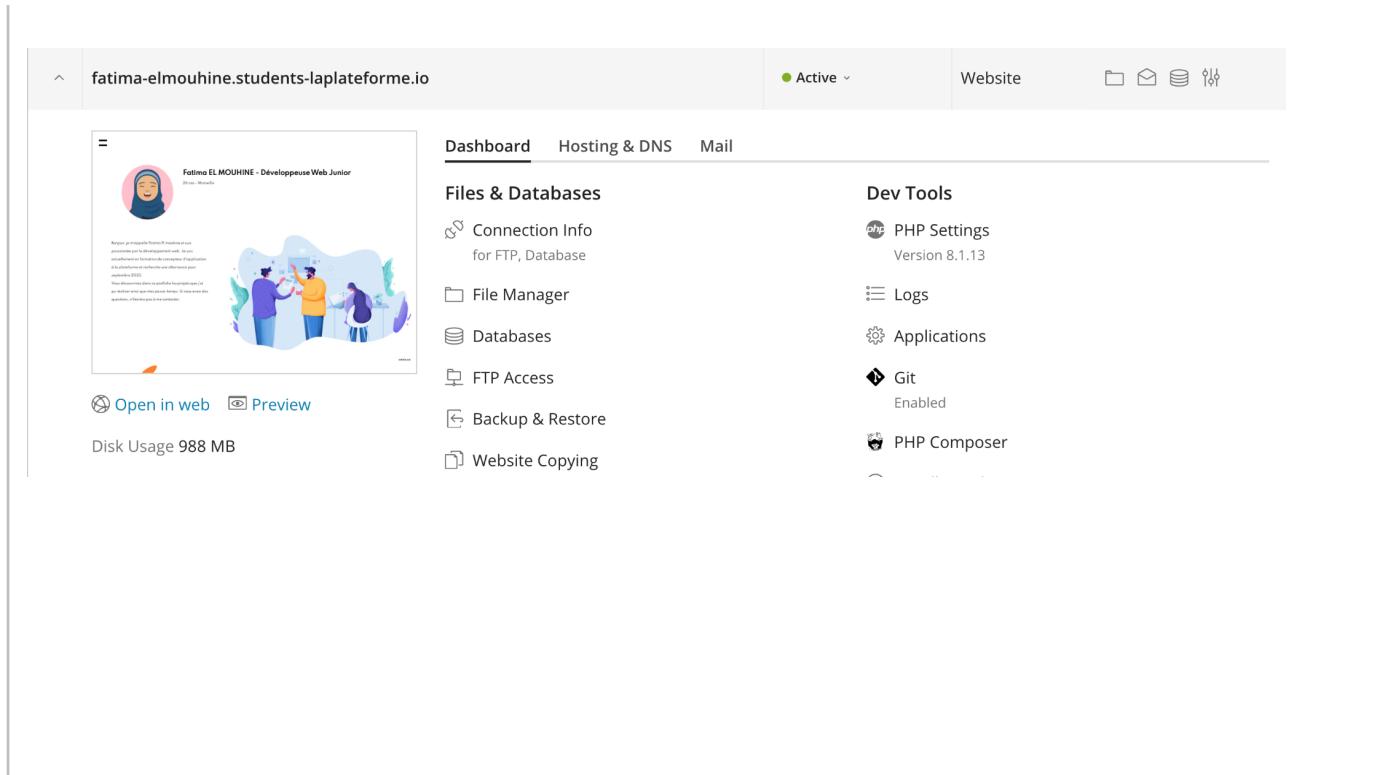
Ensuite, j'ajoute ma base de données MySQL via l'onglet **Database**. J'ai créé un utilisateur pour me connecter. Puis j'ai changé les informations de connexion dans **mes variables d'environnement**.

The screenshot shows the Plesk control panel. On the left, the sidebar includes options like Websites & Domains, Mail, Applications, Files, Databases (which is selected), Statistics, Users, Account, WordPress, and SEO Toolkit. The main area is titled 'Databases > Add a Database'. It has two sections: 'General' and 'Users'. In 'General', the 'Database name' is set to 'fatima-elmouhine\_portfolio', 'Database server' is 'localhost:3306 (default for MariaDB, v5.5.68)', and 'Related site' is 'fatima-elmouhine.students-laplateforme.io'. In 'Users', there's a checked checkbox for 'Create a database user'. Below it, 'Database user name' is 'fatima-elmouhine', 'Password' is masked, and 'Confirm password' is also masked. A note says 'User has access to all databases within the selected subscription'. At the bottom are 'OK' and 'Cancel' buttons.

J'indique le point d'entrée de mon domaine, mon site est maintenant en ligne

The screenshot shows the Plesk control panel. The sidebar includes Websites & Domains, Mail, Applications, Files, Databases, Statistics, Users, Account, WordPress, and SEO Toolkit. The main area is titled 'Websites & Domains > Hosting Settings for fatima-elmouhine'. It says 'This is where you configure website hosting settings and select the features for your website.' It shows 'Hosting type: Website', 'Website status: Active [Change]', and 'Document root: httpdocs/portfolio'. Under 'Preferred domain', the radio button for 'fatima-elmouhine.students-laplateforme.io' is selected. There's also a 'None' option and a note about selecting a URL with or without www. At the bottom are 'OK' and 'Cancel' buttons.

# DOSSIER PROFESSIONNEL (DP)



## 2. Précisez les moyens utilisés :

J'ai utilisé :

- plesk

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seule sur ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ➔ *La plateforme*

Chantier, atelier, service ➔ *Formation*

Période d'exercice ➔ Du : *20/09/2022* au : *30/10/2022*

## 5. Informations complémentaires (facultatif)

# DOSSIER PROFESSIONNEL (DP)

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

*(facultatif)*

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

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

## Déclaration sur l'honneur

Je soussigné(e) Fatima EL MOUHINE , 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 30/05/23  
pour faire valoir ce que de droit.

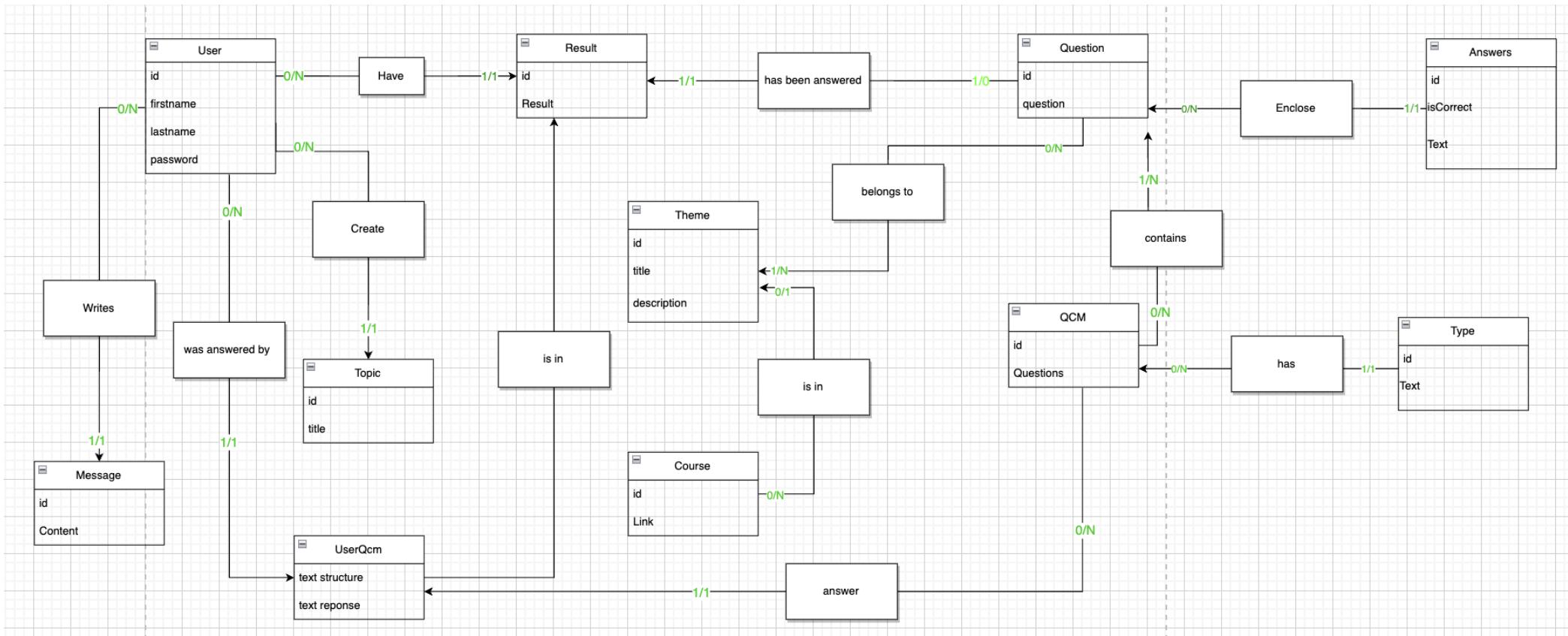
Signature :

**Fatima El mouhine**

# DOSSIER PROFESSIONNEL (DP)

## ANNEXES

### Modèle Conceptuel de Données



# DOSSIER PROFESSIONNEL (DP)

## Modèle Logique de Données

