



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ➤ SEBAA
Nom d'usage ➤ SEBAA
Prénom ➤ BADR
Adresse ➤ 258 avenue de Toulon

Titre professionnel visé

Concepteur(trice) Développeur(se) Informatique

MODALITÉ D'ACCÈS :

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

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

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



Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.	5
- Projet "ARCO" - Maquetter une application	p. p.	5
- Projet "ARCO"	p. p.	9
- Intitulé de l'exemple n° 3	p. p.	
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	p.	
- Projet "ARCO"	p. p.	14
- Projet "ARCO"	p. p.	17
- Intitulé de l'exemple n° 3	p. p.	
Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	p.	
- Projet "ARCO"	p. p.	20
- Intitulé de l'exemple n° 2	p. p.	
- Intitulé de l'exemple n° 3	p. p.	
Titres, diplômes, CQP, attestations de formation (facultatif)	p.	24
Déclaration sur l'honneur	p.	25
Documents illustrant la pratique professionnelle (facultatif)	p.	26
Annexes (Si le RC le prévoit)	p.	27

EXEMPLES DE PRATIQUE PROFESSIONNELLE

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

Exemple n°1 - Projet "ARCO" - Maquetter une application

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

Dans le cadre de ma formation j'ai conçu une application web.

Elle a pour but de permettre la communication entre une ou plusieurs personnes, à l'aide de chat textuel.

Après avoir établi les besoins des utilisateurs, j'ai défini un ensemble de scénarios en décrivant en détail ce que les utilisateurs pourraient faire sur l'application et pourquoi ils le font, ce qui inclut tous les renseignements pertinents pour le processus suivi par l'utilisateur en vue d'atteindre son objectif, qui m'ont servi de base pour mettre en place la navigation entre les interfaces :

- Page de connexion/inscription : permet de s'inscrire et de se connecter si l'on dispose déjà d'un compte.
- Page

Ce projet valide les compétences :

- **Maquetter une application**
- **Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement**

2. Précisez les moyens utilisés :

DOSSIER PROFESSIONNEL ^(DP)

Pour réaliser la maquette :

J'ai d'abord réalisé un wireframe sur papier pour mieux visualiser l'ensemble de mes pages et donné un premier coup d'organisation dans celle-ci.

j'ai ensuite élaboré une légère maquette et établi une charte graphique à l'aide de Figma pour avoir un début de design sur mes pages .

j'ai fini par concevoir le prototype final de l'application à l'aide de figma .

je me suis naturellement tourné vers TRELLO pour l'organisation des tâches .

3. Avec qui avez-vous travaillé ?

Pour ce projet j'ai travaillé avec Lucas Verdier , Kilian Guerin et Bassem Sataf

4. Contexte

Nom de l'entreprise, organisme ou association ▶ LaPlateforme_

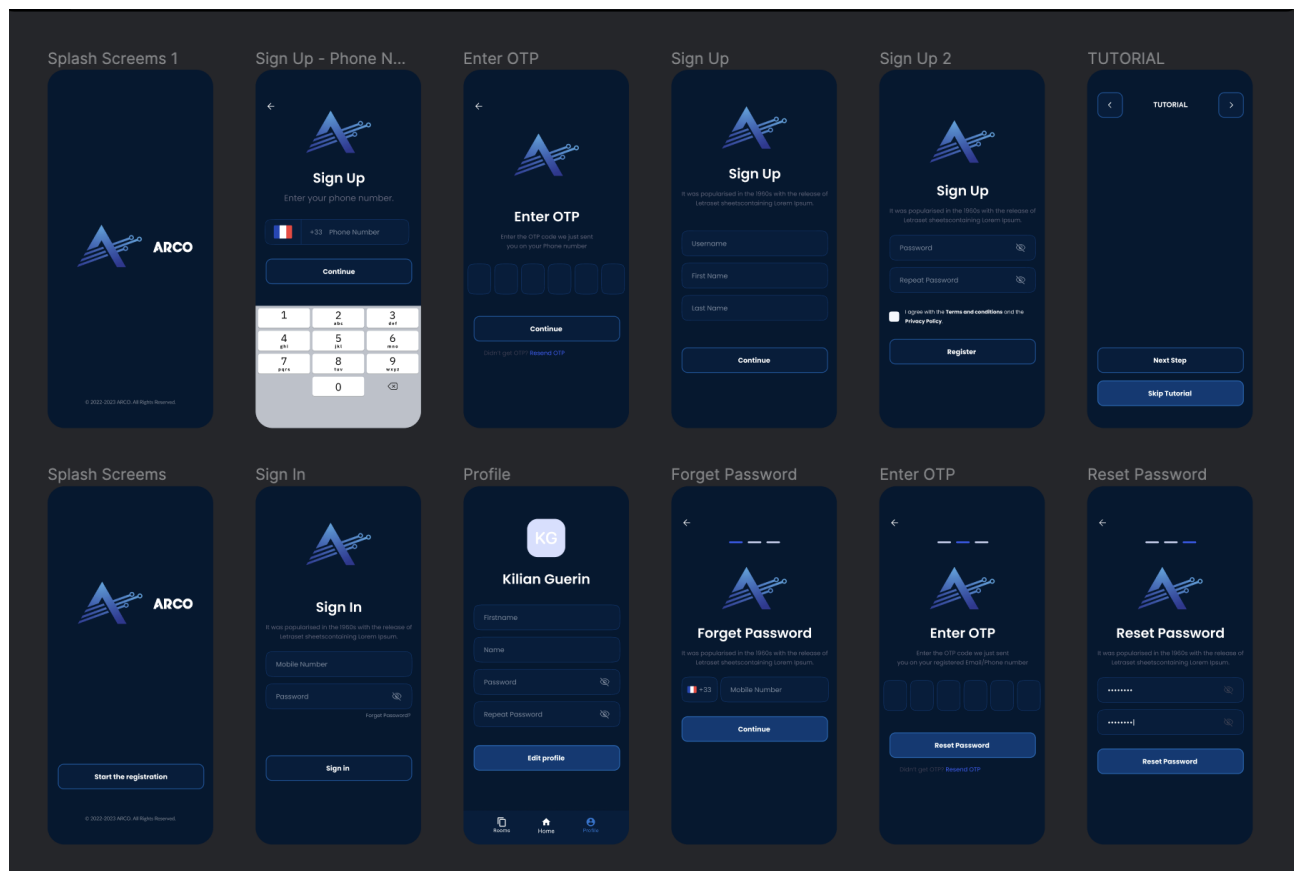
Chantier, atelier, service ▶ ARKO

Période d'exercice ▶ Du : 02/01 au : 13/01

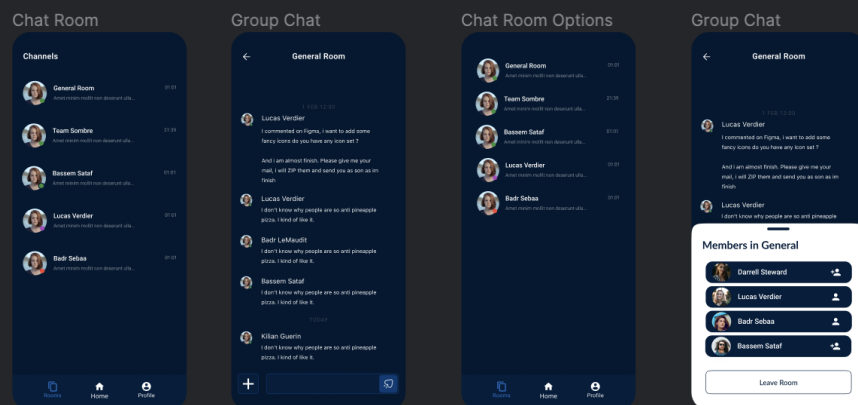
5. Informations complémentaires (facultatif)



DOSSIER PROFESSIONNEL (DP)



DOSSIER PROFESSIONNEL ^(DP)



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

Exemple n° 2 - Projet "ARCO"

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

Dans le cadre du projet ARCO, j'ai développé un panel Admin web en **REACT.JS**

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 utilisateurs, les ChatRoom et les messages envoyés.

J'ai choisi **React js** car étant la librairie javascript la plus connue de nos jours, elle dispose d'une documentation et de librairie assez large. Le format **JSX** permettant d'allier le langage de balisage qu'est l'HTML avec toute la puissance de javascript.

Pour le back j'ai opté pour **Express**, un framework de **node js** qui facilite la mise en place de l'API en nous permettant d'utiliser des routes.

```
▼ routes
  JS chatRoom.js
  JS message.js
  JS user.js
```

Pour la gestion de base de données, nous avons opté pour du **NOSQL** avec

Mongoose, le NOSQL nous apporte une flexibilité et une facilité à élargir la base de données dans le temps.

Nous créons donc des schéma :

```
const mongoose = require("mongoose");
const uniqueValidator = require("mongoose-unique-validator");

const userSchema = new mongoose.Schema({
  telephone: { type: String, required: true, unique: true },
  firstname: { type: String, required: true },
  lastname: { type: String, required: true },
  password: { type: String, required: true },
  username: { type: String, required: true, unique: true },
  role: {
    type: String,
    required: true,
    enum: ["admin", "user"],
    default: "user",
  },
});

userSchema.plugin(uniqueValidator);

module.exports = mongoose.model("user", userSchema);
```

Que nous appelons côté back dans une fonction du contrôleur qui correspond :

```
const getAllUsers = (req, res) => {
  User.find()
    .then(users => {
      res.status(200).json({ status: "ok", message: "Liste des utilisateurs", data: {users} });
    })
    .catch(error => res.status(500).json({ error }));
};
```

Pour récupérer ces données en Front nous faisons :

```
useEffect(() => {
  fetch("http://localhost:8800/api/user/users")
    .then((res) => res.json())
    .then((res) => {
      setUsers(res.data.users);
    });
}, [users]);
```

On récupère les données dans un **useEffect** avec la fonction **fetch**.

Nous avons utilisé la librairie **MUI** de **REACT.JS** qui nous a facilité la création de composants **UI** et nous donne un rendu propre et esthétique .

Ce système nous permet d'exécuter plusieurs fonctions sur la base de données en connectant celle-ci avec le Back-end et le Front-end de notre application.

Le panel admin se présente comme tel avec différents onglets :

- les utilisateurs (créer / modifier / supprimer leur compte)
- les chatroom (modifier / supprimer)
- les messages (supprimer les messages)

2. Précisez les moyens utilisés :

J'ai utilisé :

- VScode
- React js
- Node js
- Mongoose

DOSSIER PROFESSIONNEL ^(DP)

- les librairies react utilisées : MUI, Jwt-decode

3. Avec qui avez-vous travaillé ?

Pour ce projet j'ai travailler avec **Lucas Verdier** , **Kilian Guerin** et **Bassem Sataf**

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **LaPlateforme_**

Chantier, atelier, service ▶ **Arko**

Période d'exercice ▶ Du : **16/01** au : **27/01**

5. Informations complémentaires *(facultatif)*

The screenshot shows a web application interface for user management. The sidebar on the left contains the following links:

- ADMINIS
- Manage Users
- Manage ChatRooms
- Manage Messages
- Profile Form

The main content area is titled "Users" and "Managing the Users". It displays a table with the following data:

ID	lastname	firstname	telephone	username
63b453...	Verdier	Lucas	0622192000	lucue
63b458...	Zerin	Zilian	0612345654	
63b594...	Zaldersoon	Bradradmin	06123454306	leZaudit
6460d4...	admin	admin	+33676565491	admin

The user "Zadmin" is highlighted as the current user. The bottom of the page shows "Rows per page: 100" and "1-4 of 4".

DOSSIER PROFESSIONNEL (DP)



<input type="checkbox"/>	ID	lastname	firstna
<input type="checkbox"/>	63b453...	Verdier	Lucas
<input type="checkbox"/>	63b458...	Zerin	Zillian
<input type="checkbox"/>	63b594...	Zaldersoon	Bradra
<input type="checkbox"/>	6460d4...	admin	admin

User Update

First Name
Lucas

Last Name
Verdier

telephone
0622192000

Username
lucluc

UPDATE USER

username	
lucluc	<input type="checkbox"/> <input type="checkbox"/>
	<input type="checkbox"/> <input type="checkbox"/>
leZaudit	<input type="checkbox"/> <input type="checkbox"/>
admin	<input type="checkbox"/> <input type="checkbox"/>

Rows per page: 100 1-4 of 4 < >

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

Exemple n° 1 - *Projet "ARCO"*

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

Dans le cadre du développement de notre application mobile, nous avons développé une API en NodeJs pour interagir avec notre base de données (MongoDB) et envoyer/recevoir ces données côté client (front React Native). Pour faciliter le travail en équipe dès l'API terminée, je l'ai déployé sur Plesk. L'API comprend :

- Des routes permettant l'accès aux données (GET, POST, PUT, DELETE).
- Un middleware permettant de gérer l'authentification grâce à un Token d'identification utilisateur et un refresh Token qui s'update tous les 'N temps' afin de re-vérifier l'intégrité de l'utilisateur.

- API déployé sur un hébergeur (Plesk). Exemple d'URL :

"https://lucas-verdier.students-laplateforme.io/api/message/room/:id" (cette route nécessite le token d'identification + le refresh token).

Websites & Domains >

Node.js on lucas-verdier.students-laplateforme.io

[Restart App](#) [NPM install](#) [Run script](#) [Disable Node.js](#)

Node.js Version [16.20.0](#)

Package Manager [npm](#)  This is what we detected, you can change it

Document Root [/httpdocs/app-mobile-chat](#)

Application Mode [development](#)

Application URL <http://lucas-verdier.students-laplateforme.io>

Application Root [/httpdocs/app-mobile-chat](#) [\[open\]](#)

Application Startup File [server.js](#) [\[edit\]](#)

Custom environment
variables

```
- URL: mongodb+srv://Admin: [redacted]@app-mobile-chat.nkx3vhk.mongodb.net/ARCO?retryWrites=true&w=majority
- ACCESS_TOKEN_SECRET: [redacted]
- REFRESH_TOKEN_SECRET: [redacted]
- ALGO_MSG: [redacted]
- KEY_MSG: [redacted]
\[specify\]
```

Variables
d'environnement



```
router.post('/send', auth, sendMessage);
router.delete('/delete', auth, deleteMessage);
router.get('/all', auth, getAllRoomMessages);
router.get('/room/:id', auth, getMessagesByRoomId);
router.get('/decrypt/:id', auth, decryptMessage);
```

Fonctionnement de l'API :

— L'API fait le lien de manière sécurisée entre la base de données et le côté client.

MongoDB <—> API NodeJS <—> Front React Native

DOSSIER PROFESSIONNEL ^(DP)

Compétences REAC validées par ce projet :

- Construire une application organisée en couches.
- Préparer et exécuter le déploiement d'une application.
- Développer des composants d'accès aux données.
- Développer des composants dans le langage d'une base de données.

2. Précisez les moyens utilisés :

J'ai utilisé :

- IDE: VS Code
- NodeJS - Plesk
- MongoDB

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec Lucas Verdier , Kilian Guerin et Bassem Sataf

4. Contexte

Nom de l'entreprise, organisme ou association ▶ LaPlateforme_

Chantier, atelier, service ▶ ARCO

Période d'exercice ▶ Du : 02/01/23 au : 27/01/23

5. Informations complémentaires (facultatif)

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

Exemple n° 1 - *Projet "ARCO"*

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

Dans le cadre d'un projet scolaire, nous avons créé une application mobile en React Native, qui interagit avec une API nodeJS qui s'appuie sur une base de données NoSQL MongoDB. C'est un projet qui avait une durée de 4 semaines pour sortir une V1.

Nous avons travaillé de façon "Agile", c'est-à-dire avancer petit à petit, communiquer sur notre avancée, faire des points et réajuster si besoin.

Pour ce faire nous avons établis les rituels suivants :

- Réunion daily : tous les matins chacun explique ce qu'il a fait la veille et ce qu'il compte faire aujourd'hui.
- Revue de code en fin de sprint (chaque vendredi matin) : le but est que chacun montre ce qu'il a fait pendant le sprint, et tout le monde peut intervenir et donner son avis, des conseils etc...
- Rétrospective en fin de sprint également (vendredi après-midi) :

C'est le moment de se réunir avec l'équipe et d'évoquer tous les points qui ont bien marché pour continuer sur la même dynamique.

Mais aussi surtout, évoquer ce qui s'est mal passé, n'a pas marché, afin de trouver des solutions et ne plus être gêné sur le sprint suivant.

Ainsi nous avons donc établis 4 sprints de 1 semaine avec les objectifs suivants :

- Sprint 1 → Elaboration du cahier des charges, conception et création de la base de données, Maquettage Figma.

DOSSIER PROFESSIONNEL ^(DP)

- Sprint 2 → Développement de l'API NodeJS, hébergement de l'API.
- Sprint 3 → Développement de l'application mobile React Native.
- Sprint 4 → Développement de l'application mobile React Native, développement du dashboard admin React JS.

Compétences REAC couvertes par ce projet :

- **Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.**
- **Concevoir une application.**
- **Développer les composants métiers.**
- **Construire une application organisée en couches. - Développer une application mobile.**
- **Préparer et exécuter le déploiement d'une application.**

2. Précisez les moyens utilisés :

Nous avons utilisés :

- VS CODE.
- React Native.
- API nodeJS.
- Expo Go.

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec **Lucas Verdier** , **Kilian Guerin** et **Bassem Sataf**

4. Contexte

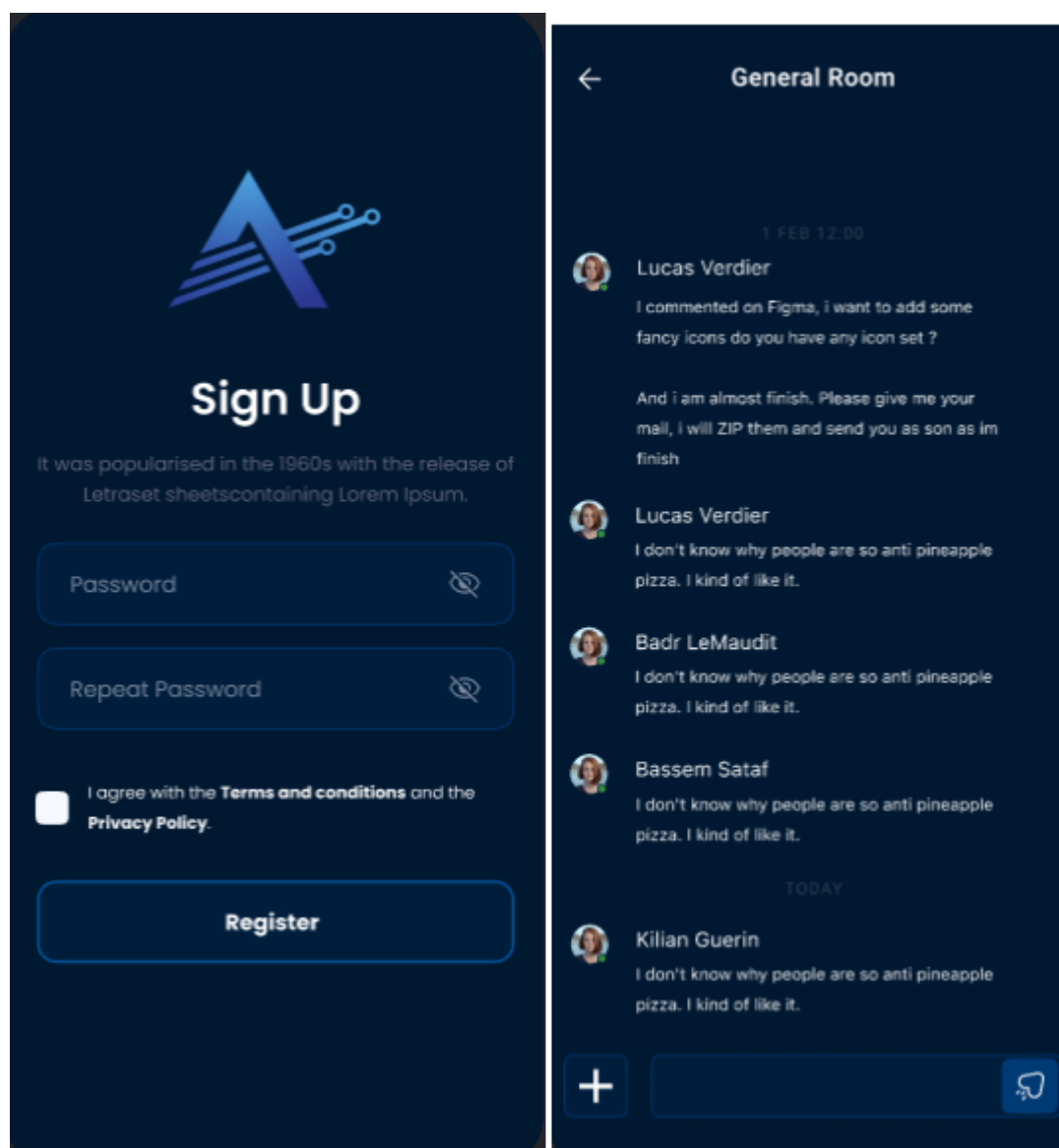
DOSSIER PROFESSIONNEL ^(DP)

Nom de l'entreprise, organisme ou association ▶ LaPlateforme_

Chantier, atelier, service ▶ ARCO

Période d'exercice ▶ Du : 02/01/23 au : 27/01/23

5. Informations complémentaires (facultatif)



Activité-type 3 Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 1 - Projet "ARCO"

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

Dans le cadre de notre projet de réalisation d'une application mobile, nous avons dû concevoir une base de données afin d'envoyer et recevoir des données vers/depuis l'API. Ayant à développer une application mobile qui a pour fonction métier un chat général simple en temps réel nous avons opté pour une base de données **No SQL** (Not Only Structured Query Language), en l'occurrence, MongoDB. Avantages d'une base de données No SQL pour une application mobile de chat :

- Modèle de données flexibles : les données peuvent être facilement ajoutées et/ou supprimées sans impacter la structure de la base de données.
- Facilement scalable: permet de gérer un très grand flux de données.
- Disponibilité des données. Le No SQL permet d'avoir de très bonnes performances sur la vitesse d'accès aux données.
- Il est aussi possible de travailler avec des données structurées (Not ONLY sql). Cependant, si l'appli avait été plus complexe et nécessitait des requêtes plus complexes, avec beaucoup de jointures par exemple, nous aurions opté pour une base de données SQL. Exemples de requêtes vers la base de données :

```
const createChatRoom = async (req, res) => {
  const { name, usersInRoom } = req.body;
  try {
    if (checkRole(req.auth.role)) {
      const chatRoom = new ChatRoom({ name, usersInRoom });
      await chatRoom.save();
      res.status(201).json({ chatRoom });
    }
  } catch (err) {
    res.status(401).json({ err });
  }
};
```

```
const getMessagesByRoomId = async (req, res) => {
  try {
    if (!req.params.hasOwnProperty('id')) {
      return res.status(400).json({ message: 'Missing room ID' });
    }

    const messages = await Message.find({chatroom_id: req.params.id});
    let decryptedObj = [];
    for (let i = 0; i < messages.length; i++) {
      const decryptedText = await decryptMessage(messages[i]._id);
      decryptedObj.push([
        decryptedText,
        messages[i].user_id,
        messages[i].createdAt,
      ]);
    }
    res
      .status(200)
      .json({
        status: "ok",
        message: `Messages de la room ${req.params.id}`,
        data: decryptedObj,
      });
  } catch (error) {
    res.status(500).json({status: "Erreur.", message: error.message });
  }
}
```

Compétences REAC validées par le projet :

- Concevoir une base de données.
- Mettre en place une base de données.

DOSSIER PROFESSIONNEL ^(DP)

- Développer des composants dans le langage d'une base de données.

2. Précisez les moyens utilisés :

J'ai utilisé :

- IDE: VS CODE
- Node.js
- Plesk
- MongoDB

3. Avec qui avez-vous travaillé ?

Pour ce projet, j'ai travaillé avec **Lucas Verdier** , **Kilian Guerin** et **Bassem Sataf**

4. Contexte

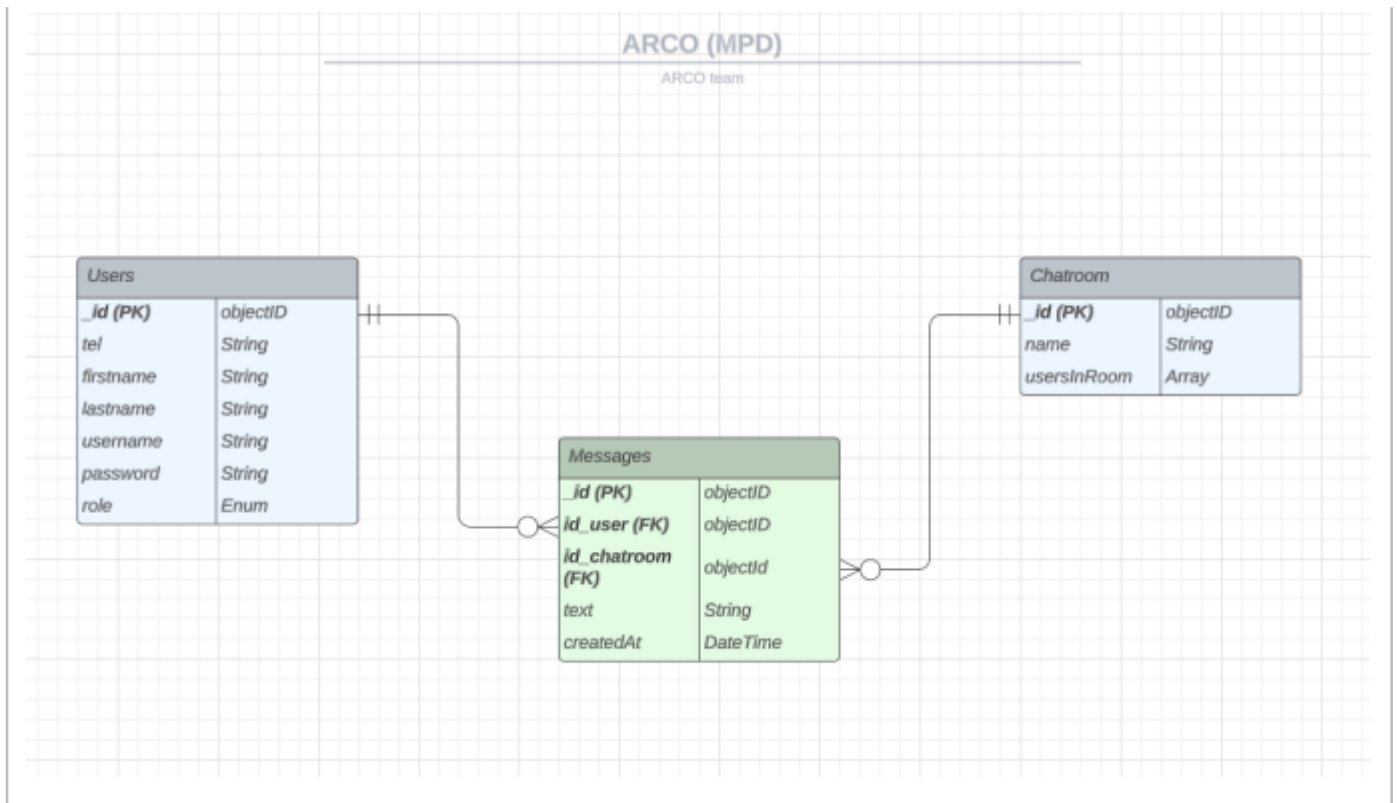
Nom de l'entreprise, organisme ou association ▶ **LaPlateforme_**

Chantier, atelier, service ▶ **ARCO**

Période d'exercice ▶ Du : **Cliquez ici** au : **Cliquez ici**

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)



DOSSIER PROFESSIONNEL ^(DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Concepteur et développeur d'application	La plateforme	Le 17/07/2023

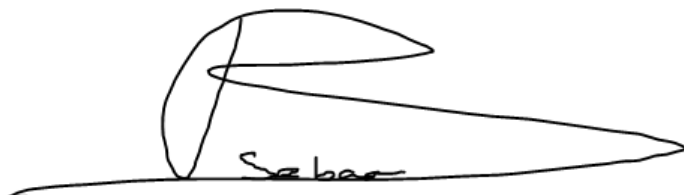
Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] **SEBAA BADR** ,
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 **17/07/23**

pour faire valoir ce que de droit.

Signature :

A handwritten signature in black ink, appearing to read 'Sebaa', with a large, stylized flourish above it.

DOSSIER PROFESSIONNEL ^(DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)