



# DOSSIER PROFESSIONNEL (DP)

|                  |                           |
|------------------|---------------------------|
| Nom de naissance | ▶ ZIDI                    |
| Nom d'usage      | ▶ ZIDI                    |
| Prénom           | ▶ Mohamed                 |
| Adresse          | ▶ 3, Rue François Massabo |

## Titre professionnel visé

*Concepteur développeur d'applications*

### MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

# DOSSIER PROFESSIONNEL (DP)

## 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.*



## 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>6</b>  |
| ▶ Maquetter une application  | p.         | 6         |
| ▶ Développer une interface utilisateur de type desktop   | p.         | 10        |
| ▶ Développer des composants d'accès aux données  | p.         | 14        |
| ▶ Développer la partie front-end d'une interface utilisateur web   | p.         | 17        |
| ▶ Développer la partie back-end d'une interface utilisateur web  | p.         | 20        |
| <b>Concevoir et développer la persistance des données en intégrant les recommandations de sécurité</b>             | <b>p .</b> | <b>25</b> |
| ▶ Concevoir une base de données  | p.         | 25        |
| ▶ Mettre en place une base de données  | p.         | 14        |
| ▶ Développer des composants dans le langage d'une base de données  | p.         | 14        |
| <b>Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité</b>   | <b>p .</b> | <b>29</b> |
| ▶ Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement         | p.         | 29        |
| ▶ Concevoir une application mobile   | p.         | 33        |
| ▶ Développer des composants métier   | p.         | 17        |
| ▶ Construire une application organisée en couches  | p.         | 33        |
| ▶ Développer une application mobile  | p.         | 33        |
| ▶ Préparer et exécuter les plans de tests d'une application  | p.         | 39        |
| ▶ Préparer et exécuter le déploiement d'une application  | p.         | 42        |

---

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

---

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

p  
.

|    |
|----|
| 46 |
|----|

**Déclaration sur l'honneur**

p  
.

|    |
|----|
| 47 |
|----|

**Documents illustrant la pratique professionnelle** *(facultatif)*

p  
.

|    |
|----|
| 48 |
|----|

**Annexes** *(Si le RC le prévoit)*

p  
.

|    |
|----|
| 49 |
|----|

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

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

Exemple n°1 ► Maquetter une application

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

Avant de commencer le développement de l'application mobile, l'une des étapes primordiales est de faire une maquette des écrans de celle-ci.

Maquetter l'application en amont permet plusieurs choses :

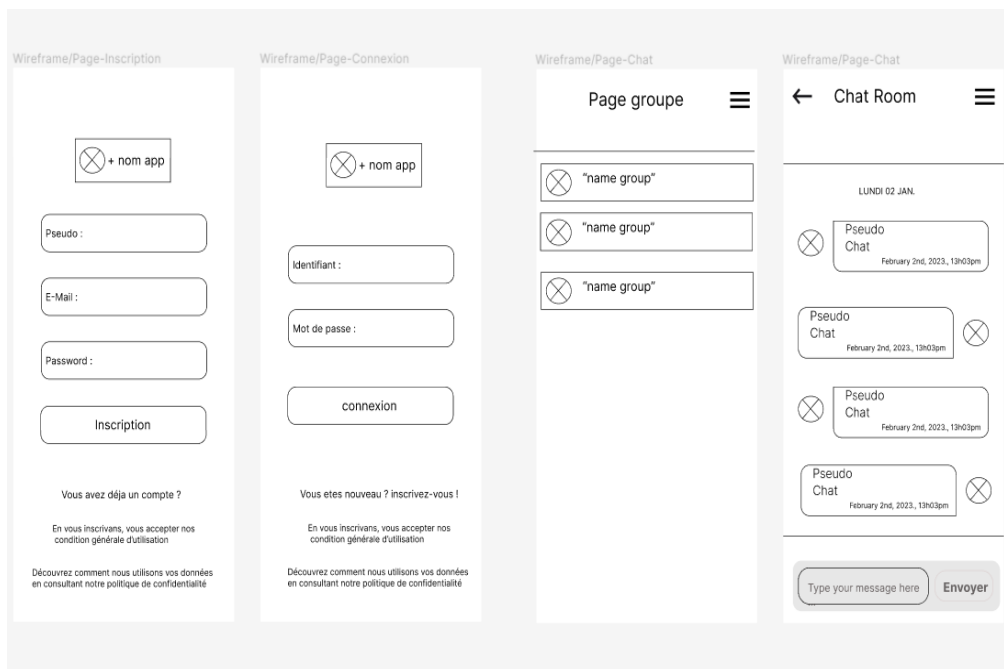
Visualisation du concept : Une maquette permet de donner vie à l'idée et de visualiser le concept global du projet. Cela permet de mieux comprendre les fonctionnalités et l'organisation des éléments à l'écran.

Communication et collaboration : facilite la communication entre les membres de l'équipe de développement, les designers et les clients. Elle sert de point de référence visuel et favorise la collaboration en permettant à chacun de partager et d'exprimer ses idées de manière concrète.

Validation du design : Une maquette permet de tester et de valider le design des écrans avant de les implémenter. Cela permet de détecter les éventuelles erreurs de conception, d'apporter des modifications et d'optimiser l'expérience utilisateur en amont, ce qui peut faire gagner du temps et éviter des retours en arrière coûteux.

Collecte de feedback : Une maquette peut être présentée aux utilisateurs ou aux parties prenantes du projet pour recueillir leurs commentaires et leurs suggestions. Cela permet d'identifier les améliorations nécessaires et d'ajuster le design en fonction des besoins et des attentes des utilisateurs finaux.

# DOSSIER PROFESSIONNEL (DP)



Avons d'avoir la maquette finale (design), mes collaborateurs sur le projet et moi-même avons réalisé une maquette wireframe (ci-dessus).

Ci-dessous une partie de la maquette finale avant le lancement du développement (écrans connexion et inscription utilisateur) :

# DOSSIER PROFESSIONNEL (DP)

The wireframe displays two side-by-side panels for a user interface. The left panel, titled 'Page Inscription', contains a yellow circular logo with a speech bubble icon and the text 'NEKO'. Below the logo are four input fields: 'Pseudo :', 'E-mail :', and 'Mot de passe :'. An orange 'Inscription' button is positioned below these fields. At the bottom of the panel, there is a link 'Vous avez déjà un compte ?' and two lines of small text: 'En vous inscrivant, vous acceptez nos conditions générales d'utilisation' and 'Découvrez comment nous utilisons vos données en consultant notre politique de confidentialité'. The right panel, titled 'Page Connexion', features the same 'NEKO' logo. It includes two input fields: 'Identifiant :' and 'Mot de passe :'. An orange 'Connexion' button is located below the 'Mot de passe :' field. Similar to the left panel, it has a link 'Vous êtes nouveau ? inscrivez-vous !' and the same two lines of small text at the bottom.

## 2. Précisez les moyens utilisés :

Un ordinateur ainsi qu'une connexion internet.  
L'outil figma a été utilisé pour maquetter le projet.  
Trello pour se définir les tâches et partager le travail

## 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf  
Mohamad Sheiko  
Chaïma Aicheche  
Khellaf Rachedi



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

## 4. Contexte

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

Chantier, atelier, service ► *Dans le cadre de la formation concepteur/ développeur d'application*

Période d'exercice ► Du : 02/01/23 au : 10/01/23

## 5. Informations complémentaires *(facultatif)*

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

Exemple n° 2 ► Développer une interface utilisateur de type desktop

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

Au cours de mon année, l'un des projets obligatoires à réaliser était de développer un jeu en python. Bien sûr j'ai pu m'exercer pendant une semaine durant la runtrack prévue à cet effet pour mieux appréhender ce langage.

J'ai eu l'idée de développer un shooter en 2D dans lequel notre personnage aura à faire face à des adversaires et devra esquiver des comètes pour rester en vie le plus longtemps possible.

Le joueur pourra se déplacer ainsi que tirer des projectiles pour éliminer ses adversaires. Chaque adversaire tué lui apportera des points cumulés pour augmenter son score, si un ennemi ou une comète atteint le joueur celui perdra des points de vie. Si sa barre de vie atteint zéro alors ce sera le game over.

Pour développer ce projet j'ai utilisé la bibliothèque gratuite et open source pygame.

Gestion de la fenêtre du jeu :

```
pygame.init()

# generer la fenetre
pygame.display.set_caption("Shooter game")
screen = pygame.display.set_mode((1080, 720))
```

J'ai développé mon jeu en utilisant la programmation orientée objet, j'ai une classe qui gère les actions du joueur, une pour les monstres etc.

---

## DOSSIER PROFESSIONNEL (DP)

---

```
def launch_projectile(self):  
    # instancier classe projectile  
    projectile = Projectile(self)  
    self.all_projectiles.add(projectile)  
    # demarrer animation du lancer  
    self.start_animation()  
    self.sound.play("tir")
```

Méthode de la classe Player permettant de gérer le tir de projectile

```
def fall(self):  
    self.rect.y += self.velocity  
  
    if self.rect.y >= 500:  
        self.comet_event.destroy(self)  
  
    # Verifier si la boule de feu touche le joueur  
    if self.comet_event.game.check_collision(self, self.comet_event.game.all_players):  
        # supprimer la boule de feu  
        self.comet_event.destroy(self)  
        # subir 20 points de degats  
        self.comet_event.game.player.damage(20)
```

Methode fall de la classe Comet dans laquelle je gère la chute des comètes ainsi que la collision avec le sprite du joueur.

## DOSSIER PROFESSIONNEL (DP)



Rendu final du jeu.

### 2. Précisez les moyens utilisés :

Un ordinateur avec connexion internet  
Pycharm comme IDE.  
Le langage de programmation python  
Bibliothèque open source et gratuite pygame  
Documentation python  
Documentation pygame

### 3. Avec qui avez-vous travaillé ?

J'ai effectué ce projet seul

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

## 4. Contexte

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

Chantier, atelier, service ► *Dans le cadre de ma formation concepteur/développeur d'application*

Période d'exercice ► Du : *17/04/23* au : *21/04/23*

## 5. Informations complémentaires *(facultatif)*

## Activité-type 1

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

Exemple n° 3 ► Développer des composants d'accès aux données

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

Dans le cadre de la conception du projet application mobile "Neko", nous avons décidé d'utiliser Node.js avec Express et MongoDB car l'an dernier nous avons surtout travaillé avec des bases de données relationnelles, donc voulions découvrir les bases de données NoSql pour en apprendre le plus possible sur les différents types de bases de données.

Express est un framework web pour Node.js qui facilite la création d'applications web et d'API REST. Il offre une structure légère et flexible, permettant de gérer les routes, les requêtes HTTP et les réponses de manière efficace.

Pour la persistance des données, nous avons utilisé MongoDB, une base de données NoSQL orientée document. MongoDB offre une flexibilité importante dans la modélisation des données, ce qui est adapté aux besoins évolutifs de notre application.

L'ORM (Object-Relational Mapping) utilisé avec Node.js et MongoDB est Mongoose. Ce dernier permet de définir des schémas de données, de valider les entrées et d'interagir avec la base de données de manière simple et intuitive.

Ainsi, nous avons pu mettre en place notre API REST en utilisant Express et MongoDB avec l'aide de Mongoose pour la gestion de la persistance des données. Cette combinaison nous a permis de développer rapidement et efficacement notre application mobile "Neko" en exploitant les avantages de Node.js et MongoDB.

Ligne de commande permettant de créer un fichier package :

```
Npm init
```

Ligne de commande pour démarrer un projet express :

```
Npm install express
```

Ligne de commande pour initialiser mongoose :

Npm install mongoose

```
1  const mongoose = require('mongoose');
2
3  const connectDb = async () => {
4
5      try {
6          const conn = await mongoose.connect(process.env.MONGO_URI);
7          console.log(`mongodb connected: ${conn.connection.host}`.cyan.underline);
8      } catch (error) {
9          console.log(error);
10         process.exit(1)
11     }
12 }
13
14 module.exports = connectDb
```

Code gérant la connexion à la base données mongoDB.

```
1  const mongoose = require('mongoose');
2
3  const messageSchema = mongoose.Schema({
4      user: {
5          type: mongoose.Schema.Types.ObjectId,
6          required: true,
7          ref: 'User',
8      },
9      userPseudo: {
10         type: mongoose.Schema.Types.String,
11         required: true,
12         ref: 'User',
13     },
14     text: {
15         type: String,
16         required: [true, "Please add a text value"]
17     },
18     isValid: {
19         type: String,
20         default: 'true',
21     },
22 }, {
23     timestamps: true
24 })
25
26
27
28 module.exports = mongoose.model('Message', messageSchema)
```

Définition du schéma de modèle utilisé par le module mongoose pour l'entité « message ». La création de ce modèle permet de réaliser des opérations CRUD sur les documents « message » dans la base de données mongoDB.

En utilisant Mongoose et MongoDB, l'approche de ma base de données est plutôt "code first" (ou "schema first"), cela signifie que je défini initialement les schémas des documents des collections formant ma base de données dans mon code à l'aide de Mongoose.

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

## 2. Précisez les moyens utilisés :

Un ordinateur avec une connexion internet.  
Un IDE (Visual Studio Code).  
Git pour le versionning.  
GitHub comme plateforme de contrôle de version de git afin de pouvoir les utiliser en collaboration.  
Symfony, Api-platform.  
Trello pour la gestion de projet.  
Documentation MongoDB, express, mongoose.  
<https://www.mongodb.com/docs/>  
<https://expressjs.com/fr/>  
<https://mongoosejs.com/docs/guide.html>

## 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf  
Mohamad Sheiko  
Chaïma Aicheche  
Khellaf Rachedi

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_*  
Chantier, atelier, service ► *Dans le cadre de ma formation concepteur/développeur d'application.*  
Période d'exercice ► Du : *02/01/2023* 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

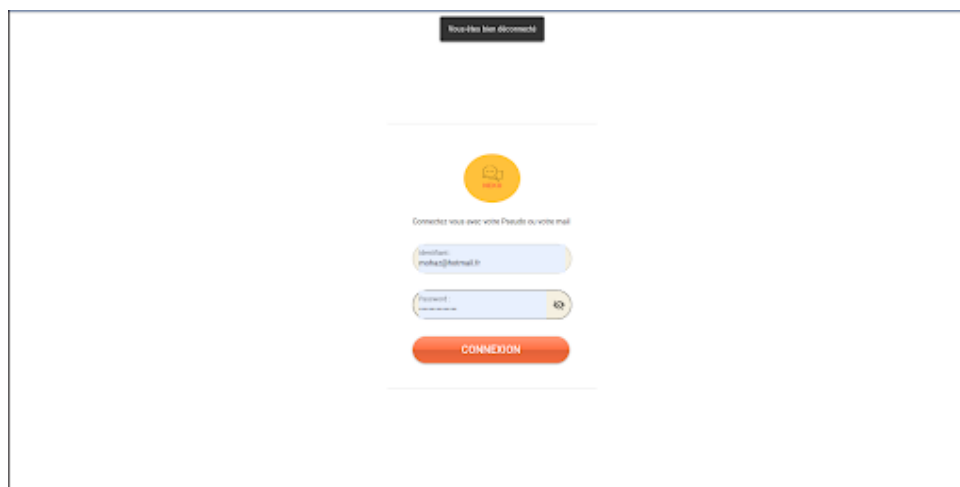
**Exemple n° 4** ► Développer la partie front-end d'une interface utilisateur web

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

L'un des projets les plus importants de cette année est de développer d'une interface administrateur web spécialement conçue pour gérer et contrôler mon application mobile « Neko ».

Pour développer la partie front end de cette interface, mes collaborateurs et moi-même avons fait le choix d'utiliser Quasar qui est un framework open-source basé sur Vue.js qui permet de développer des applications web et mobiles à l'aide de JavaScript ou de TypeScript.

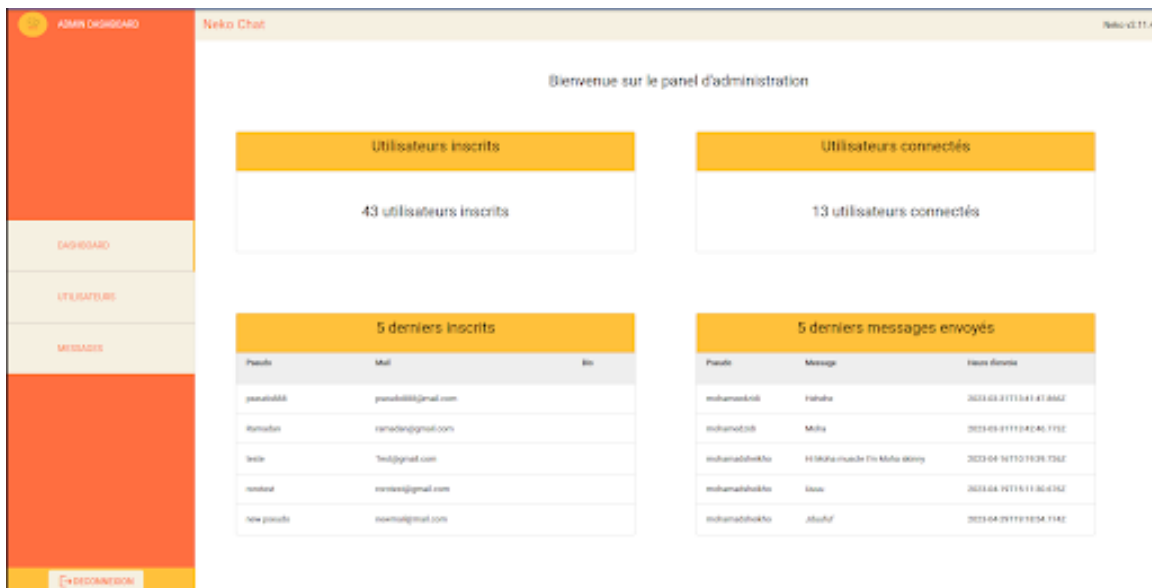
Personnellement j'ai choisi d'utiliser Quasar/Vue.js car premièrement ce framework est souvent vu comme un choix idéal pour les petites à moyennes applications. Deuxièmement, après avoir étudié react au cours de ma formation et angular en entreprise je souhaitais également faire un projet en Vue.js, j'ai trouvé que c'était une bonne occasion de monter en compétences sur cette stack et ainsi de compléter un minimum de compétences sur les frameworks front end les plus populaires sur le marché actuellement.



Page de connexion au panel admin

# DOSSIER PROFESSIONNEL (DP)

Au niveau des fonctionnalités de cette interface, un utilisateur possédant les droits administrateurs sur mon application pourra gérer les utilisateurs inscrits sur l'application (gestion des droits de ceux-ci, bannissement d'utilisateurs malveillants, voir le nombre d'utilisateurs inscrits et connectés), il aura aussi une main sur la gestion de contenu de l'application.



## Page d'accueil

Pour récupérer des données depuis la base de données, la communication entre le front et le back de mon application, qui est géré par une seule et même api, se fait en utilisant axios. C'est une bibliothèque javascript fonctionnant comme un client http, cela permet de créer des requêtes avec des méthodes (GET, POST, PUT et DELETE).

## 2. Précisez les moyens utilisés :

Un ordinateur disposant d'une connexion internet

Un IDE ( visual studio code)

Git et Github pour le versionning

Trello pour le travail collaboratif

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

Quasar

Vue.js

### 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf

Mohamad Sheiko

Chaïma Aicheche

Khellaf Rachedi

### 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_.*

Chantier, atelier, service ► *Dans le cadre de ma formation concepteur/développeur d'application.*

Période d'exercice ► Du : *02/01/2023* 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° 5* ► Développer la partie back-end d'une interface utilisateur web

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

Dans le périmètre du projet d'application mobile, j'avais à développer avec d'autres étudiants de ma promotion une API REST qui peut être consommée par mon application mobile et également par l'interface administrateur web.

Nous avons le choix entre deux stack, php et symfony ou bien nodeJs et express, notre projet s'est tourné vers l'utilisation du framework javascript car après des recherches poussées nous avons trouvé que nodeJs/Express est souvent utilisé pour des cas d'utilisation et une manipulation intensive de données en temps réel, et nous voulions également être libre d'organiser notre code comme on le voulait et express offre une grande flexibilité et modularité au niveau de l'architecture du projet.

Une fois le choix effectué nous pouvions commencer le développement, premièrement il y a l'initialisation du projet :

Créer un nouveau répertoire pour votre projet et initialisez-le en utilisant npm (Node Package Manager) avec la commande `npm init`. Cela créera un fichier `package.json` qui contiendra les dépendances et les informations de configuration de votre projet nécessaires pour le projet.

Ensuite il faut installer express :

En utilisant la commande `npm install express` cela permettra de l'installer et de l'ajouter aux dépendances de mon projet.

Une fois ces deux étapes réalisées il faut créer le fichier qui servira de point d'entrée de mon application, dans mon cas ce sera server.js

```
const express = require('express');  
const app = express();
```

Maintenant je peux commencer à définir mes routes qui seront appelées par mon application mobile et mon panel admin :

```
router.route('/').post(registerUser).get(protect, getAllUsers);  
router.route('/:id').get(protect, getUser).delete(protect, deleteUser);  
router.route('/admin/:id').delete(protect, deleteUserAdmin).put(protect, updateUserAdmin);  
router.post('/login', login);  
router.put('/logout', protect, logout);
```

Sur l'exemple ci-dessus, j'ai au préalable créé une nouvelle instance du router grâce à celui-ci je peux définir des endpoints (URI) qui peuvent accepter différentes requêtes http et ensuite permettent d'effectuer pour chacun une exécution d'une fonction d'un de mes controllers qui contient la logique métier et qui est responsable des échanges avec ma BDD (certaines de ces URI/routes sont protégées par un middleware d'authentification).

Par exemple, la route qui termine par « /login » accepte uniquement une requête http POST contrairement à la route racine terminant par « / » qui accepte les requêtes POST et GET (cette dernière est protégée par un middleware), une fois appelée par le client cette route appelle la fonction login de mon controller qui gère les utilisateurs.

**Middleware** : Un middleware est une fonction ou un ensemble de fonctions dans une application web qui s'exécute entre la réception d'une requête du client et l'envoi d'une réponse par le serveur. Il agit comme une couche intermédiaire entre le client et le serveur, traitant et manipulant les requêtes avant qu'elles n'atteignent leur destination finale.

```
// @desc Recupérer un user
// @route GET /api/users/:id
// @access private
const getUser = asyncHandler(async (req, res) => {
  await User.findById(req.params.id)
    .then((user) => {
      res.status(200).json(user);
    })
    .catch((error) => res.status(400).json({ error: "l'utilisateur n'existe pas" }));
});
```

Exemple d'une fonction de mon User controller qui permet de récupérer les informations d'un utilisateur avec son id.

Une fois que les routes, les controller et les middleware pour le router sont définis, j'ai pu monter le router sur mon application Express principale en utilisant la méthode `app.use()`

```
app.use('/api/users', require('./routes/usersRoute'));
app.use('/api/messages', require('./routes/messagesRoute'));
```

A partir de là on peut démarrer l'api en local qui écoute sur le port choisi en amont en utilisant la commande `npm start`.

```
// Code pour démarrer le serveur
http.listen(port, () => {
  console.log(`server started on port ${port}`);
});
```

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

## 2. Précisez les moyens utilisés :

Un ordinateur disposant d'une connexion internet

Git et github pour le versionning et le travail collaboratif

Trello pour l'organisation de travail

Un IDE

NPM

Express

Postman pour tester les appels à mon API

## 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf

Mohamad Sheiko

Chaïma Aicheche

Khellaf Rachedi

## 4. Contexte

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

Chantier, atelier, service ► Dans le cadre de ma formation concepteur/développeur d'application.

Période d'exercice ► Du : 02/01/2023 au : 31/01/2023

---

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

---

---

## 5. Informations complémentaires *(facultatif)*

---



## Activité-type 2

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

Exemple n° 1 ► Concevoir une base de données

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

L'an dernier, pour valider les compétences requises pour passer mon titre RNCP de développeur web/web mobile j'ai dû valider le projet le plus important de cette année-là.

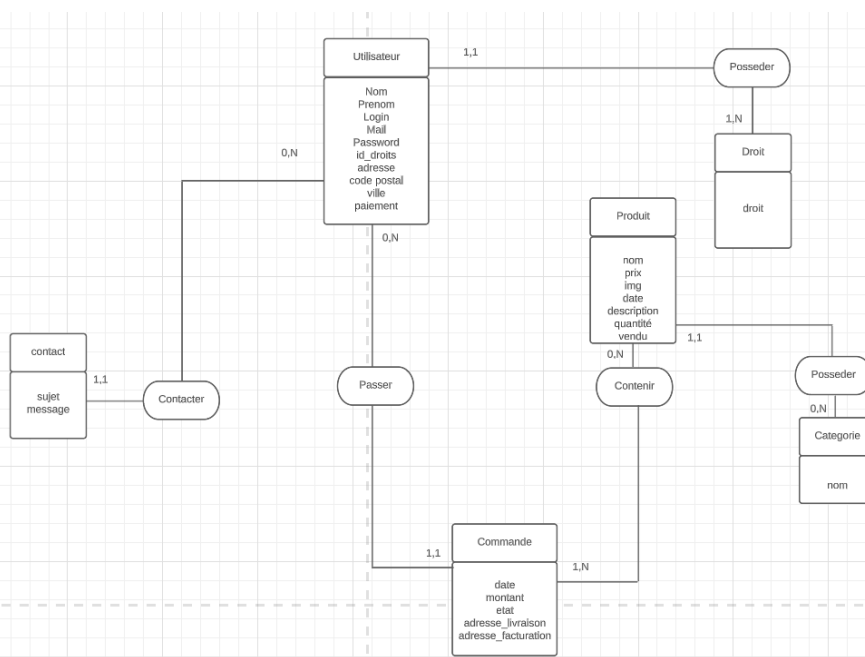
Pour réaliser ce projet je devais concevoir une base de données, pour faire cela j'ai utilisé la méthode MERISE (Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise).

Le fait de suivre cette méthode m'a permis de suivre un processus qui comprend les phases d'analyse des besoins, d'organisation des données en table, la spécificité des clés primaires et l'analyse des relations.

J'ai commencé par tracer le MCD (Modèle Conceptuel de Données), ce modèle simplifie la compréhension de la BDD, c'est un schéma Entité/Association. Les tables sont seulement au stade d'entités, entre les différentes entités il y a des liaisons que l'on nomme association, qui est un verbe d'action qui décrit l'opération qui se fait entre les deux entités.

Après avoir déterminé les besoins des utilisateurs de ma e-boutique. J'ai établi les règles de gestion des données à conserver. Par la suite j'ai déterminé toutes les entités nécessaires au bon fonctionnement de notre application, les relations entre entités, la modélisation des participations mini et maxi d'une entité à une relation c'est à dire les cardinalités, les propriétés qui correspondent aux informations descriptives d'une entité et la détermination unique d'une occurrence par un identifiant.

# DOSSIER PROFESSIONNEL (DP)



Voici le modèle conceptuel de données que j'avais réalisé pour la conception du projet boutique en ligne.

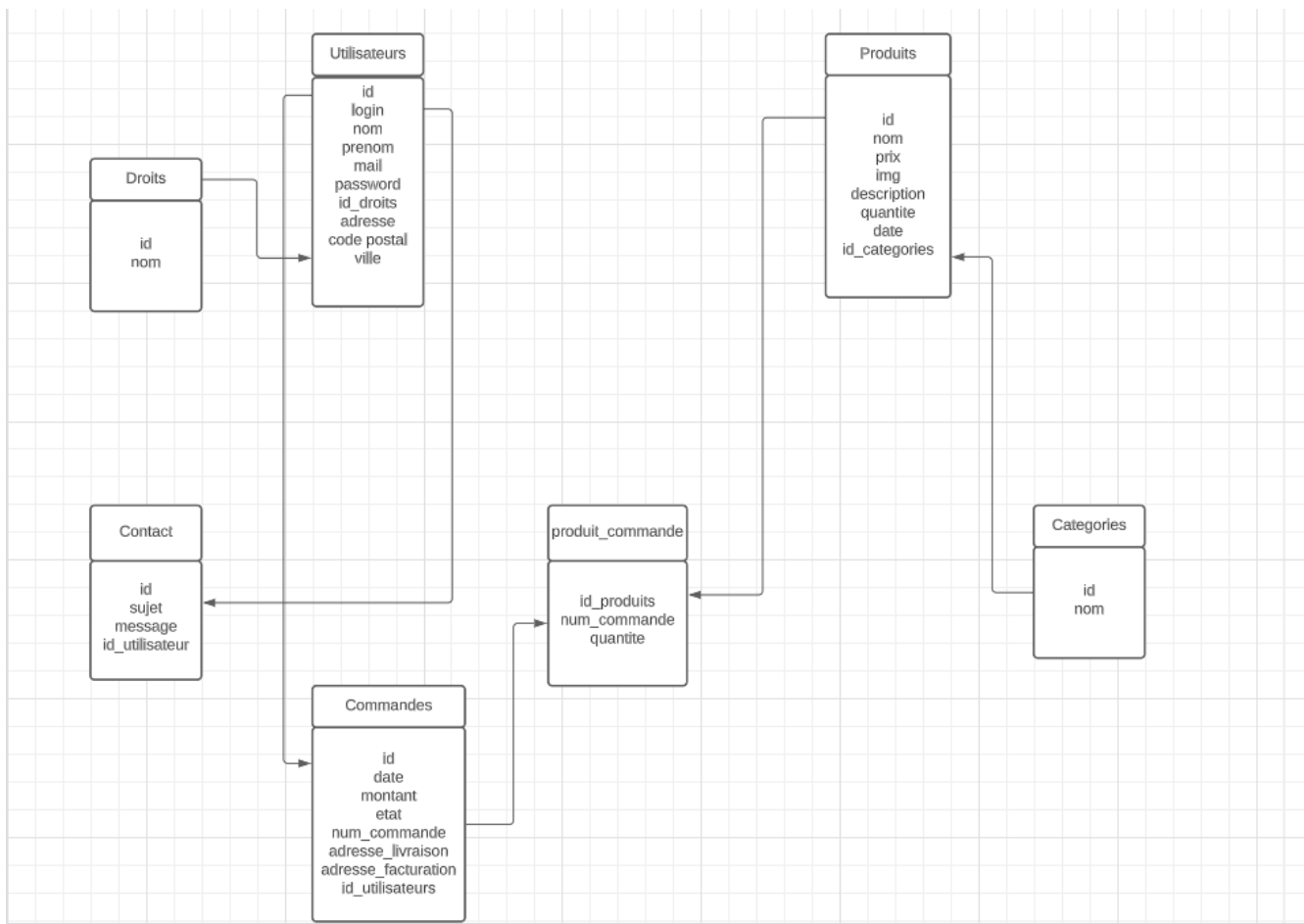
Le MLD (Modèle Logique de Données), est réalisé pour représenter les données manipulées par un système de gestion de base de données relationnel (SGBD).

Pour passer du MCD au MLD il y a des règles basiques à suivre :

Toutes les entités deviennent des tables, l'identifiant de l'entité devient une clé primaire de cette table, les propriétés de l'entité deviennent des attributs, les tables devant recevoir une clé étrangère les reçoivent, les cardinalités disparaissent ainsi que les verbes d'action.

Au niveau des relations, si prenons pour exemple la relation entre l'entité commande et l'entité produit leur relation est 1, N – 0, N donc au passage du MCD au MLD une table de liaison sera nécessaire qui accueillera les clés primaires de ces deux entités qui seront devenues des tables.

# DOSSIER PROFESSIONNEL (DP)



Modèle logique de données

## 2. Précisez les moyens utilisés :

Un ordinateur disposant d'une connexion internet

Lucid chart pour concevoir le MCD et le MLD

## 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf et Khellaf Rachedi

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

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_.*

Chantier, atelier, service ► *Dans le cadre de ma formation concepteur/développeur d'application.*

Période d'exercice ► Du : *02/04/2022* au : *10/04/2022*

## 5. Informations complémentaires (facultatif)

### Activité-type 3

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

**Exemple n° 1** ► Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

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

Au cours des nombreux projets en groupe j'ai eu à réaliser tels que l'e-boutique l'an dernier ou l'application mobile cette année, la collaboration a joué un rôle crucial dans la gestion de ces projets informatique et dans l'organisation de l'environnement de développement. Plusieurs outils ont facilité cette collaboration et ont permis à mes camarades et moi même de travailler de manière efficace et coordonnée. Parmi ces outils figure Trello, une plateforme de gestion de projet basée sur des tableaux kanban. Trello permet de créer des listes de tâches, d'assigner des responsabilités, de suivre l'avancement du projet et de partager des documents importants. Grâce à Trello, chaque membre de l'équipe peut voir clairement les tâches qui lui sont assignées, ainsi que l'état d'avancement général du projet.

Pour assurer une gestion efficace du code source, l'utilisation de Git est primordiale. Git est un système de contrôle de version qui permet de suivre les modifications apportées au code et de collaborer sur celui-ci. GitHub, quant à lui, est une plateforme d'hébergement de code source basée sur Git. Elle facilite le travail collaboratif en permettant à plusieurs développeurs de travailler simultanément sur un même projet. Chaque fonctionnalité développée peut être réalisée dans une branche distincte, ce qui évite les conflits et facilite la gestion des modifications.

L'utilisation de branches pour chaque fonctionnalité développée est une pratique courante dans les projets informatiques. Chaque membre de l'équipe crée sa propre branche lorsqu'il travaille sur une fonctionnalité spécifique. Cette approche permet de travailler de manière isolée, sans interférer avec le travail des autres développeurs. Une fois la fonctionnalité développée et testée localement, elle peut être fusionnée avec la branche principale

(généralement appelée "master" ou "main"). Cette fusion permet d'intégrer le travail de chaque membre dans une version consolidée du code.

Parallèlement aux branches spécifiques aux fonctionnalités, il est courant d'utiliser une branche "dev" (ou "développement") qui permet de regrouper le travail réalisé par tous les collaborateurs. Cette branche "dev" sert de point central pour la collaboration et permet de tester les différentes fonctionnalités intégrées ensemble. Elle offre également une visibilité globale sur l'état d'avancement du projet. Les développeurs peuvent régulièrement synchroniser leurs branches de fonctionnalités avec la branche "dev", ce qui facilite l'intégration continue et permet de détecter rapidement d'éventuels problèmes.

Nous avons également décidé de faire des réunions entre collaborateurs sur le projet assez souvent. Ces dernières sont très importantes, car elles permettent aux membres de l'équipe de se synchroniser, de partager des informations importantes et de prendre des décisions collectives. Ces réunions régulières favorisent la communication, renforcent la collaboration et garantissent que tous les membres de l'équipe sont sur la même longueur d'onde.

L'une des réunions les plus courantes est la réunion de suivi de projet, également appelée réunion de planification ou de revue. Cette réunion permet à l'équipe de discuter de l'état d'avancement du projet, de mettre à jour le planning, de revoir les objectifs et de prendre des décisions concernant les prochaines étapes. Elle offre l'occasion d'identifier les éventuels problèmes ou obstacles et de trouver des solutions en impliquant tous les membres de l'équipe.

Une autre réunion importante est la réunion de rétroaction, qui a lieu après la réalisation d'une itération, d'une phase ou d'un jalon spécifique du projet. L'objectif de cette réunion est d'évaluer les résultats obtenus, de discuter des leçons apprises et d'identifier les améliorations à apporter. Elle permet de prendre du recul, d'analyser les succès et les échecs, et de mettre en place des actions correctives pour optimiser les prochaines itérations ou phases du projet.

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

Les réunions dites de coordination sont également fréquentes. Elles sont utilisées pour synchroniser les efforts de l'équipe, discuter des tâches en cours, résoudre les problèmes, partager des mises à jour et aligner les priorités. Ces réunions peuvent être quotidiennes (appelées "stand-up meetings" ou "mêlées quotidiennes") pour une communication agile et rapide, ou avoir lieu à des intervalles réguliers, comme une fois par semaine, en fonction des besoins de l'équipe.

Pour conclure, tous les points que j'ai cité en amont ont permis d'avoir un travail d'équipe agréable, coordonné et efficace.

### 2. Précisez les moyens utilisés :

Un ordinateur avec une connexion internet.

Un IDE (Visual Studio Code).

Git pour le contrôle de version.

GitHub comme plateforme de contrôle de version de git afin de pouvoir les utiliser en collaboration.

Trello pour la gestion de projet.

### 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf et Khellaf Rachedi

### 4. Contexte

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

Chantier, atelier, service ► Dans le cadre de ma formation concepteur/développeur d'application.

Période d'exercice ► Du : 02/01/2023 au : 31/01/2023

---

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

---

---

## 5. Informations complémentaires *(facultatif)*

---



### Activité-type 3

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

Exemple n° 2 ► Concevoir une application mobile

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

Le projet le plus important de cette année était le développement d'une application mobile. J'ai fait le choix de développer une application de chat pour une entreprise fictive, offrant à ses employés une plateforme sécurisée et efficace pour communiquer. Grâce à l'utilisation de la technologie React Native pour la partie front-end, j'ai pu créer une interface utilisateur intuitive et réactive, adaptée à différentes plateformes mobiles. Pour assurer une expérience fluide, j'ai conçu une API performante en utilisant Node.js avec Express, offrant une gestion rapide des requêtes et une scalabilité sans faille. La conception de l'API a été axée sur la robustesse, garantissant une disponibilité maximale et une réactivité optimale en cas de montées en charge. Une attention particulière a été portée à la sécurité des données, en implémentant des mesures de protection avancées pour prévenir toute atteinte à la confidentialité ou à l'intégrité des informations de l'entreprise. L'application se positionne ainsi comme une solution de communication incontournable pour une collaboration efficace et confidentielle au sein de l'entreprise fictive.

L'échange de message repose sur l'utilisation de websockets pour des échanges instantanés rapides et réactifs. Les employés peuvent ainsi communiquer en temps réel de manière fluide et efficace. Pour garantir un niveau de sécurité élevé, j'ai mis en place un système d'authentification basé sur des tokens. Chaque utilisateur doit s'authentifier avec un token valide pour accéder à l'application et communiquer avec les autres membres de l'entreprise.

En ce qui concerne la sécurité de mon API, j'ai pris plusieurs mesures pour protéger les données de l'entreprise. J'ai utilisé des expressions régulières (regex) et le module "express-validator" pour valider et filtrer toutes les données reçues par l'API. Cela permet de s'assurer que seules les données appropriées et correctement formatées sont traitées, réduisant ainsi les risques d'injection de code malveillant ou de données incorrectes.

Pour renforcer davantage la sécurité de mon application et gérer les problèmes de sécurité liés

aux requêtes http j'ai décidé d'utiliser HELMET et CORS.

Le premier est une bibliothèque middleware qui aide à sécuriser les applications Express en configurant de manière appropriée les en-têtes HTTP. Elle permet de mettre en place diverses mesures de sécurité telles que la protection contre les attaques de type Cross-Site Scripting (XSS), la désactivation de la mise en cache côté client, la protection contre les attaques de type Clickjacking, et bien d'autres encore. En utilisant Helmet, j'ai pu facilement ajouter une couche de sécurité supplémentaire à mon application en exploitant les fonctionnalités de sécurité intégrées aux en-têtes http.

Cors, quant à lui, est un middleware qui permet de gérer les problèmes de politique de partage des ressources entre différents domaines (Cross-Origin Resource Sharing). Il permet de définir des règles spécifiques sur les ressources que votre serveur est autorisé à partager avec d'autres domaines. Cors m'a permis de contrôler finement les autorisations d'accès aux ressources de mon API en fonction de l'origine de la requête, ce qui aide à prévenir les attaques de type Cross-Site Request Forgery (CSRF) et garantit une meilleure sécurité dans les interactions entre les clients et le serveur.

User Story :

-En tant qu'utilisateur,

Je souhaite pouvoir m'inscrire et me connecter à l'application

Afin d'accéder à toutes les fonctionnalités et interagir avec d'autres utilisateurs.

Critères d'acceptation :

L'utilisateur doit pouvoir créer un nouveau compte en fournissant son nom, son adresse e-mail et un mot de passe sécurisé.

Lors de l'inscription, l'application doit vérifier si l'adresse e-mail est unique et si le mot de passe respecte les critères de sécurité.

---

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

---

Une fois inscrit, l'utilisateur doit pouvoir se connecter en utilisant son adresse e-mail et son mot de passe.

En cas d'échec de la connexion, l'application doit afficher un message d'erreur approprié pour informer l'utilisateur.

Lorsque l'utilisateur est connecté, il doit être redirigé vers l'écran principal de l'application.

-En tant qu'utilisateur connecté,

Je souhaite pouvoir envoyer des messages à d'autres utilisateurs

Afin de communiquer et d'établir des conversations.

Critères d'acceptation :

L'utilisateur doit pouvoir rechercher d'autres utilisateurs en utilisant leur nom ou leur adresse e-mail.

L'utilisateur doit pouvoir ouvrir une conversation existante pour voir les messages précédents et envoyer de nouveaux messages.

-En tant qu'utilisateur connecté,

Je souhaite pouvoir modifier mon profil

Afin de mettre à jour mes informations personnelles.

Critères d'acceptation :

L'utilisateur doit pouvoir accéder à son profil depuis le menu de l'application.

---

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

---

L'application doit afficher les informations actuelles de l'utilisateur, telles que son nom et son adresse e-mail.

L'utilisateur doit pouvoir modifier ses informations personnelles, telles que son nom et son adresse e-mail.

L'application doit valider les modifications et les enregistrer correctement.

L'utilisateur doit pouvoir modifier son mot de passe en fournissant l'ancien mot de passe et en choisissant un nouveau mot de passe sécurisé.

-En tant qu'utilisateur connecté,

Je souhaite pouvoir voir le profil d'autres utilisateurs

Afin de connaître leurs informations personnelles.

Critères d'acceptation :

L'utilisateur doit pouvoir rechercher d'autres utilisateurs en utilisant leur nom ou leur adresse e-mail.

Une fois trouvé, l'utilisateur doit pouvoir afficher le profil d'un autre utilisateur.

L'application doit afficher les informations personnelles de l'utilisateur, telles que son nom et son adresse e-mail.

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

L'application développée par mes camarades et moi adopte une architecture en trois couches.

1 - Une couche interface utilisateur

2- Une couche métier

3 - Une couche data



## 2. Précisez les moyens utilisés :

Un ordinateur avec une connexion internet. Pour la création de la maquette nous avons utilisé figma, il permet de collaborer en ligne. Nous avons utilisé Trello pour la gestion du projet. Git et github pour ce qui est versionning du code.

## 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf, Khellaf Rachedi, Chaima Aicheche et Mohamad Sheikho.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_.*

Chantier, atelier, service ► *Dans le cadre de ma formation concepteur/développeur d'application.*

Période d'exercice ► Du : *02/01/2023* au : *31/01/2023*

---

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

---

---

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

---

### Activité-type 3

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

*Exemple n° 6* ► Préparer et exécuter les plans de tests d'une application

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

Durant mon alternance au sein de l'équipe de développement chez Richardson, j'ai eu l'opportunité de travailler sur la mise en place d'une couverture de test end-to-end automatisée pour l'application développée par l'équipe. L'objectif principal était de garantir la qualité du logiciel en identifiant rapidement les problèmes potentiels lors des déploiements.

#### Description du projet :

J'ai utilisé le framework Playwright en conjonction avec TypeScript pour mettre en œuvre cette couverture de test automatisée. Playwright est un outil puissant qui permet d'automatiser les tests d'applications web sur différents navigateurs tels que Chrome, Firefox et Safari. TypeScript, quant à lui, offre une typisation solide et facilite le développement en fournissant des fonctionnalités avancées pour la création de scripts de test.

Le processus de mise en place de la couverture de test a été divisé en plusieurs étapes :

Identification des scénarios de test : J'ai travaillé en étroite collaboration avec les membres de l'équipe pour comprendre les fonctionnalités clés de l'application et déterminer les scénarios de test les plus pertinents à automatiser.

Développement des scripts de test : En utilisant Playwright et TypeScript, j'ai écrit des scripts de test pour chaque scénario identifié. Ces scripts ont été conçus pour simuler les actions d'un

utilisateur réel sur l'application, telles que la saisie de données, la navigation entre les pages et l'interaction avec les éléments de l'interface utilisateur.

Intégration dans la CI/CD : J'ai configuré les scripts de test pour être exécutés automatiquement après chaque pull request du front de l'application. Cela a permis d'assurer une validation continue de la fonctionnalité de l'application à chaque modification de code.

Résultats obtenus :

La mise en place de la couverture de test end-to-end automatisée a eu un impact significatif sur la qualité du logiciel développé par l'équipe. Les résultats obtenus ont été les suivants :

Détection précoce des problèmes : Les scripts de test ont permis de détecter rapidement les régressions et les anomalies fonctionnelles, ce qui a réduit le temps de débogage et d'investigation.

Amélioration de la fiabilité : Les tests automatisés ont augmenté la fiabilité de l'application en vérifiant régulièrement son bon fonctionnement, même lors de modifications fréquentes du code.

Réduction des erreurs humaines : L'automatisation des tests a réduit les erreurs humaines lors des validations manuelles, ce qui a permis d'économiser du temps et des ressources.

Conclusion :

En mettant en œuvre une couverture de test end-to-end automatisée à l'aide de Playwright et TypeScript, j'ai contribué à l'amélioration de la qualité et de la fiabilité de l'application développée par l'équipe du projet de refonte du système d'information. Cette solution a permis une détection précoce des problèmes et une réduction des erreurs humaines. De plus, en



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

intégrant les tests automatisés dans le processus de CI/CD, nous avons garanti une validation continue de la fonctionnalité de l'application après chaque modification de code.

Ce projet m'a permis de renforcer mes compétences en développement TypeScript, en automatisation des tests et en collaboration au sein d'une équipe de développement.

### 2. Précisez les moyens utilisés :

Un ordinateur avec une connexion internet. Un IDE (visual studio code). Playwright. Typescript. NPM.

### 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul pour le développement des tests et en collaboration avec mon tuteur et le dev ops de l'entreprise pour mettre en place les tests dans la CI/CD.

### 4. Contexte

Nom de l'entreprise, organisme ou association ► *Richardson*

Chantier, atelier, service ► *Dans le cadre de mon alternance en entreprise*

Période d'exercice ► Du : *24/10/2022* au : *1/02/2023*

### 5. Informations complémentaires (facultatif)

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


*Exemple n° 7* ► Préparer et exécuter le déploiement d'une application

---


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

Après avoir bien avancé le développement de l'API, j'ai décidé d'héberger celle-ci afin qu'elle soit disponible et utilisable par tous, cela permet également d'optimiser le processus de développement, avec mes collaborateurs nous avons mis en place une intégration continue fluide en utilisant Plesk et GitHub. Grâce à cette intégration, chaque modification apportée à la branche principale de notre API sur GitHub déclenche automatiquement un déploiement sur notre serveur Plesk. Une fois que les modifications sont poussées sur la branche principale, le système détecte automatiquement ces changements et lance le processus de déploiement.


## DOSSIER PROFESSIONNEL (DP)

 **api-rcmk.git**

URL




Branch




Deploy key

Latest commits


- 2023-03-09 20:45 Merge pull request #13 from mohamed-zidi/dev Dev
- 2023-03-09 20:44 Merge pull request #12 from mohamed-zidi/features/moha...




 [show more](#)

 Pull now

Deployment

main branch automatically to [/httpdocs](#)

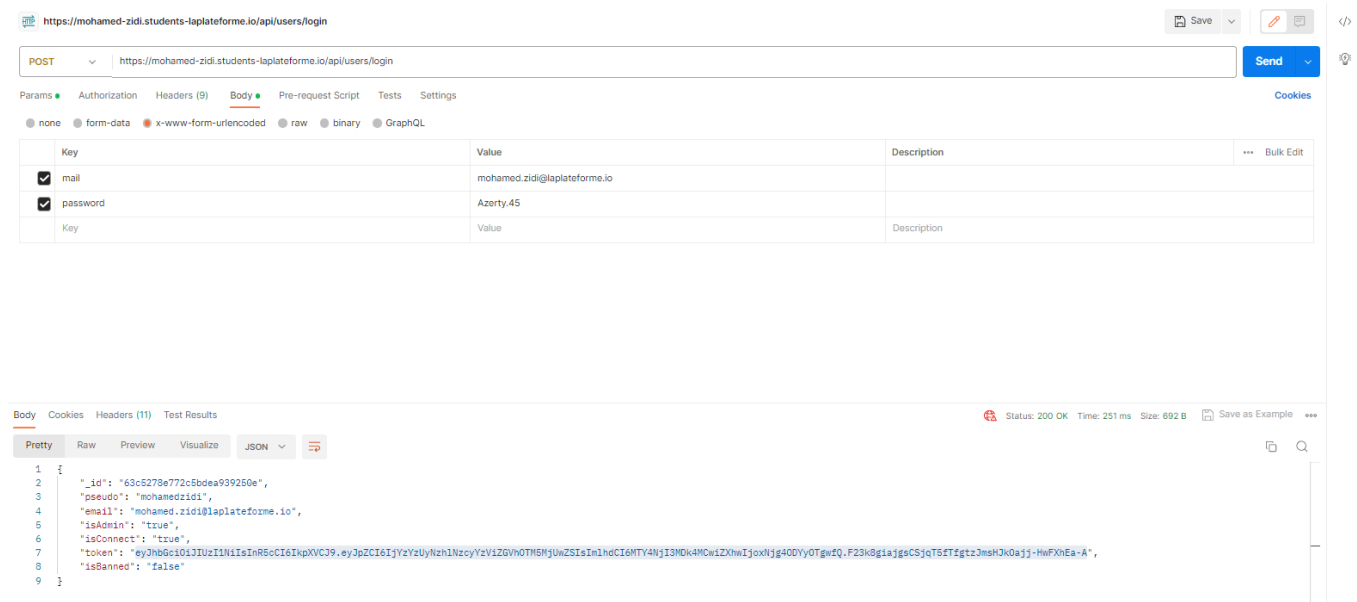
 Deploy now

Plesk, notre plateforme d'hébergement, joue un rôle essentiel dans ce processus. Elle offre une interface conviviale qui nous permet de configurer facilement les paramètres de déploiement, tels que la synchronisation avec notre référentiel Git, le choix de la branche à déployer et la gestion des dépendances. Une fois que le processus de déploiement est enclenché, Plesk récupère automatiquement les dernières modifications depuis GitHub et les déploie sur notre serveur en utilisant les protocoles appropriés. Cette intégration transparente entre Plesk et GitHub nous permet de maintenir une version toujours à jour de notre API sur notre environnement de production, sans nécessiter d'interventions manuelles fastidieuses.

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

En synchronisant notre flux de travail de développement avec notre gestionnaire de versions Git et notre plateforme d'hébergement Plesk, nous bénéficions d'une gestion efficace de notre code et d'une diffusion continue de notre API aux utilisateurs finaux. Cela nous permet de fournir rapidement de nouvelles fonctionnalités, de corriger les bugs et d'améliorer la performance de notre API, tout en maintenant une infrastructure de déploiement stable et sécurisée. Grâce à cette intégration continue, nous avons pu rationaliser notre processus de développement et offrir une expérience fluide à nos utilisateurs.



Ci-dessus l'appel de la route login de mon API avec l'URL fourni par plesk depuis postman prouvant que mon api est bien hébergée est fonctionnelle car elle renvoie une réponse 200 et le bon jeu de données.

## 2. Précisez les moyens utilisés :

Un ordinateur avec une connexion internet.

Plesk.

git.

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

## 3. Avec qui avez-vous travaillé ?

Roehan Abdul Rauf, Khellaf Rachedi, Chaima Aicheche et Mohamad Sheikho.

## 4. Contexte

Nom de l'entreprise, organisme ou association ► *LaPlateforme\_.*

Chantier, atelier, service ► *Dans le cadre de ma formation concepteur/développeur d'application.*

Période d'exercice ► Du : *02/01/2023* au : *31/01/2023*

## 5. Informations complémentaires (facultatif)

---

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

---

### 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) [prénom et nom] **ZIDI Mohamed** ,  
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 **09/06/2022**

pour faire valoir ce que de droit.

Signature : ZIDI Mohamed

---

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

---

### Documents illustrant la pratique professionnelle

*(facultatif)*

| Intitulé                         |
|----------------------------------|
| Cliquez ici pour taper du texte. |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |
|                                  |



---

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

---

### ANNEXES

*(Si le RC le prévoit)*