



DOSSIER PROFESSIONNEL (DP)

Nom de naissance ➤ BOUSSIOUF
Nom d'usage ➤ BOUSSIOUF
Prénom ➤ MOURAD
Adresse ➤ 57 rue peyssonnel 13003 Marseille

Titre professionnel visé

Concepteur développeur d'applications

MODALITÉ D'ACCÈS :

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

Présentation du dossier

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

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

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

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

Pour prendre sa décision, le jury dispose :

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

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

Ce dossier comporte :

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

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

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

Sommaire

Exemples de pratique professionnelle

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	p.	5
- Développement d'une API en node-JS – "gestion élèves"	p.5 p.	7
- Persistance des données côté client via Webstorage - application "My visit live"	p.8 p.	10
Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.	11
- Développement d'une calculatrice en react native – Calculator 9000	p.11 p.	13
- Maquette d'interface utilisateur mobile moderne - App Open Speech	p.14 p.	18
- Interface utilisateur pour logiciel de bureau JAVA - E-mail manager	p.19 p.	22
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	p.	23
- Utilisation de l'ORM Sequelize dans une API en node-JS – "gestion élèves"	p.23 p.	26
- Utilisation de la librairie mysql2 dans une API en node-JS – "Open Speech"	p.27 p.	30
Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i>	p.	31
Déclaration sur l'honneur	p.	32
Documents illustrant la pratique professionnelle <i>(facultatif)</i>	p.	33
Annexes <i>(Si le RC le prévoit)</i>	p.	34

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

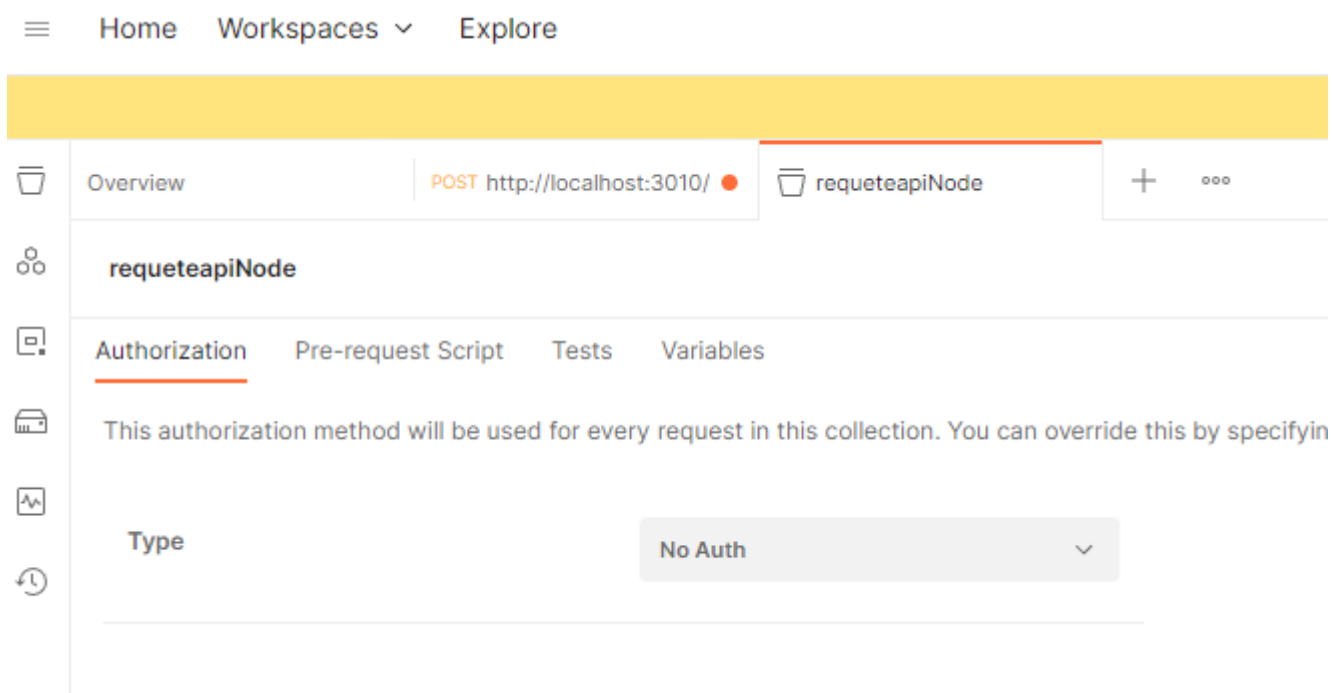
Exemple n°1 - Développement d'une API en node-JS – "gestion élèves"

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

Le projet que j'ai réalisé consistait à développer une API REST en nodeJS pour gérer les requêtes entrantes et gérer une base de données SQL contenant des informations d'élève. Pour ce faire, j'ai utilisé Express.js pour faciliter le routing de l'interface ainsi que Sequelize en tant qu'ORM pour accéder à la base de données SQL.

L'objectif principal de ce projet était de créer une API RESTful qui soit facilement utilisable par des développeurs tiers pour récupérer des informations sur les élèves stockées dans une base de données. Pour cela, j'ai utilisé les meilleures pratiques de développement de l'API REST en termes de sécurité, de performance et de simplicité d'utilisation.

Pour tester les requêtes entrantes et sortantes de l'API REST, j'ai utilisé le logiciel Postman, qui permet de tester facilement les API RESTful en envoyant des requêtes HTTP à l'interface sur ses endpoints et en recevant les réponses correspondantes comme visible sur l'image ci-dessous.



Le choix de NodeJS était motivé par la rapidité d'exécution du code JavaScript, ce qui permet de traiter les requêtes entrantes très rapidement. Express.js a été utilisé pour faciliter le routing de l'API en fournissant des méthodes simples et intuitives pour définir les endpoints de l'interface. Ci-dessous une capture d'écran de la définition de certains endpoints (via la méthode `app.get` propre à expressJS) dans mon IDE (VSC).

```
4 module.exports = function(app) {  
5   app.use(function(req, res, next) {  
6     res.header(  
7       "Access-Control-Allow-Headers",  
8       "x-access-token, Origin, Content-Type, Accept"  
9     );  
10    next();  
11  });  
12  
13  app.get("/api/test/all", controller.allAccess);  
14  
15  app.get("/api/test/all/listeNoms", controller.listeNoms);  
16  
17  app.get("/api/test/all/listeGroupes", controller.listeGroupes);  
18  
19  app.get("/api/test/all/listeGroupesEtNoms", controller.listeGroupesEtNoms);
```

Sequelize a été utilisé en tant qu'ORM (pour accéder à la base de données SQL, ce qui facilite la gestion des opérations CRUD (Create, Read, Update, Delete) en fournissant des méthodes simples pour exécuter ces opérations sur la base de données. Un ORM est une abstraction qui permet de manipuler une base de données SQL en utilisant des objets et des méthodes plutôt qu'en écrivant du SQL brut.

En termes de fonctionnalités, l'API REST permet de récupérer des informations sur les élèves, d'ajouter de nouveaux élèves, de mettre à jour les informations des élèves existants et de supprimer des élèves de la base de données. L'interface a également été conçue pour être facilement extensible pour ajouter de nouvelles fonctionnalités en fonction des besoins futurs.

Le projet que j'ai réalisé consistait en la création d'une API REST en nodeJS pour gérer les requêtes entrantes et gérer une base de données SQL contenant des informations d'élève. Pour cela, j'ai utilisé Express.js pour faciliter le routing de l'API et Sequelize en tant qu'ORM pour accéder à la base de données SQL. Le logiciel Postman a été utilisé pour tester les requêtes entrantes et sortantes de l'API. L'interface a été conçue pour être sécurisée et performante, tout en restant facilement utilisable par des développeurs tiers. En effet, l'interface est facilement extensible pour ajouter de nouvelles fonctionnalités en fonction des besoins futurs. En utilisant le logiciel Postman pour tester les requêtes entrantes et sortantes de l'API, j'ai pu m'assurer que celle-ci fonctionnait parfaitement et répondait aux attentes du projet initial.

DOSSIER PROFESSIONNEL ^(DP)

2. Précisez les moyens utilisés :

Ordinateur sous windows

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Projet d'études

Chantier, atelier, service ▶ Formation concepteur d'applications à La Plateforme

Période d'exercice ▶ Du : 05/10/2022 au : 20/10/2022

5. Informations complémentaires *(facultatif)*

Activité-type 1

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

Exemple n° 2 - Persistance des données côté client via Webstorage - application "My visit live"

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

My Visit Live, une application de visite virtuelle sur le web qui permet aux utilisateurs de visiter des endroits à 360 degrés directement depuis leur navigateur internet. L'application est conçue pour offrir une expérience de visite immersive et interactive aux utilisateurs, qu'ils soient chez eux ou en déplacement.

Cette application Web utilise les technologies modernes du développement web telles que HTML5, CSS3, VUE js et WebGL pour créer une expérience de visite virtuelle en ligne fluide et réactive. Elle permet aux utilisateurs de visiter différents endroits tels que des musées, des galeries d'art, des monuments, des parcs, des jardins, etc. Les utilisateurs peuvent se déplacer librement dans l'espace virtuel.



Lorsque j'ai travaillé sur cette application en tant qu'alternant, j'ai dû gérer des données qui concernaient les préférences des utilisateurs et qui devaient être conservées entre les connexions/déconnexions à l'outil. Afin de résoudre cette problématique j'ai utilisé "LocalStorage".

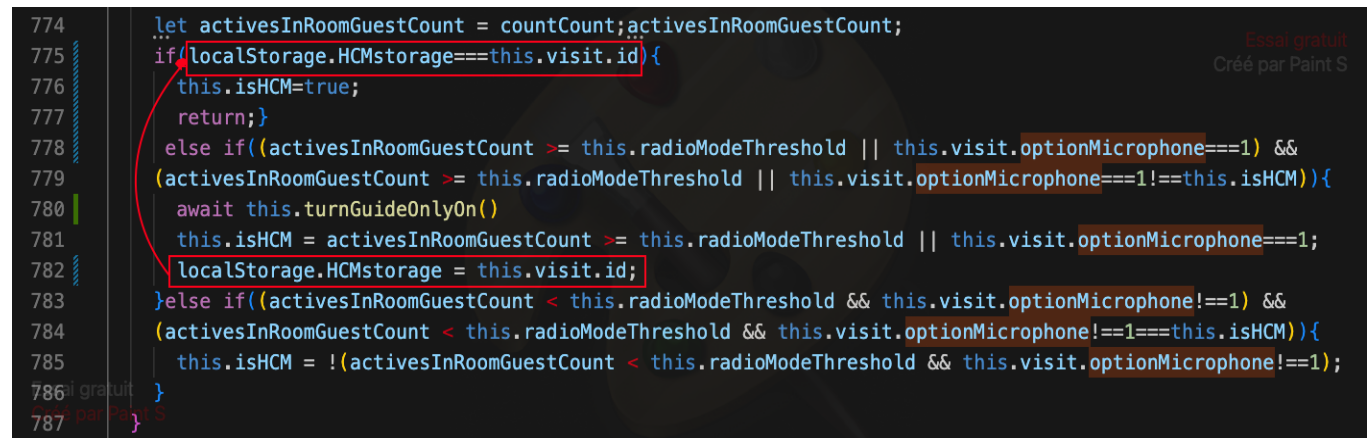
LocalStorage ou plus formellement le Web Storage est une fonctionnalité fournie par les navigateurs modernes et qui permet de stocker des données dans le navigateur de l'utilisateur. Cela signifie que les données peuvent être stockées localement sur l'appareil de l'utilisateur, plutôt que sur un serveur distant.

L'utilisation de LocalStorage présente plusieurs avantages pour notre application. Tout d'abord, cela permet de garantir la persistance des données. Les données stockées dans LocalStorage restent disponibles même après la fermeture du navigateur ou la fermeture de l'application. Cela signifie que les utilisateurs peuvent reprendre leur visite virtuelle là où ils l'ont laissée, même s'ils ont fermé leur navigateur ou leur ordinateur entre-temps.

De plus, l'utilisation de LocalStorage permet de réduire les coûts d'infrastructure, car les données ne

sont pas stockées sur un serveur distant. Cela permet de réduire les coûts liés à la maintenance et à la mise à niveau des serveurs, ce qui est particulièrement utile pour les petites entreprises ou les startups avec des budgets limités, ce qui est le cas pour My Visit Live.

Il convient toutefois de noter que LocalStorage n'est pas une solution parfaite pour la persistance des données. Tout d'abord, la capacité de stockage de LocalStorage est limitée. Les navigateurs modernes permettent généralement de stocker jusqu'à 5 Mo de données dans LocalStorage. Cela signifie que si l'application génère beaucoup de données, il est possible que LocalStorage ne soit pas suffisant. De plus, LocalStorage n'est pas sécurisé par défaut. Les données stockées dans LocalStorage sont accessibles à tout code JavaScript s'exécutant sur la même page web. Cela signifie qu'il est important de prendre des précautions supplémentaires pour garantir la sécurité des données sensibles.



```
774 let activesInRoomGuestCount = countCount;activesInRoomGuestCount;
775 if(localStorage.HCMstorage===this.visit.id){
776   this.isHCM=true;
777   return;}
778 else if((activesInRoomGuestCount >= this.radioModeThreshold || this.visit.optionMicrophone===1) &&
779 (activesInRoomGuestCount >= this.radioModeThreshold || this.visit.optionMicrophone===1!==this.isHCM)){
780   await this.turnGuideOnlyOn()
781   this.isHCM = activesInRoomGuestCount >= this.radioModeThreshold || this.visit.optionMicrophone===1;
782   localStorage.HCMstorage = this.visit.id;
783 }else if((activesInRoomGuestCount < this.radioModeThreshold && this.visit.optionMicrophone!==1) &&
784 (activesInRoomGuestCount < this.radioModeThreshold && this.visit.optionMicrophone!==1===this.isHCM)){
785   this.isHCM = !(activesInRoomGuestCount < this.radioModeThreshold && this.visit.optionMicrophone!==1);
786 }
787 }
```

Comme on le constate sur la capture d'écran ci-dessus, la syntaxe de l'utilisation du LocalStorage dans le framework Vue JS diffère du javascript natif «vanilla». VueJS propose une syntaxe très simple qui permet d'utiliser le web storage via le keyword LocalStorage suivi par le nom de la donnée à stocker prenant la forme d'une simple déclaration de variable. Là où en Javascript Vanilla on doit utiliser les méthodes javascript `setItem()`, `getItem()`, `removeItem()` et `clear()` pour utiliser `window.localStorage`.

Finalement, l'utilisation de LocalStorage a été une solution efficace pour garantir la persistance des données dans l'application My Visit Live sur un cas d'utilisation précis.

Bien que cette solution présente des limites, elle permet bien de fournir une certaine persistance des données tout en réduisant les coûts d'infrastructure. En tant que développeur, j'ai pris soin de prendre en compte les limites de LocalStorage en amont et de ne pas l'utiliser pour des données sensibles qui doivent être stockées uniquement du côté des serveurs dont l'accès et la sécurité sont maîtrisés par l'entreprise.

DOSSIER PROFESSIONNEL ^(DP)

2. Précisez les moyens utilisés :

Macbook

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur cette fonctionnalité.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Entreprise My Tour live durant mon alternance

Chantier, atelier, service ▶ Application MY VISIT LIVE

Période d'exercice ▶ Du : 05/01/2023 au : 20/01/2023

5. Informations complémentaires *(facultatif)*

Activité-type 2

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

Exemple n°1 - Développement d'une calculatrice en react native – Calculator 9000

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

Durant mon apprentissage react JS, j'ai réalisé une calculatrice. J'ai donc dû mettre en place l'interface utilisateur de cette calculatrice.

```
JS Calculator.js      108      }  
JS GreatOperationBut... 109      }  
JS ItSOverNineThous... 110  
JS MagnificentEqual... 111      return (  
JS TheTitle.js       112      <main>  
# App.css            113      <TheTitle />  
JS App.js             114      <div className="Calculator">  
JS App.test.js        115      <ItSOverNineThousand messageOver9000={messageOver9000} />  
# index.css           116      <BeautifulScreen nombre1={nombre1} operator={operator} nombre2={nombre2} />  
JS index.js           117      <GreatOperationButton  
logo.svg              118      |   setTheOperator = {setTheOperator}  
JS reportWebVitals.js 119      |   resetExpression = {resetExpression}  
JS setupTests.js      120      />  
.gitignore            121      <AmazingNumberButton  
package-lock.json     122      |   setFirstNumber = {setFirstNumber}  
package.json          123      |   setSecondNumber = {setSecondNumber}  
README.md             124      />  
                      125      <MagnificentEqualButton  
                      126      |   setTheResult = {setTheResult}  
                      127      />  
                      128      </div>  
                      129      </main>
```

Tout d'abord, j'ai créé un composant principal `<main>` qui englobe toute la structure de ma calculatrice. À l'intérieur de ce composant, j'ai ajouté différents sous-composants pour gérer les différentes parties de la calculatrice.

Le premier sous-composant que j'ai utilisé est `<TheTitle />`, qui affiche simplement le titre de la calculatrice. Cela peut être un en-tête ou une bannière avec un texte explicatif pour aider les utilisateurs à comprendre le but de la calculatrice.

J'ai ajouté un `<div>` avec la classe CSS "Calculator" pour encapsuler tous les autres sous-composants de la calculatrice. Cela me permet de styliser et de positionner facilement l'ensemble de la calculatrice.

J'ai utilisé le sous-composant `<BeautifulScreen />` pour afficher les nombres et l'opérateur de l'opération en cours. J'ai transmis les valeurs des nombres et de l'opérateur en tant que props nommées "nombre1", "operator" et "nombre2" à ce sous-composant. Cela permet au sous-composant de

recupérer ces valeurs et de les afficher à l'écran.

Le sous-composant suivant est `<GreatOperationButton />`. Il s'agit d'un bouton qui déclenche une action spécifique, comme définir l'opérateur sur une valeur donnée ou réinitialiser l'expression. J'ai transmis des fonctions en tant que props nommées "setTheOperator" et "resetExpression" à ce sous-composant. Lorsque le bouton est cliqué, le sous-composant appelle ces fonctions pour effectuer les actions correspondantes.

Pour permettre aux utilisateurs de saisir des nombres, j'ai utilisé le sous-composant `<AmazingNumberButton />`. Ce sous-composant représente un bouton numérique. J'ai transmis des fonctions en tant que props nommées "setFirstNumber" et "setSecondNumber" à ce sous-composant. Lorsqu'un utilisateur clique sur un bouton numérique, le sous-composant appelle ces fonctions pour définir le premier ou le deuxième nombre en fonction du contexte.

Enfin, j'ai utilisé le sous-composant `<MagnificentEqualButton />`. Ce bouton est responsable de l'évaluation de l'expression mathématique et de l'affichage du résultat. J'ai transmis une fonction en tant que prop nommée "setTheResult" à ce sous-composant. Lorsque l'utilisateur clique sur le bouton, le sous-composant appelle cette fonction pour effectuer le calcul et afficher le résultat.

```
#supprbutton {
  background-color: red;
}

#supprbutton:active {
  transform: translateY(4px) translateX(4px);
  box-shadow: 0px 0px 0 0 #405259;
}

#supprbutton:hover{
  background: #a20000;
}
```

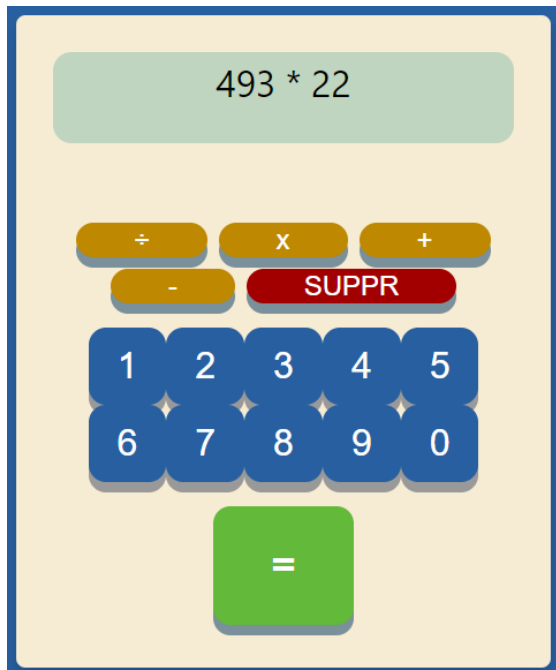
Pour ajouter un effet de relief lorsqu'un bouton est pressé, j'ai utilisé la pseudo-classe ":active" sur les boutons de la classe "boutonsChiffres". Dans cette pseudo-classe, j'ai utilisé la propriété "transform" pour déplacer légèrement le bouton vers le bas et vers la droite en utilisant les valeurs "translateY" et "translateX". Cela crée un effet de pression visuelle lorsque le bouton est activé. J'ai également utilisé la propriété "box-shadow" pour ajouter une ombre légère en dessous du bouton, ce qui

contribue à l'effet de relief.

En survolant les boutons numériques avec la souris, j'ai utilisé la pseudo-classe ":hover" pour appliquer un autre style. J'ai défini la propriété "background" avec une couleur différente pour donner un retour visuel à l'utilisateur lorsqu'il survole les boutons.

J'ai également utilisé des classes similaires pour styliser les autres types de boutons de la calculatrice. Par exemple, j'ai utilisé la classe "buttonEgale" pour les boutons d'égalité et la classe "operateurs" pour les boutons des opérateurs mathématiques. J'ai utilisé des propriétés similaires pour le padding, la taille de la police et la couleur de fond, mais avec des valeurs spécifiques à chaque classe pour obtenir des styles différents.

Voici le résultat de l'interface :



J'ai donc utilisé React JS pour créer une calculatrice avec plusieurs sous-composants qui se chargent de différentes parties de l'interface qui communiquent entre eux. J'ai également réalisé le visuel de toute l'interface grâce à la technologie CSS ainsi qu'à ses propriétés.

2. Précisez les moyens utilisés :

Ordinateur sous windows

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Projet d'études

Chantier, atelier, service ▶ Formation concepteur d'applications à La Plateforme

Période d'exercice ▶ Du : 10/09/2022 au : 20/09/2022

5. Informations complémentaires (facultatif)

Activité-type 2

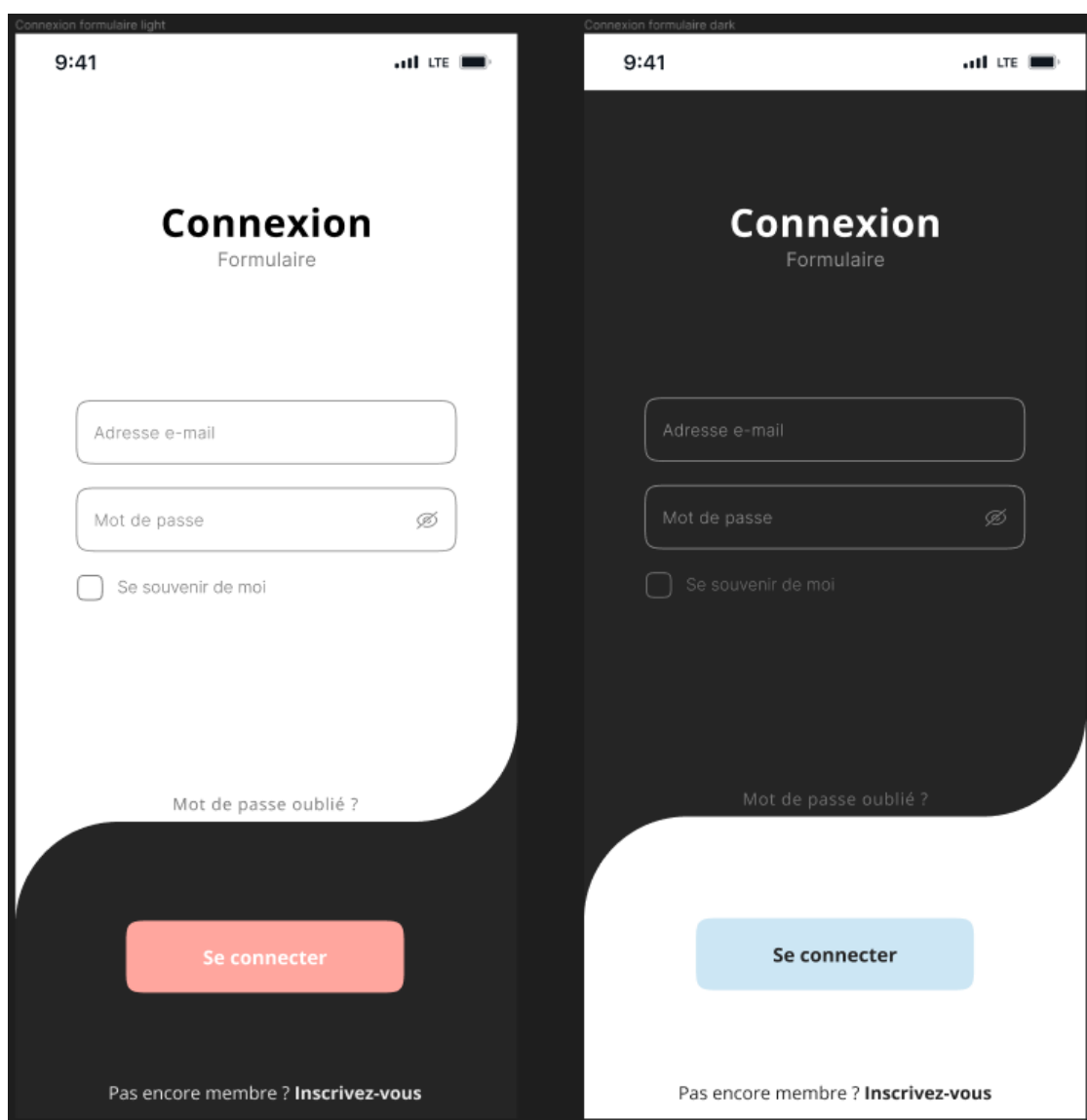
Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°2 - Maquette d'interface utilisateur mobile moderne - App Open Speech

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

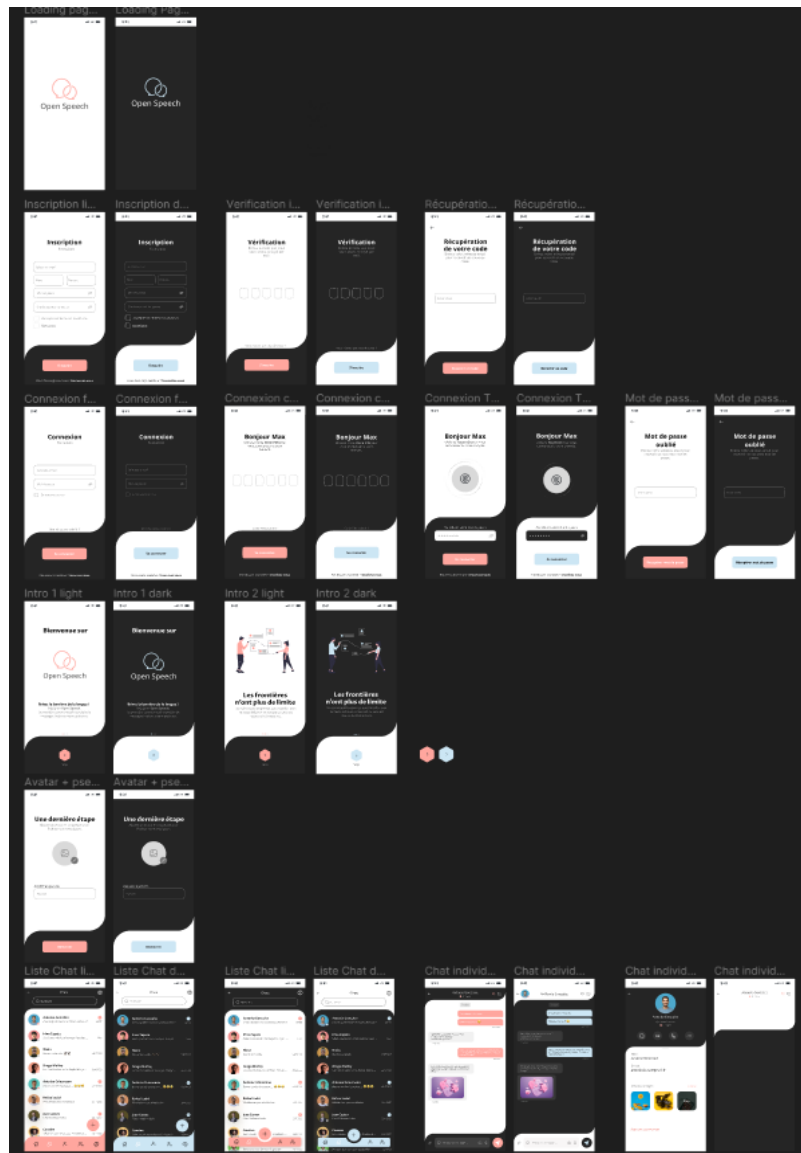
J'ai créé un compte sur le site web de Figma, ce qui m'a donné un accès instantané à leur éditeur en ligne. Une fois connecté, j'ai créé un nouveau projet dédié à ma maquette d'interface utilisateur mobile. J'ai choisi de me concentrer sur une maquette mobile car je crois fermement en l'importance de rester à jour avec les tendances actuelles et de concevoir des interfaces adaptées aux appareils mobiles qui font désormais partie intégrante de notre quotidien.

Passons maintenant à la création du menu. J'ai opté pour un menu à la fois esthétique et fonctionnel, offrant une expérience utilisateur fluide et intuitive. J'ai pris le temps de dessiner le menu avec soin, en utilisant les outils de dessin de Figma tels que les formes, les lignes et les calques. J'ai également intégré des icônes vectorielles de haute qualité pour ajouter des éléments visuels attrayants comme sur l'image à la page suivante.



L'un des aspects importants de ma maquette était la capacité du menu à s'adapter en fonction du thème choisi par l'utilisateur, que ce soit le mode sombre ou le mode clair. Pour cela, j'ai utilisé les fonctionnalités de Figma pour créer des variantes de couleurs. J'ai choisi une palette de couleurs comprenant des tons de rouge et de bleu, qui pouvaient être ajustés dynamiquement en fonction du thème sélectionné. Cette approche permet d'offrir une expérience utilisateur cohérente et agréable, quel que soit le choix de thème.

Lors de la création des environ 30 écrans de ma maquette, j'ai veillé à maintenir une qualité élevée tout au long du processus. J'ai utilisé les fonctionnalités avancées de Figma telles que les grilles, les guides et les alignements pour m'assurer que tous les éléments étaient positionnés avec précision. J'ai également pris soin de peaufiner les détails, en ajustant les espaces, les tailles de police et les marges pour créer une harmonie visuelle.



Outre les fonctionnalités de base de Figma, j'ai également exploré les plugins disponibles pour améliorer mon flux de travail et accélérer certaines tâches. J'ai utilisé d'autres outils pour coordonner les écrans de la maquette tel que Thrello et Slack.

Le choix des couleurs dans la conception d'une interface utilisateur est crucial, car il influence l'expérience des utilisateurs. J'ai sélectionné une palette de couleurs comprenant des tons pastel de bleu, de rouge et des niveaux de gris. Cette combinaison offre une esthétique moderne, agréable et s'adapte aux thèmes sombres et clairs, tout en renforçant l'identité visuelle de l'interface.



Une fois la maquette terminée, j'ai utilisé les fonctionnalités de partage de Figma pour partager mon travail avec les membres de mon équipe. Ils ont pu visualiser les écrans, ajouter des commentaires et proposer des modifications directement dans l'éditeur en ligne.

2. Précisez les moyens utilisés :

laptop sous windows

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Max Machin sur ce projet.

DOSSIER PROFESSIONNEL ^(DP)

4. Contexte

Nom de l'entreprise, organisme ou association ▶ Projet d'études "Open Speech"

Chantier, atelier, service ▶ Formation concepteur d'applications à La Plateforme

Période d'exercice ▶ Du : 15/12/2022 au : 10/04/2023

5. Informations complémentaires *(facultatif)*

Activité-type 2

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

Exemple n°3 - Interface utilisateur pour logiciel de bureau JAVA - E-mail manager

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

J'ai utilisé Swing pour afficher une fenêtre et gérer son aspect visuel dans le cadre d'un logiciel de gestion d'e-mails.

Tout d'abord, j'ai créé une classe appelée "WindowReturnString" qui hérite de la classe JFrame de Swing. Cette classe représente la fenêtre de l'application. Pour rendre l'interface utilisateur plus ergonomique, j'ai décidé d'utiliser une disposition de type FlowLayout pour organiser les composants à l'intérieur de la fenêtre.

J'ai également commencé par définir le titre de la fenêtre en utilisant la méthode "setTitle". Ensuite, j'ai défini la taille de la fenêtre avec la méthode "setSize" en spécifiant une largeur de 300 pixels et une hauteur de 150 pixels. J'ai également utilisé la méthode "setDefaultCloseOperation" pour spécifier que la fenêtre devrait se fermer lorsque l'utilisateur clique sur le bouton de fermeture.

```
public class WindowReturnString extends JFrame {  
  
    private String outputValue;  
    private JTextField textField;  
  
    public String FormWindow() {  
        setTitle("Secteur de profession");  
  
        setSize(300, 150);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setLocationRelativeTo(null);  
    }  
}
```

J'ai aussi créé un objet JPanel pour contenir les composants de la fenêtre. J'ai utilisé la méthode "setLayout" pour définir le gestionnaire de disposition du panel sur FlowLayout. Cela me permet de placer les composants de manière horizontale, un après l'autre.

```
JPanel panel = new JPanel();  
panel.setLayout(new FlowLayout());
```

J'ai ajouté un champ de texte JTextField au panel à l'aide de la classe JTextField. J'ai également créé un bouton JButton avec le texte "Envoyer" et je l'ai ajouté au panel. Pour rendre l'interface plus explicite, j'ai également ajouté des JLabels qui décrivent les options disponibles à l'utilisateur.

```
textField = new JTextField( columns: 10);  
panel.add(textField);  
  
JButton button = new JButton( text: "Envoyer");  
panel.add(button);
```

J'ai ajouté un ActionListener au bouton à l'aide de la méthode "addActionListener". Dans cette implémentation d'ActionListener, j'ai récupéré la valeur saisie par l'utilisateur dans le champ de texte en utilisant la méthode "getText" du JTextField. J'ai effectué toutes les opérations nécessaires sur cette valeur et, dans cet exemple, j'ai simplement affiché la valeur saisie dans la console à l'aide de "System.out.println". J'ai également utilisé la méthode "dispose" pour fermer la fenêtre une fois que l'utilisateur a cliqué sur le bouton. Le dispose() permet de fermer la fenêtre une fois l'entrée saisie.

```
button.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        outputValue = textField.getText();  
  
        System.out.println("Input value: " + outputValue);  
        dispose();  
    }  
});
```

Pour ajouter les JLabels au JPanel qui permettent d'afficher du texte, j'ai créé des objets JLabel correspondant à chaque description que je voulais afficher. Par exemple, j'ai créé des JLabels pour représenter les différentes options disponibles dans le contexte de mon logiciel de gestion d'e-mails.

```
JLabel descriptionLabel1 = new JLabel( text: "Entrez le secteur de profession de l'employé");
JLabel descriptionLabel2 = new JLabel( text: "\\ \n1 pour Sales");
JLabel descriptionLabel3 = new JLabel( text: "\\ \n2 pour Développement");
JLabel descriptionLabel4 = new JLabel( text: "\\ \n3 pour Comptabilité");

panel.add(descriptionLabel1);
panel.add(descriptionLabel2);
panel.add(descriptionLabel3);
panel.add(descriptionLabel4);
```

Puis, j'ai ajouté le panel à la fenêtre en utilisant la méthode "getContentPane().add(panel)". Cela permet de placer le panel à l'intérieur de la fenêtre. J'ai rendu la fenêtre visible en utilisant la méthode "setVisible(true)".

```
getContentPane().add(panel);

setVisible(true);
```

Pour attendre la fermeture de la fenêtre, j'ai utilisé une boucle while avec la méthode "isVisible". Cette boucle permet à la fenêtre de rester ouverte tant qu'elle est visible. J'ai ajouté une légère pause avec "Thread.sleep(100)" pour éviter une utilisation excessive du processeur.

Au final, je peux retourner la valeur saisie par l'utilisateur à partir de la méthode "FormWindow" de la classe au reste des classes pour traiter la donnée saisie.

```
while (isVisible()) {
    try {
        Thread.sleep( millis: 100);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}

return outputValue;
```

DOSSIER PROFESSIONNEL ^(DP)

Voici comment j'ai utilisé Swing pour afficher une fenêtre et gérer son aspect visuel dans le contexte d'un logiciel de gestion d'e-mails. Swing offre de nombreuses possibilités pour créer des interfaces graphiques professionnelles et ergonomiques en Java.

2. Précisez les moyens utilisés :

PC sous windows

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **Projet personnel**

Chantier, atelier, service ▶ **Logiciel de gestion d'email personnalisé**

Période d'exercice ▶ Du : **05/05/2023** au : **15/07/2023**

5. Informations complémentaires *(facultatif)*

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

Exemple n°1 - Utilisation de l'ORM Sequelize dans une API en node-JS – "gestion élèves"

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

Dans mon API Node.js de gestion d'élèves, j'ai utilisé l'ORM Sequelize pour créer une table "USER" et gérer les entités correspondantes. L'utilisation de Sequelize m'a permis de simplifier les opérations de création, de lecture, de mise à jour et de suppression des données de la table "USER" en interagissant avec la base de données.

Après avoir installé Sequelize et le pilote de base de données, j'ai configuré la connexion à la base de données dans mon fichier d'initialisation de l'API. J'ai spécifié les informations de connexion telles que le nom d'utilisateur, le mot de passe, le nom de la base de données, l'hôte, etc

```
module.exports = {  
  ...  
  HOST: "localhost",  
  USER: "root",  
  PASSWORD: "",  
  DB: "annuaire2",  
  dialect: "mysql",  
  pool: {  
    max: 5,  
    min: 0,  
    acquire: 30000,  
    idle: 10000  
  }  
};
```

Une fois la configuration de la base de données terminée, j'ai procédé à la création de la table "USER" à l'aide de Sequelize. Pour cela, j'ai défini un modèle pour la table "USER" en utilisant la fonction `define` de Sequelize. Dans ce modèle, j'ai spécifié les différents attributs de la table, tels que le nom, l'âge, le sexe, etc., ainsi que les types de données correspondants.

Voici le code pour définir le modèle de la table "USER" avec quelques attributs dans mon API nodejs :

```
module.exports = (sequelize, Sequelize) => {  
  ...  
  const User = sequelize.define("users", {  
    username: {  
      type: Sequelize.STRING  
    },  
    email: {  
      type: Sequelize.STRING  
    },  
    password: {  
      type: Sequelize.STRING  
    }  
  });  
  return User;  
};
```

Une fois que j'ai défini le modèle, Sequelize se charge de générer automatiquement les requêtes SQL nécessaires pour créer la table correspondante dans la base de données.

Maintenant que j'ai créé le modèle "User", je peux utiliser les méthodes fournies par Sequelize pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur la table "USER".

Pour créer de nouvelles entités dans la table "USER", j'utilise la méthode create de Sequelize. Cette méthode prend un objet représentant les valeurs des attributs de la nouvelle entité à créer, et elle renvoie une Promesse contenant l'instance de l'entité nouvellement créée.

Voici un exemple de code pour créer une nouvelle entité "User" dans la table "USER" :

```
const Op = db.Sequelize.Op;

var jwt = require("jsonwebtoken");
var bcrypt = require("bcryptjs");

exports.signup = (req, res) => {

  User.create({
    username: req.body.username,
    email: req.body.email,
    firstname: req.body.firstname,
    lastname: req.body.lastname,
    password: bcrypt.hashSync(req.body.password, 8),
  })
```

Ainsi, utiliser un ORM (Object-Relational Mapping) dans le développement d'une application est considéré comme une approche multi-couche en raison de la séparation des préoccupations et de la gestion des différentes couches de l'application.

2. Précisez les moyens utilisés :

J'ai utilisé un PC sous windows lors de la réalisation de cette API.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Projet d'études

Chantier, atelier, service ▶

Formation concepteur d'applications à La Plateforme

DOSSIER PROFESSIONNEL ^(DP)

Période d'exercice ▶ Du : 05/10/2022 au : 20/10/2022

5. Informations complémentaires *(facultatif)*

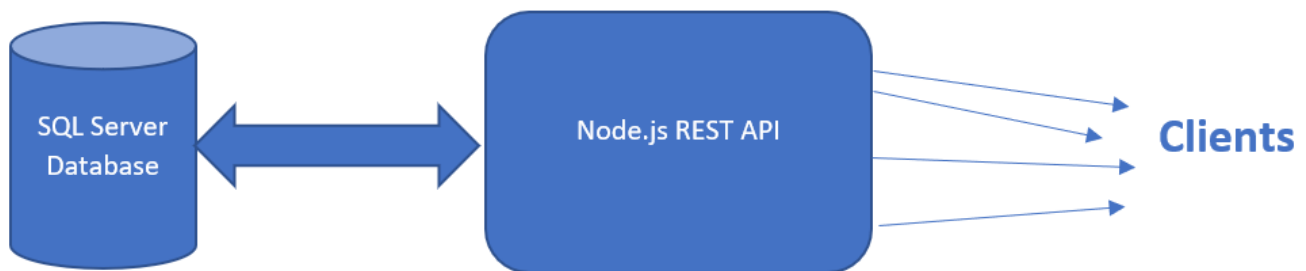
Activité-type 3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n°2 - Cliquez ici pour entrer l'intitulé de l'exemple

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

Nous avons créé une application utilisant Node.js et MySQL. Sur ce projet nous avons utilisé la librairie mysql2 pour assurer une couche d'abstraction avec notre base de données.

Au final, voici le schéma du fonctionnement de l'application résumé le plus simplement possible :



J'ai commencé par installer le module mysql2 en utilisant le gestionnaire de paquets npm. Cette bibliothèque facilite l'interaction avec la base de données MySQL. J'ai exécuté la commande appropriée dans le répertoire de mon application pour installer le module. Ensuite, j'ai pu l'importer en utilisant require.

```
const mysql = require('mysql2');
```

Pour gérer efficacement les connexions à la base de données, j'ai configuré et créé une pool MySQL. J'ai spécifié les informations de connexion telles que l'hôte, le port, le nom d'utilisateur, le mot de passe et le nom de la base de données. J'ai veillé à adapter ces valeurs en fonction de ma configuration. La pool MySQL permet de réutiliser les connexions existantes plutôt que d'en créer une nouvelle à chaque fois, ce qui améliore les performances de l'application.

```
const pool = mysql.createPool({
  host: process.env.DB_HOST || 'localhost',
  user: process.env.DB_USERNAME || 'root',
  password: process.env.DB_PASSWORD || '',
  database: process.env.DB_DBNAME || 'chat-nodejs',
  queueLimit : 0,
  connectionLimit : 0
});

module.exports = pool.promise();
```

Après avoir créé la pool MySQL, j'ai pu l'utiliser pour exécuter des requêtes SQL dans ma base de données. J'ai utilisé la méthode `query` pour envoyer les requêtes SQL et traiter les résultats. Cette méthode m'a permis de supprimer une invitation faite à un utilisateur spécifique. J'ai spécifié l'ID de l'utilisateur dans ma requête et j'ai vérifié si des erreurs s'étaient produites lors de l'exécution de la requête. Le code de cette méthode est disponible à la page suivante.

```
deleteInvitation: async (req, res, next) => {
  try {
    const idInvitations = req.params.id

    const sql = "DELETE FROM invitations WHERE id_invitation = ?"

    const [deleteInv] = await pool.query(sql, idInvitations)

    return res
      .status(200)
      .json({message: "Invitation supprimée avec succès"})
      .end()
  } catch (error) {
    console.log(error)
  }
}
```

En résumé, nous avons réussi à créer une application multicouche avec Node.js et une base de données MySQL en suivant ces étapes. J'ai importé le module mysql2, configuré et créé une pool MySQL, puis utilisé cette pool pour exécuter des requêtes SQL.

Ce projet a été une expérience enrichissante et m'a permis de renforcer mes compétences en développement web et en back end en général.

2. Précisez les moyens utilisés :

Ordinateur portable sous windows

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Max Machin sur ce projet.

4. Contexte

Nom de l'entreprise, organisme ou association - Projet d'études "Open Speech"

DOSSIER PROFESSIONNEL ^(DP)

Chantier, atelier, service ▶

Formation concepteur d'applications à La Plateforme

Période d'exercice ▶ Du : 15/12/2022 au : 10/04/2023

5. Informations complémentaires *(facultatif)*

DOSSIER PROFESSIONNEL ^(DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Développeur web & web mobile	Ministère chargé de l'emploi	06/09/2022

Déclaration sur l'honneur

Je soussigné Mourad Nacim BOUSSIOUF ,
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 **20/06/2023**

pour faire valoir ce que de droit.

Signature : MNB

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

DOSSIER PROFESSIONNEL ^(DP)

ANNEXES

(Si le RC le prévoit)