

<i>Nom de naissance</i>	- Attar
<i>Nom d'usage</i>	- Attar
<i>Prénom</i>	- Latifa
<i>Adresse</i>	- 16 rue des boucheries 93200 Saint-denis

## **Titre professionnel visé**

Développeur web et web mobile

### **MODALITÉ D'ACCÈS :**

- 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.

# DOSSIER PROFESSIONNEL (DP)

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



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

# DOSSIER PROFESSIONNEL (DP)

## Sommaire

### Exemples de pratique professionnelle

<b>Intitulé de l'activité-type n° 1. Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité</b>	<b>p. 7 à 20</b>
▶ Développer la partie front-end d'une application web : maîtriser le maquettage.	p. 7 à 11
▶ Se perfectionner en JavaScript en mettant en pratique les connaissances acquises dans l'ensemble du cours	p. 12 à 16
▶ Création d'une application Angular pour la visualisation des établissements scolaires sur une carte	p. 17 à 20
<b>Intitulé de l'activité-type n° 2. Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité</b>	<b>p. 21 à 32</b>
▶ Implémentation d'une application Web sécurisée avec Flask pour la gestion des utilisateurs	p. 21 à 25
▶ Développement sécurisé de la partie back-end d'une application de réservation d'hôtels en PHP orienté objet (POO)	p. 26 à 29
▶ Portfolio HTML/CSS/Bootstrap avec formulaire sécurisé en PHP	p. 30 à 32

<b>Titres, diplômes, CQP, attestations de formation (facultatif)</b>	<b>p. 33</b>
--	--------------

<b>Déclaration sur l'honneur</b>	<b>p. 34</b>
----------------------------------	--------------

<b>Documents illustrant la pratique professionnelle (facultatif)</b>	<b>p. 35 à 41</b>
--	-------------------

<b>Annexes (Si le RC le prévoit)</b>	<b>p. 42</b>
--------------------------------------	--------------

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



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

## EXEMPLES DE PRATIQUE PROFESSIONNELLE

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 1

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

*Exemple n°1 - Développer la partie front-end d'une application web : maîtriser le maquettage.*

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

Lors de ma formation DWWM chez ESECAD, j'ai réalisé un exercice consistant à créer la maquette de mon CV, avec pour objectif de concevoir un site web responsive.

#### Réalisations :

**Maquettage et Structure :** Utilisation de la grille Bootstrap pour structurer le site et créer un design professionnel et compatible avec tous les navigateurs.

**Personnalisation CSS :** Création d'un fichier CSS personnalisé pour styliser les éléments du site et renforcer l'identité visuelle.

#### Présentation Synthétique des Réalisations Front-End

##### ➤ NavBar Responsive :

- Implémentation d'une barre de navigation avec Bootstrap.
- Interface intuitive et responsive.

```
<nav id="myNavbar" class="navbar navbar-expand-lg sticky-top">
  <div class="container-fluid">
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav"
      aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item m-3">
          <a tabindex="1" class="nav-link active" aria-current="page" href="#about">Moi</a>
        </li>
        ...
      </ul>
    </div>
  </div>
</nav>
```

##### ➤ Section "About Me" :

- Mise en page flexible avec Bootstrap.
- Intégration d'une image et d'un texte descriptif.

# DOSSIER PROFESSIONNEL (DP)

```
<!--Section about me -->
<section id="about" class="sparkle-bg">
<div class="container-fluid p-5">
    <div class="d-flex align-items-center justify-content-around p-5 my-4 content box">
        <div class="heading p-2 me-2">
            <h2>Bonjour, Je suis Latifa</h2>
            <h3>Développeur Web</h3>
        </div>
        <div class="col-8 col-lg-2 image text-align-center">
            
        </div>
    </div>
</div>
</section>
```

## ➤ Progress Bars pour les Compétences :

- Visualisation des compétences avec des progress bars Bootstrap.
- Représentation graphique interactive.

```
<!--Section Compétences -->
<section id="skills" class="m-5 pb-3">
<div class="divider black mx-auto mb-5"></div>
<div class="heading py-2 mb-3">
    <h2>Compétences</h2>
</div>

<div class="container">
    <div class="row m-5">
        <div class="col-lg-6">
            <div class="progress mb-3">
                <div class="progress-bar" role="progressbar" style="width: 100%;" aria-
valuenow="100" aria-valuemin="0" aria-valuemax="100">
                    <h5>HTML 100%</h5>
                </div>
            </div>
        </div> ...
    </div>
</section>
```

## ➤ Timeline pour l'Expérience Professionnelle :

- Création d'une timeline stylisée en CSS/Bootstrap.
- Design esthétique et responsive.

# DOSSIER PROFESSIONNEL (DP)

```
<li>
<div class="timeline-badge"><span class="bi-briefcase-fill"></span></div>
<div class="timeline-panel-container-inverted">
<div class="timeline-panel">
<div class="timeline-heading">
<h4>Développeur Web</h4>
<p class="text-muted"><span class="bi-clock-fill"> 2022-2024</span></p>
</div>
<div class="timeline-body">
<div class="clearfix">
<figure class="col-md-4 float-md-end ms-md-3">

<figcaption>Développeur Web</figcaption>
</figure>
<p>...</p>
...
</li>
```

## ➤ Portfolio avec Cards Bootstrap :

- Présentation des projets avec des cartes Bootstrap.
- Organisation en grille, contenu multimédia.

```
<!--Section Portfolio -->
<section id="portfolio" class="sparkle-bg p-5 my-3">
<div class="divider mx-auto mb-5"></div>
<div class="heading py-2 mb-5">
<h2>portfolio</h2>
</div>
<div class="row row-cols-1 row-cols-md-3 g-5 mx-3 projet">
<div class="col">
<div class="card h-100 text-center">

<div class="card-body">
<h5 class="card-title">JS-se perfectionner- projet n°1</h5>
<div class="d-flex justify-content-around flex-wrap">
<a tabindex="9" href=".//doc/consigne/Se-perfectionner-en-javaScript-devoir2.pdf" class="button p-2 mb-2">Consigne</a>
<a tabindex="10" href="https://notedemusic1.lati.compteweb.com/" target="_blank" class="button p-2 mb-2">Mon projet</a>
</div>
</div>
</div>
</div>
```

# DOSSIER PROFESSIONNEL (DP)

## ➤ Carousel pour les Recommandations :

- Intégration d'un carousel Bootstrap.
- Composant dynamique pour afficher les témoignages.

```
<!--Section Recommandations -->
<section id="recommendations" class="py-5 px-3 m-5">
<div class="divider black mx-auto mb-5"></div>
    <div class="heading py-2 mb-3">
        <h2>recommandations</h2>
    </div>
    <div id="myCarousel" class="carousel carousel-light slide text-center p-5" data-bs-ride="carousel">
        <div class="carousel-indicators">
            <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
            <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="1" aria-label="Slide 2"></button>
            <button type="button" data-bs-target="#myCarousel" data-bs-slide-to="2" aria-label="Slide 3"></button>
        </div>
        <div class="col-10 col-lg-4 mx-auto">
            <div class="carousel-inner">
                <div class="carousel-item active pt-2" >
                    <h3>...</h3>
                    <h4>...</h4>
                    ...
                </div>
            </div>
        </div>
    </div>
</section>
```

## ➤ Personnalisation CSS :

- Affinage du design via un fichier CSS personnalisé.
- Adaptation des styles à ma vision esthétique.

```
***** ABOUT *****/
.content{
    position: relative;
    overflow: hidden;
}
.content::before{
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-image: url(../img/bgportfolio.jpg);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center;
    opacity: 0.8;
    border-radius: 50%;
    animation: beam-animation 5s infinite;
}
```

```
@keyframes beam-animation{
    0%{
        transform: scale(0);
    }
    50%{
        transform: scale(1.5);
    }
    100%{
        opacity: 0;
        transform: scale(2);
    }
}
.img-fluid{
    border-radius: 7px;
}
```

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

- **Technologies** : HTML, CSS, et Bootstrap pour concevoir une interface utilisateur professionnelle et responsive.
- **Environnement de développement** : Visual Studio Code (VSCode).

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans le cadre de mes études à l'école ESECAD.

## 4. Contexte

**Nom de l'entreprise, organisme ou association** ESECAD

**Chantier, atelier, service** ► Développement web

**Période d'exercice** ► Du 08-08-2022 au 10-08-2022

## 5. Informations complémentaires (facultatif)

À partir de cet exercice, j'ai créé mon site web, accessible à l'adresse suivante : <https://lati.compteweb.com/>. J'y ai intégré un formulaire de contact sécurisé développé en PHP, illustrant l'exemple 3 de l'activity-type 2 : "Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité."

Vous pouvez retrouver des captures d'écran illustrant ma pratique professionnelle dans la section "Documents", et accéder directement au projet via ce lien. (<https://lati.compteweb.com>)

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°2 - Se perfectionner en JavaScript en mettant en pratique les connaissances acquises dans l'ensemble du cours

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

Lors de ma formation DWWM chez ESECAD, j'ai travaillé sur un exercice visant à perfectionner mes compétences en JavaScript, en particulier sur l'intégration dynamique d'éléments HTML de manière unobtrusive, ainsi que l'utilisation efficace des boucles et structures de données.

#### Méthodologie

- Analyse : Compréhension des exigences et des fonctionnalités à développer.
- Planification : Définition des étapes de développement et des technologies à utiliser.
- Développement : Écriture du code de manière incrémentale, avec tests à chaque étape.
- Tests et débogage : Utilisation de la console du navigateur pour corriger les erreurs.
- Optimisation : Amélioration de la lisibilité, de la performance et de la maintenabilité du code.

#### Demande 1 : Note de musique en JavaScript unobtrusive

**Création de la structure HTML :** J'ai commencé par une page web avec un élément `<body>` vide, auquel j'ai lié un script JavaScript et une feuille de style CSS.

**Fonctionnalité développée :** L'objectif était de permettre à l'utilisateur de sélectionner une note de musique et d'afficher sa correspondance américaine sans rechargement de la page.

```
!<--CDN Bootstrap.css-->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" >
<link rel="stylesheet" href="style.css">

<!-- Attar Latifa 10/10/2023 Devoir. Se perfectionner en JavaScript Note de musique tout en unobtrusive Javascript-->
<title>Note de musique tout en unobtrusive Javascript</title>
</head>

<body>
<div>
    <h1 class="heading fig p-3 m-5 mx-auto">Note de musique</h1>
</div>
<!--Fichier script-->
<script src="script.js"></script>
</body>
</html>
```

# DOSSIER PROFESSIONNEL (DP)

## Insertion Dynamique d'un Formulaire

Une fois la page chargée (événement onload), j'ai utilisé JavaScript pour insérer dynamiquement un formulaire (<form>) dans l'élément <body>.

```
– □ ×  
  
window.onload = function(){  
    // Créer le formulaire  
    var formElement = document.createElement('form');  
    // Ajouter la classe Bootstrap pour le positionnement en ligne  
    formElement.classList.add('form-line');  
    // Créer un label  
    var labelElement = document.createElement('label');  
    // Ajouter la classe Bootstrap pour les marges  
    labelElement.classList.add('mr-sm-2');  
    labelElement.setAttribute('for', 'noteSelect');  
    labelElement.textContent = 'Choisissez une note de musique:';  
  
    // Centrer le texte  
    labelElement.style.textAlign = 'center';  
    // Définir la marge  
    labelElement.style.margin = '30px';
```

Ce formulaire contenait un menu déroulant (<select>) permettant à l'utilisateur de choisir une note de musique en notation classique. Afin de rendre le JavaScript unobtrusive, j'ai veillé à ne pas insérer de code directement dans les attributs HTML (par exemple, pas d'attribut onclick ou onchange dans les balises HTML).

```
– □ ×  
  
// Créer le menu déroulant(balise <select>)  
var selectElement = document.createElement('select');  
// Ajouter la classe Bootstrap pour la mise en forme du menu déroulant  
selectElement.classList.add('form-control');  
selectElement.id = 'noteSelect';  
// Définir la largeur du select  
selectElement.style.width = '50%';  
// Centrer le select  
selectElement.style.marginLeft = 'auto';  
selectElement.style.marginRight = 'auto';  
// Créer une option vide par default  
var defaultOption = document.createElement('option');  
defaultOption.value = '';  
// Ajouter une option vide au selectElement  
selectElement.appendChild(defaultOption);
```

# DOSSIER PROFESSIONNEL (DP)

Ensuite, j'ai créé un tableau de données (array) en JavaScript pour stocker les correspondances entre les notations musicales classiques et américaines.

```
/*Tableau de "correspondances" contenant les correspondances entre les notes classiques
françaises et les notes américaines*/
var correspondances = [
  ["do", "C"],
  ["ré", "D"],
  ["mi", "E"],
  ["fa", "F"],
  ["sol", "G"],
  ["la", "A"],
  ["si", "B"]
];
```

## Génération Dynamique des Options du Menu Déroulant

À l'aide d'une boucle `foreach`, j'ai parcouru ce tableau pour générer les éléments `<option>` du menu déroulant (`<select>`). J'ai également ajouté une option vide par défaut, permettant à l'utilisateur de ne pas faire de choix initial.

```
// Parcourir toutes les entrées du tableau correspondances
correspondances.forEach(function([noteClassique, correspondanceAmericaine]){
  // Créer un nouvel élément d'option
  var option = document.createElement('option');
  // Définir la valeur et le texte de l'option en utilisant la note classique
  option.value = noteClassique;
  option.textContent = noteClassique;
  // Ajouter l'option nouvellement créée au selectElement
  selectElement.appendChild(option);
});

// Créer une div contenant le message
var messageDisplay = document.createElement('div');
messageDisplay.id = 'messageDisplay';

// Ajouter des éléments au formulaire
formElement.appendChild(labelElement)
  .appendChild(selectElement)
  .appendChild(messageDisplay);
```

## Demande 2 : Interaction avec l'utilisateur

### Écoute de la Modification du Menu Déroulant

Pour répondre à la deuxième demande, j'ai complété le code JavaScript en ajoutant un écouteur

# DOSSIER PROFESSIONNEL (DP)

d'événement (event listener) sur l'élément <select>. Ce dernier détecte les changements de sélection effectués par l'utilisateur.

```
// Ecouter l'événement de modification de sélection
selectElement.addEventListener('change', handleSelectChange);
```

## Affichage de la Correspondance Américaine

Lorsque l'utilisateur sélectionne une note, le script JavaScript récupère la valeur choisie et cherche sa correspondance dans le tableau de données. Une fois trouvée, j'affiche un message informatif à l'utilisateur, par exemple : « La notation américaine pour la note do est C ».

```
// Fonction de traitement de l'événement de modification de sélection
function handleSelectChange(){
    //var selectElement = document.getElementById('noteSelect');
    var selectedNote = selectElement.value;

    var messageDisplay = document.getElementById('messageDisplay');
    messageDisplay.innerHTML = "";

    // Vérifier si la note sélectionnée a une correspondance américaine dans
    // l'objet "correspondances"
    if(selectedNote){
        correspondances.forEach(function([noteClassique, correspondanceAmericaine])
{
            if(noteClassique === selectedNote){
                // Construire le message avec la note sélectionnée et sa
                // correspondance américaine
                var message = 'La notation américaine pour la note ' + selectedNote
                + ' est ' + correspondanceAmericaine;
                // Afficher le message dans l'élément de texte "noteDisplay"
                messageDisplay.textContent = message;
                // Afficher une alerte avec le message
                alert(message);
            }
        });
    }else{ // Si le champ vide est sélectionnée
        // Construire le message si le champ vide sélectionné
        var message = 'Pas de choix fait, veuillez choisir une note';
        // Afficher le message dans l'élément de texte "noteDisplay"
        messageDisplay.textContent = message;
        // Afficher une alerte avec le message
        alert(message);
    }
};
```

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

- **Environnement de développement :** Visual Studio Code (VSCode).
- **Langages utilisés :** HTML, CSS, JavaScript.
- **frameworks :** Bootstrap pour la mise en forme des éléments HTML.
- **Outils de débogage :** Console du navigateur pour le débogage JavaScript.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans le cadre de mes études à l'école ESECAD.

## 4. Contexte

### Nom de l'entreprise, organisme ou association

➤ ESECAD

Chantier, atelier, service ➤ Projet d'évaluation JavaScript

Période d'exercice ➤ Du 09-10-23 au 11-10-2023

## 5. Informations complémentaires (facultatif)

Vous pouvez retrouver des captures d'écran illustrant ma pratique professionnelle dans la section "Documents", et accéder directement au projet via ce lien. (<https://lati.compteweb.com>)

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°3 - *Création d'une application Angular pour la visualisation des établissements scolaires sur une carte*

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

Dans le cadre d'un projet d'évaluation, j'ai développé une application Angular permettant de visualiser les établissements scolaires d'une ville sur une carte interactive.

Les principales tâches incluaient :

➤ **Conception de l'interface utilisateur** : Création de composants Angular pour visualiser les écoles sur une carte dynamique, avec intégration d'une bibliothèque de cartographie et gestion des données géographiques.

```
export class LeafletMapComponent implements AfterViewInit {
  private map : L.Map | undefined

  constructor(private markerService: MarkerService, private schoolService: SchoolService){}

  ngAfterViewInit(): void {
    let x : number = 49.3040; // Latitude de Fameck
    let y : number = 6.1035; // Longitude de Fameck
    let z : number = 15; // Niveau de zoom

    this.map = new L.Map('map', {
      center: [x, y],
      zoom: z
    });

    L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {
      maxZoom: 19,
      attribution: '&copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
    }).addTo(this.map)

    this.schoolService.getSchools().subscribe((schools: School[]) => {
      // Appeler la méthode placeMarker()
      if(this.map){
        this.markerService.placeMarker(this.map, schools);
        console.log('Marker placement completed');
      }
    });
  }
}
```

➤ **Intégration API REST** : Développement d'un service Angular pour interagir avec l'API de l'Annuaire de l'Éducation nationale, en utilisant les services HTTP pour un traitement efficace des données.

# DOSSIER PROFESSIONNEL (DP)

```
@Injectable({
  providedIn: 'root'
})
export class SchoolService {
  constructor(private http: HttpClient) { }

  getSchools(): Observable<School[]>{
    const baseUrl: string =
    'https://data.education.gouv.fr/api/explore/v2.1/catalog/datasets/fr-en-annuaire-education/records?
    select=latitude%2Clongitude%2Cnom_etablissement%2Cadresse_1%2Ccode_postal%2Ctype_et
    ablissemement%2Cstatut_public_prive&where=code_postal%20like%20%2257290%22%20or%20cod
    e_commune%20like%20%22Fameck%22&limit=20&lang=fr';

    return this.http.get(baseUrl).pipe(map((response: any) => {
      const results: any[] = response.results || [];
      const schools: School[] = [];
      for(let i=0; i < results.length; i++){
        const result = results[i];
        const school: School = {
          id: i,
          latitude: result.latitude,
          longitude: result.longitude,
          nom_etablissement: result.nom_etablissement,
          adresse_1: result.adresse_1,
          code_postal: result.code_postal,
          type_etablissement: result.type_etablissement,
          statut_public_prive: result.statut_public_prive
        };
        schools.push(school);
      }
      return schools
    }))
  }

  getSchoolById(schoolId: number): Observable<School | null>{
    return this.getSchools().pipe(map(schools => schools.find(school => school.id
    === schoolId)? null));
  }
}
```

- **Système de navigation** : Mise en place d'une navigation fluide entre les différentes pages de l'application.

# DOSSIER PROFESSIONNEL (DP)

```
const routes: Routes = [
  {path: '', component: HomeComponent},
  {path: 'contact', component: ContactComponent},
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

➤ **Formulaire de contact** : Création d'un composant de formulaire réactif avec validation côté client, permettant aux utilisateurs de contacter directement une école.

```
@Component({
  selector: 'app-form',
  templateUrl: './form.component.html',
  styleUrls: ['./form.component.css']
})
export class FormComponent {
  // Création du formGroup, qui contient chaque FormControl
  form = new FormGroup({
    firstname: new FormControl('', [Validators.required, Validators.minLength(4), Validators.required, Validators.maxLength(20)]),
    lastname: new FormControl('', [Validators.required, Validators.minLength(4), Validators.required, Validators.maxLength(20)]),
    email: new FormControl('', [Validators.required, Validators.email]),
    subject: new FormControl('', [Validators.required]),
    message: new FormControl('', [Validators.required]),
    acceptTerms: new FormControl(false, [Validators.requiredTrue])
  });

  // Getter permettant d'accéder au propriétés depuis notre template
  get firstname() { return this.form.get('firstname');}
  get lastname() { return this.form.get('lastname');}
  get email() { return this.form.get('email');}
  get subject() { return this.form.get('subject');}
  get message() { return this.form.get('message');}
  get acceptTerms() { return this.form.get('acceptTerms');}
}
```

Ce projet m'a permis de démontrer ma maîtrise d'Angular et de ses outils pour répondre à une problématique front-end spécifique. L'application finale offre une expérience utilisateur interactive et simplifiée pour la consultation des informations scolaires et la prise de contact.

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

Pour le développement de l'application:

- j'ai utilisé Angular CLI, TypeScript, et Angular Material pour le design de l'interface.
- La librairie de cartographie Leaflet a été utilisée pour afficher les données géographiques.
- Les appels à l'API étaient gérés via un service Angular dédié, assurant ainsi une séparation claire entre la logique de présentation et la logique de gestion des données.
- Le développement a été effectué sur Visual Studio Code.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans le cadre de mes études à l'école ESECAD.

## 4. Contexte

Nom de l'entreprise, organisme ou association **ESECAD**

Chantier, atelier, service **Développement web**

Période d'exercice **Du 28-11-2023 au 29-11-2023**

## 5. Informations complémentaires (facultatif)

Vous pouvez retrouver des captures d'écran illustrant ma pratique professionnelle dans la section "Documents", et accéder directement au projet via ce lien. (<https://lati.compteweb.com>)

# DOSSIER PROFESSIONNEL (DP)

## Activité-type 2

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

**Exemple n°1 - Implémentation d'une application Web sécurisée avec Flask pour la gestion des utilisateurs**

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

J'ai développé une application Web en Python pour gérer l'inscription, la connexion et la déconnexion des utilisateurs, avec des mesures de sécurité telles que le hachage des mots de passe et la validation des formulaires côté serveur.

➤ J'ai défini la classe PythonSQL pour gérer la connexion à la base de données et l'exécution des requêtes SQL, démontrant ainsi ma capacité à effectuer des opérations comme la sélection et l'insertion de données sécurisées (emails et mots de passe hachés).

```
class PythonSQL:
    def __init__(self):
        self.db = connector.connect(
            user='root',
            password='',
            database='FlaskSqlDevoir',
            host='localhost'
        )
        self.ma_bdd = self.db.cursor()

    def __del__(self):
        self.db.close()

    def selectData(self, req, params=None):
        if params:
            self.ma_bdd.execute(req, params)
        else:
            self.ma_bdd.execute(req)
        return self.ma_bdd.fetchall()

    def insertData(self, table, email, password):
        hashed_password = generate_password_hash(password)
        req = "INSERT INTO users (email, password) VALUES (%s, %s)"
        values = (email, hashed_password)

        self.ma_bdd.execute(req, values)
        self.db.commit()

class User:
    def __init__(self, email, password):
        self.email = email
        self.password = password

    def __repr__(self):
        return f"User('{self.email}')"
```

# DOSSIER PROFESSIONNEL (DP)

➤ J'ai ensuite créé les classes SignupForm et LoginForm pour l'inscription et la connexion des utilisateurs. Elles intègrent des méthodes de validation côté serveur, assurant l'intégrité des données et démontrant ma maîtrise de la validation et de la création de formulaires interactifs.

```
# Formulaire de création de compte
class SignupForm(FlaskForm):
    email = StringField('Email')
    password = PasswordField('Mot de Passe', [
        validators.EqualTo('confirm', message='les mots de passe doivent
correspondre.')]
    )
    confirm = PasswordField('Confirmer le mot de passe.')
    submit = SubmitField('Valider')

    def validate_email(self, email):
        # required côté serveur
        if not email.data:
            raise ValidationError(custom_error_messages['email_required'])

        # Format email
        if not re.match(r"^[^@]+@[^@]+.[^@]+", email.data):
            raise ValidationError(custom_error_messages['email_invalid_format'])

        # Longueur mail
        if len(email.data) < 6 or len(email.data) > 35:
            raise ValidationError(custom_error_messages['email_length'])

        # Vérifier si l'email est déjà connu dans la base de données
        req = f"SELECT * FROM users WHERE email = '{email.data}'"
        results = connexion_sql.selectData(req)
        if results :
            raise ValidationError(custom_error_messages['email_already_exists'])

    def validate_password(self, password):
        if not password.data:
            raise ValidationError(custom_error_messages['password_required'])

        if len(password.data) < 8:
            raise ValidationError(custom_error_messages['password_length'])

        if not re.match(r'^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}$', password.data):
            raise ValidationError(custom_error_messages['password_complexity'])
```

# DOSSIER PROFESSIONNEL (DP)

```
# Formulaire de connexion de compte
class LoginForm(FlaskForm):
    email = StringField('Email')
    password = PasswordField('Mot de Passe')
    submit = SubmitField('Valider')

    def validate_email(self, email):
        # required côté serveur
        if not email.data:
            raise ValidationError(custom_error_messages['email required'])

        # Format email
        if not re.match(r"^[^@]+@[^@]+.[^@]+", email.data):
            raise ValidationError(custom_error_messages['email_invalid_format'])

        # Longueur mail
        if len(email.data) < 6 or len(email.data) > 35:
            raise ValidationError(custom_error_messages['email_length'])

    def validate_password(self, password):
        if not password.data:
            raise ValidationError(custom_error_messages['password_required'])
```

➤ Puis, j'ai défini les Routes pour la page d'accueil ainsi que pour la gestion de l'inscription, la connexion, la déconnexion des utilisateurs, et la page d'erreur.

```
user_connected = []

def isUserCo():
    if len(user_connected) != 0:
        return True
    else:
        return False
```

```
# Route pour la page d'accueil
@app.route('/', methods=['GET'])
def routePrincipal():
    if len(user_connected) != 0:
        contenu = 'Bonjour ' + user_connected[0]
    else:
        contenu = "Bonjour utilisateur anonyme"

    return render_template(
        'home.html',
        title= 'Home',
        data= contenu,
        menu_user_co=isUserCo())
```

# DOSSIER PROFESSIONNEL (DP)

```
# Route pour la page de signup
@app.route('/signup', methods=['GET', 'POST'])
def routeSignup():
    form = SignupForm()

    if request.method == 'POST' and form.validate_on_submit():
        user = User(form.email.data, form.password.data)

        inscription_sql=PythonSQL()
        inscription_sql.insertData('users', user.email, user.password)
        flash('Merci d\'avoir crée votre compte, vous pouvez vous logger',
        'success')
        return render_template(
            'signup.html',
            title="Signup",
            form=form,
            user_inscrit = True,
            menu_user_co=isUserCo()
        )
    else:
        return render_template(
            'signup.html',
            title="Signup",
            form=form,
            menu_user_co=isUserCo()
        )
```

```
# Route pour la page de login
@app.route('/login', methods=['GET', 'POST'])
def routeLogin():
    form = LoginForm()
    print(LoginForm())

    if request.method == 'POST' and form.validate_on_submit():


```

```
# Route pour la deconnexion
@app.route('/logout', methods=['GET'])
def routeLogout():
    user_connected.clear()
    return redirect(url_for('routePrincipal'))
```

# DOSSIER PROFESSIONNEL (DP)

```
@app.errorhandler(404)
def error404(code):
    return render_template(
        'pagenotfound.html',
        title='Page 404',
        menu_user_co=isUserCo()
    )
if __name__=='__main__':
    app.run(debug=True)
```

## 2. Précisez les moyens utilisés :

- Langage de programmation : Python
- Framework : Flask, Bootstrap
- Bibliothèques : Flask-WTF, Werkzeug (sécurisation des mots de passe)
- Système de gestion de base de données : MySQL
- Outils de développement : Éditeur de texte (VS Code)

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans le cadre de mes études à l'école ESECAD.

## 4. Contexte

Nom de l'entreprise, organisme ou association ➤ ESECAD

Chantier, atelier, service ➤ Développement web

Période d'exercice ➤ Du 02-03-2023 au 03-03-2023

## 5. Informations complémentaires (facultatif)

Vous pouvez retrouver des captures d'écran illustrant ma pratique professionnelle dans la section "Documents".

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°2 - Développement sécurisé de la partie back-end d'une application de réservation d'hôtels en PHP orienté objet (POO)

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

Dans le cadre du développement d'une application de gestion de réservations d'hôtels, j'ai conçu et implémenté la logique back-end en utilisant PHP orienté objet (POO). Les principales entités du domaine, telles que **Client**, **Hôtel**, **Chambre**, et **Réservation**, ont été modélisées sous forme de classes pour structurer le code de manière modulaire et maintenable.

J'ai également mis en place une connexion sécurisée à la base de données MySQL en utilisant PDO, garantissant la sécurité des requêtes SQL par le biais de paramètres liés et de requêtes préparées.

```
<?php
require_once ('./Database.php');
require_once ('Class.php');
$database = new Database();
$dbh = $database->getConnection();
```

Cela a permis de réduire les risques d'injections SQL et de protéger les données des utilisateurs.

J'ai défini la **Classe EntityManager** ( Gestionnaire générique pour les opérations sur la base de données) gérant

➤ la connexion à la base de données

```
class EntityManager{
    protected $db; // Propriété protégée pour stocker l'objet de connexion à la bdd
    protected $table; // Propriété protégée pour stocker le nom de la table sur laquelle exécuter les opérations d'insertions

    // Méthode pour définir la connexion à la base de données.
    public function __construct($db){
        $this->setDb($db);
    }
    // prend en argument un objet PDO $db et le stocke dans la propriété $db
    public function setDb(PDO $db){
        $this->db = $db;
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

➤ l'exécution des requêtes SQL, ainsi que la récupération et l'insertion des données. Les erreurs de connexion à la base de données sont gérées à l'aide d'exceptions pour offrir une expérience utilisateur plus robuste.

```
// Méthode pour récupérer les données d'une table à partie de la base de données
public function getData($table){
    $result = array();
    $sql = "SELECT * FROM $table";
    $stmt = $this->db->prepare($sql);

    // Gestion des erreurs SQL
    try {
        $stmt->execute(); // Exécution de la requête
        while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
            $result [] = $row;
        }
    }catch(PDOException $e){
        echo 'Erreur SQL:' . $e->getMessage();
        return null;
    }
    return $result;
}

// Méthode pour récupérer les données d'une table à partie de la bdd par colonne
public function getDataBy($table, $column, $value){

    $sql = "SELECT * FROM $table WHERE $column = :value";
    $stmt = $this->db->prepare($sql);
    $stmt->bindParam(':value', $value);

    // Gestion des erreurs SQL
    try {
        $stmt->execute(); // Exécution de la requête
        // Retourner les données
        $result = $stmt->fetch(PDO::FETCH_ASSOC);

        // Vérifier si les données existent
        if($result){
            return $result;
        }else{
            return array(); // un tableau vide si aucune donnée trouvée
        }
    }catch(PDOException $e){
        echo 'Erreur SQL:' . $e->getMessage();
        return null;
    }
}

// Méthode pour insérer les données à stocker dans la bdd
public function addData($table, $data){
    $columns = implode(', ', array_keys($data));
    $values = implode(', ', array_fill(0, count($data), '?'));

    $sql = "INSERT INTO $table($columns) VALUES ($values)";
    $stmt = $this->db->prepare($sql);

    // Gestion des erreurs SQL
    try {
        $stmt->execute(array_values($data));
    }catch(PDOException $e){
        echo 'Erreur SQL:' . $e->getMessage();
        return null;
    }
}
```

# DOSSIER PROFESSIONNEL (DP)

➤ J'ai ensuite défini les Classes spécifiques pour chaque entité : **ClientManager**, **HotelManager**, **RoomsManager** et **BookingManager**

Ces classes héritent d'**EntityManager** et ajoutent des fonctionnalités spécifiques pour chaque entité si besoin

```
class ClientManager extends EntityManager{  
}  
  
class HotelManager extends EntityManager{  
}  
  
class RoomsManager extends EntityManager{  
}  
  
class BookingManager extends EntityManager{  
    protected $dt_deb_reservation;  
    protected $dt_fin_reservation;  
    protected $id_hotel;  
    protected $roomId;  
  
    public function getFirstAvailableRoom($dt_deb_reservation, $dt_fin_reservation,  
$id_hotel){  
  
        // Requête SQL pour obtenir la première chambre disponible  
        $sql = "SELECT r.id, r.number_rooms, r.id_hotel FROM rooms r  
WHERE r.id_hotel = :id_hotel AND r.id NOT IN  
(SELECT b.id_room FROM booking b WHERE  
b.dt_deb_reservation <= :dt_fin_reservation  
AND b.dt_fin_reservation > :dt_deb_reservation)  
ORDER BY r.id  
LIMIT 1";  
  
        $stmt = $this->db->prepare($sql);  
        $stmt->bindParam(':id_hotel', $id_hotel);  
        $stmt->bindParam(':dt_deb_reservation', $dt_deb_reservation);  
        $stmt->bindParam(':dt_fin_reservation', $dt_fin_reservation);  
  
        // Gestion des erreurs SQL  
        try {  
            $stmt->execute();  
            // Retourner uniquement la première chambre disponible  
            return $stmt->fetch(PDO::FETCH_ASSOC);  
        } catch(PDOException $e){  
            echo 'Erreur SQL:' . $e->getMessage();  
            return null;  
        }  
    }  
  
    // Méthode pour récupérer le dernier id de la table booking  
    public function getLastInsertedId(){  
        return $this->db->lastInsertId();  
    }  
}  
?>
```

# DOSSIER PROFESSIONNEL (DP)

- La logique de validation des réservations a été soigneusement implémentée pour respecter les règles de gestion définies, telles que l'attribution de la première chambre disponible en fonction des critères de réservation.
- J'ai également intégré la fonction generateMenu pour générer dynamiquement le menu de navigation de l'application, améliorant ainsi l'ergonomie de l'interface utilisateur. Les interactions avec la base de données sont centralisées dans les fichiers config.php, database.php, et classManager.php.
- Enfin, la structure du code repose sur les principes de la programmation orientée objet, avec un usage judicieux de l'héritage pour réutiliser le code commun entre les différentes entités, assurant ainsi la modularité et la maintenabilité de l'application.

## 2. Précisez les moyens utilisés :

**WampServer** : Environnement de développement local pour exécuter et tester l'application PHP .

**Visual Studio Code** : Éditeur de code utilisé pour développer la logique back-end en PHP orienté objet (POO).

**Bootstrap et CSS** : Pour un design réactif et personnalisé de l'application.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans le cadre de mes études à l'école ESECAD.

## 4. Contexte

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

**Chantier, atelier, service** ▶ Développement web

**Période d'exercice** ▶ Du 20-09-2023 au 01-10-2023

## 5. Informations complémentaires (facultatif)

Vous pouvez retrouver des captures d'écran illustrant ma pratique professionnelle dans la section "Documents", et accéder directement au projet via ce lien. (<https://lati.compteweb.com>)

# DOSSIER PROFESSIONNEL (DP)

## Exemple n°3 - Portfolio HTML/CSS/Bootstrap avec formulaire sécurisé en PHP

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

Dans le cadre du développement de mon portfolio personnel, j'ai conçu et développé une fonctionnalité de formulaire de contact sécurisé en utilisant HTML, CSS, Bootstrap pour la partie front-end, et PHP pour la partie back-end. Mon objectif principal était d'intégrer des mesures de sécurité afin de protéger les données soumises par les utilisateurs.

#### ➤ Conception et Développement du Portfolio :

Développement de la structure du site en HTML, stylisée avec CSS et Bootstrap pour une interface utilisateur responsive.

(cf Activité-type 1 Exemple 1)

Intégration d'un formulaire de contact pour permettre aux visiteurs de me contacter directement depuis le site.

```
<!--Section Contact -->
<section id="contact" class="p-5 m-5">
    <div class="heading py-2 mb-5">
        <h2>Contactez-moi</h2>
    </div>

    <div class="row">
        <div class="col-lg-8 mx-auto">
            <form id="contact-form" method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>#contact">
```

#### ➤ Sécurisation du Formulaire de Contact :

**Validation Côté Serveur :** Implémentation d'une validation des données en PHP pour garantir l'intégrité des informations (exemple : vérification des formats d'email ).

**Sanitization des Entrées Utilisateur :** Utilisation de fonctions PHP comme htmlspecialchars(), trim(), et stripslashes() pour éviter les attaques XSS et autres injections malveillantes.

**Envoi Sécurisé des Données :** Mise en place de l'envoi d'emails via PHP avec des en-têtes de sécurité pour prévenir les tentatives de falsification des emails.

# DOSSIER PROFESSIONNEL (DP)

```
<?php
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $firstname = verifyInput($_POST['firstname']);
    $name = verifyInput($_POST['name']);
    $email = verifyInput($_POST['email']);
    $phone = verifyInput($_POST['phone']);
    $message = verifyInput($_POST['message']);
    $isSuccess = true;
    $emailText = "";

    if(!isEmail($email)){
        $emailError = "Donnez moi un email correct, s'il vous plaît !";
        $isSuccess = false;
    }else{
        $emailText .= "email: $email\n";
    }

    if($isSuccess){
        $headers = "From: $firstname $name <$email>\r\nReply-To: $email";
        $headers .= "Content-Type: text/plain; charset=utf-8\r\n";
        mail($emailTo, "Un message de votre site", $emailText, $headers);
        $firstname = $name = $email = $phone = $message = '';
    }
}

function isEmail($var){
    return filter_var($var, FILTER_VALIDATE_EMAIL);
}

function verifyInput($var){
    $var = trim($var ?? '');
    $var = stripslashes($var);
    $var = htmlspecialchars($var);

    return $var;
}

?>
```

```
<div class="row">
    <div class="col-md-6 p-3">
        <div class="mb-3">
            <label for="email" class="form-label">Address email<span class="blue">*</span></label>
            <input tabindex="10" name="email" id="email" type="email" autocomplete="off"
                class="formcontrol"
                placeholder="name@example.com" aria-label="Adresse mail" value="<?php echo $email; ?>">
            <p class="comments mt-2"><?php echo $emailError; ?></p>
        </div>
    </div>
```

```
<p class="thank-you mt-3" style="display:<?php if($isSuccess) echo 'block'; else echo 'none';?>">Votre
message a bien été envoyé. Merci d'avoir pris contact !</p>
```

# DOSSIER PROFESSIONNEL (DP)

## 2. Précisez les moyens utilisés :

- **Technologies** : HTML, CSS, et Bootstrap pour concevoir une interface utilisateur professionnelle et responsive.
- **Back-End** : PHP pour le traitement sécurisé des données du formulaire.
- **Environnement de développement** : Visual Studio Code (VSCode).
- **Test** : Envoi réussi d'un email via le formulaire de contact, avec réception de toutes les données utilisateur sur ma boîte mail.

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

## 4. Contexte

Nom de l'entreprise, organisme ou association ➤ ESECAD

Chantier, atelier, service ➤ Développement web

Période d'exercice ➤ Du 10-02-2024 au 14-02-2024

## 5. Informations complémentaires (facultatif)

Vous pouvez retrouver des captures d'écran illustrant ma pratique professionnelle dans la section "Documents", et accéder directement au projet via ce lien. (<https://lati.compteweb.com>)

# DOSSIER PROFESSIONNEL (DP)

## Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Infirmière Diplomée d'état	<b>Ifsi de clermont-de-l'oise</b>	13-09-2000 au 27-11-2003
DWWM	<b>Esecad</b>	2022 - 2024
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

# DOSSIER PROFESSIONNEL (DP)

## Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] *Latifa Attar*, 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 *Saint-denis* pour faire valoir ce que de droit.

*Le 24-04-2024*

Signature : Attar Latifa

# DOSSIER PROFESSIONNEL (DP)

## Documents illustrant la pratique professionnelle

(facultatif)

### Capture d'écran illustrant les différents exemples présentés dans Activité-type 1 et 2

#### Activité-type1 Exemple 1

The image displays four screenshots of a professional portfolio website. The top-left screenshot shows a header with a photo of a woman with glasses, the text 'BONJOUR, JE SUIS LATIFA' and 'Développeur Web', and a background of binary code. The top-right screenshot shows a grid of project cards: 'JS-se perfectionner- projet n°1' (with 'CONSULTE' and 'MON PROJET' buttons), 'JS-se perfectionner- projet n°2' (with 'CONSULTE' and 'MON PROJET' buttons), 'PHP-expert-projet n°1' (with 'CONSULTE' and 'MON PROJET' buttons), 'Créer une application Angular' (with 'CONSULTE' and 'MON PROJET' buttons), and 'PHP-expert-projet n°2' (with 'CONSULTE' and 'MON PROJET' buttons). The bottom-left screenshot shows a 'FORMATION' section with three cards: '1995-1998 Lycée Saint-Exupéry-Fameck Baccalauréat Technologique', '2000-2001 IFSI Clermont-de-l'Oise Formation Développeur d'application', and '2002-2005 ESECAD Skill and you Formation Développeur Web'. The bottom-right screenshot shows a 'COMPETENCES' section with three cards: 'HTML/CSS', 'JavaScript', and 'AngularJS'. The right side of the image shows a larger view of the website's main page, featuring a 'FORMATION' section with a text box about learning to develop a web application, a 'COMPETENCES' section with a text box about learning various programming languages, and a 'DEVELOPPEUR WEB' section with a text box about developing a web application using Angular and a screenshot of the application.

# DOSSIER PROFESSIONNEL (DP)

## Activité-type1 Exemple 2

**NOTE DE MUSIQUE**



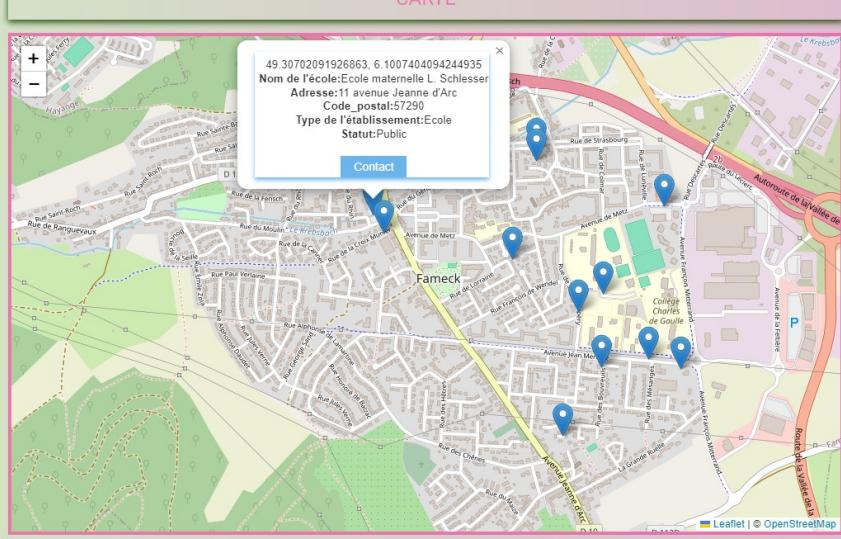
Choisissez une note de musique:  
do

La notation américaine pour la note do est C



## Activité-type 1 Exemple 3

**CARTE**



49.30702091926863, 6.1007404094244935  
Nom de l'école:Ecole maternelle L. Schlessier  
Adresse:11 avenue Jeanne d'Arc  
Code\_postal:57290  
Type de l'établissement:Ecole  
Statut:Public

Contact

Leaflet | © OpenStreetMap

# DOSSIER PROFESSIONNEL (DP)

## FORMULAIRE DE CONTACT: ECOLE MATERNELLE L. SCHLESSER

Nom:\*

Prénom:\*

Email:\*

Objet de contact:\*

Message:\*

J'accepte les conditions générales d'utilisation

VALIDER

## Activité-type 2 Exemple 1

Merci d'avoir crée votre compte, vous pouvez vous logger

Email  
ag@gmail.com

Mot de Passe

Confirmer le mot de passe

VALIDER

Page Login

# DOSSIER PROFESSIONNEL (DP)

Serveur : MySQL:3306 » Base de données : flasksqldevoir » Table : users

Parcourir Structure SQL Rechercher Insérer Exporter Importer Priv

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0008 seconde(s).)

```
SELECT * FROM `users`
```

Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table

Options supplémentaires

	id	email	password
<input type="checkbox"/>	17	ag@gmail.com	scrypt:32768:8:1\$vAz2KvB7OvvTeS4h\$78583adc709f6bac...

Éditer Copier Supprimer Exporter

WEBPYTHONFLASK

HOME LOGIN SIGNUP

Email  
ag@gmail.com

Mot de Passe  
.....

VALIDER

# DOSSIER PROFESSIONNEL (DP)



WEBPYTHONFLASK

HOME

LOGOUT

Vous êtes connecté(e)

Bonjour.ag@gmail.com

## Activité-type2 Exemple 2



ACCUEIL LISTES DES HOTELS RÉSERVATIONS

APPLICATION BOOKING



# DOSSIER PROFESSIONNEL (DP)



ACCUEIL LISTES DES HOTELS RÉSERVATIONS

## LISTES DES HOTELS



### Mandarin

251 rue Saint Honoré, 75001 Paris - France



Date de début de réservation:  
26/05/2024

Date de fin de réservation:  
27/05/2024

### Hotel la Comtesse

29, Avenue de Tourville, 75 007 Paris - France



Date de début de réservation:  
jj/mm/aaaa

Date de fin de réservation:  
jj/mm/aaaa

[Vérifier Disponibilité](#)

Première chambre disponible: 101

[Réserver](#)

[Vérifier Disponibilité](#)

Première chambre disponible: 101

**Paris j'Adore Hotel & Spa**  
7 rue Beaudant, 75 017 Paris - France



**Au Royal Mad**  
68, rue Sedaine, 75 011 Paris - France



ACCUEIL LISTES DES HOTELS RÉSERVATIONS

## RÉSERVATIONS



Nom\*

Lati

Email\*

al@gmail.com

Hotel

Mandarin

Chambre attribuée

101

Début de réservation

2024-05-26

Fin de réservation

2024-05-27

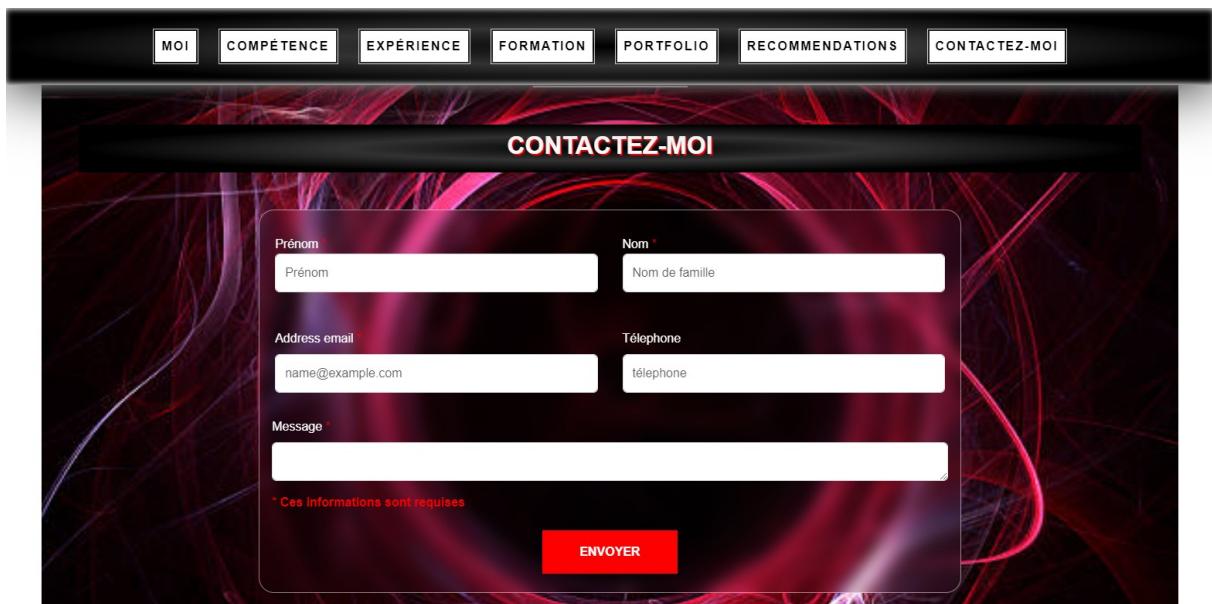
\* Ces informations sont requises

Accepter les termes et conditions

[Réserver](#)

# DOSSIER PROFESSIONNEL (DP)

## Activité-type2 Exemple 3



The screenshot shows a contact form titled "CONTACTEZ-MOI" on a dark background with a red and black abstract pattern. The form is divided into two columns. The left column contains fields for "Prénom" (First name) and "Address email". The right column contains fields for "Nom" (Last name) and "Téléphone" (Phone). Below these fields is a large "Message" text area. At the bottom left of the form, a note states "\* Ces informations sont requises" (These informations are required). At the bottom right is a red "ENVOYER" (Send) button.

MOI COMPÉTENCE EXPÉRIENCE FORMATION PORTFOLIO RECOMMENDATIONS CONTACTEZ-MOI

**CONTACTEZ-MOI**

Prénom \*  
Prénom

Nom \*  
Nom de famille

Address email \*  
name@example.com

Téléphone  
téléphone

Message \*

\* Ces informations sont requises

ENVOYER

## ANNEXES

*(Si le RC le prévoit)*