|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | Anglade |
| *Nom d’usage* |  | Anglade |
| *Prénom* |  | Jessica |
| *Adresse* |  | 10 lieu-dit capelanios 82400 POMMEVIC |
|  |  |

|  |  |
| --- | --- |
| **Titre professionnel visé** | |
|  | |
| Développeur web et web mobile | |
|  | |
| **Modalité d’accès :** | |
|  | |
|  | Parcours de formation |
|  | Validation des Acquis de l’Expérience (VAE) |
|  | |

**1**

|  |  |  |  |
| --- | --- | --- | --- |
| **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**](http://travail-emploi.gouv.fr/titres-professionnels) | |

**2**

**3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sommaire** | | | |
| **Exemples de pratique professionnelle** | | | |
| **Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | **p.** |  |
|  | |  |  |
| **Compétences liées à l’activité :**   * Maquetter une application * Réaliser une interface utilisateur web statique et adaptable * Développer une interface utilisateur web dynamique * Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce | |  |  |
|  |  Intitulé de l’exemple n° 1  J’ai réalisé une maquette pour une application mobile | p. |  |
|  |  Intitulé de l’exemple n° 2  J’ai réalisé un site vitrine responsive | p. |  |
|  |  Intitulé de l’exemple n° 3 | p. |  |
|  |  |  |  |
| **Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | **p.** |  |
|  | |  |  |
| **Compétences liées à l’activité :**   * Créer une base de données * Développer les composants d’accès aux données * Développer la partie back-end d’une application web ou web mobile * Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce | |  |  |
|  |  Intitulé de l’exemple n° 1  J’ai mis en place un Webservice | p. |  |
|  |  Intitulé de l’exemple n° 2 J’ai réalisé une base de données avec ses requêtes | p. |  |
|  |  Intitulé de l’exemple n° 3  J’ai créé une application météo | p. |  |
|  |  |  |  |
|  |  |  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | p. |  |
| **Déclaration sur l’honneur** | | p. |  |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | p. |  |
| **Annexes** *(Si le RC le prévoit)* | | p. |  |

**Exemples de pratique**

**professionnelle**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **1** | | | **Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°1***  | | | | ***J’ai réalisé une maquette pour une application mobile*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| **Afin de tester nos connaissances globales, on nous a donné un exercice en groupe qui consistait à créer une application android récupérant deux API, une donnant la localisation des stations de vélo et une autre les stations de métro sur Toulouse afin de les afficher. J’ai donc réalisé une maquette avec mon groupe.**    **Pour commencer, j’ai créé le zoning pour avoir une vue d’ensemble des éléments apparaissant sur l’application. J’ai mis trois blocs : un pour le titre, un pour la carte et enfin un pour le filtrage.**    **Ensuite, j’ai réalisé le Wireframe. Il n’y a plus que des blocs, j’ai mis du texte à la place du titre la carte à une icon et le filtrage est en checkbox.**  **Pour finir, j’ai créé le mockup. C’est la version finale de la maquette, ce à quoi l’application ressemblera. J’ai ajouté les couleurs, le vrai texte et l’image de la carte.** | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| **Pour la maquette, j’ai utilisé le logiciel adobe XD.** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| **J’ai réalisé cette maquette en groupe avec 3 autres personnes.** | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***ADRAR*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **TP** | | | | | |
| **Période d’exercice** | |  | **Du** | | **13/12/2021** | | **au** | **13/12/2021** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| * **J’ai réalisé la maquette avec ses trois étapes** * **Le travail de groupe m’a permis de partager nos opinions** * **J’ai pu utiliser adobe XD.** | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **1** | | | **Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°2***  | | | | ***J’ai réalisé un site vitrine responsive*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| **Lors de mon stage, on m’a donné comme mission de créé un site vitrine pour un centre de formation. Je vais vous présenter le code de la page d’accueil. Je commence par mettre en place le site en langue française avec « lang ». Ensuite, je relie bootstrap au fichier avec « link » pour le CSS et « script » pour le javascript. Je lis également un fichier CSS personnalisé.**  <!DOCTYPE html> <html lang="fr"> <head>  <meta charset="UTF-8">  <title>Nom de l’entreprise</title>  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>  <link rel="stylesheet" href="css/style.css" > </head>  **Je créé ensuite le header de la page en utilisant la balise « header » dans la balise « body ». Je mets ensuite la balise « nav » dans le header auquel j’agrémente les classes de mise en forme de bootstrap. Par exemple la classe « navbar-expand-lg » permet de rendre la barre de navigation responsive. Je créé ensuite un container avec la balise « div » dans laquelle je mets une image, balise « img » avec pour source, balise « src », le logo dans le dossier img. Cette balise se trouve dans la balise « a » qui sert à mettre un lien sur l’image. L’attribut « href » donne le chemin pour atteindre la page à ouvrir. Ensuite, je créais une balise « div » dans laquelle j’utilise une liste à puce avec les balises « ul » et « li » et ajoute la balise « a » pour faire les liens.**  <body> <header>  <nav class="navbar navbar-expand-lg navbar-light">  <div class="container-fluid">  <a class="navbar-brand" href="./index.php"><img class="logo" src="img/logo.png"></a>  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  <span class="navbar-toggler-icon"></span>  </button>  <div class="collapse navbar-collapse barNav" id="navbarSupportedContent">  <ul class="navbar-nav me-auto mb-2 mb-lg-0">  <li>  <a href="./a\_propos.php">A propos</a>  </li>  <li>  <a aria-current="page" href="./formations.php">Formations</a>  </li>  <li class="dropdown ">  <a class="dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">Particulier</a>  <ul class="dropdown-menu dd\_menu" aria-labelledby="navbarDropdown">  <li><a href="./formations.php">Formations</a></li>  <li><a href="./info\_admin.php">Informations administratives</a></li>  </ul>  </li>  </ul>  </div>  </div>  </nav> </header>  **Une fois le header terminer et tester, je passe au centre de la page. Je créais une balise « section » avec une classe personnalisé nommé « banner ». Ensuite, j’y mets une image et dans une balise « div », je mets un titre en « h2 » et un paragraphe dans la balise « p ».**  <section class="banner">  <img src="img/centre\_c2rt.jpg" alt="photo du centre">  <div>  <h2 class="title\_index">Presentation entreprise</h2>  <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>  </div> </section>  **Du coté de CSS, je mets la section de classe « banner avec un margin à 0 et le place en position relative. J’utilise les sélecteurs pour sélectionner l’image dans la section avec « .banner img » afin de gérer sa taille et son opacité. Pour la div contenant le titre et le paragraphe en position absolu. J’utilise ensuite le transform ainsi que top, et left afin de placer où je le désire, c’est-à-dire au centre de l’image. Je mets ensuite en forme le titre. Je place toutes les tailles d’image et les div en pourcentage afin qui s’adapte à l’écran. La taille des textes est mise en vw qui est une unité qui s’adapte à la taille en longueur de l’écran.**  .banner{  margin: 0;  position: relative; } .banner img{  width: 100%;  opacity: 70%; } .banner div{  position: absolute;  transform: translate(-50%,-50%);  top: 50%;  left: 50%;  width: 80%; } .banner p{  font-size: 1.1vw;  color: white;  font-weight: bold;  text-align: justify;  text-shadow: black 2px 2px 3px; } .title\_index{  color: white;  font-weight: bold;  text-transform: uppercase;  font-size: 3vw;  text-shadow: black 2px 2px 3px; }  **Je rends ensuite la bannière responsive. J’utilise donc le @media avec longueur maximum d’écran de 800px, c’est à dire que quand la fenêtre est en dessous de 800 pixels le code dans le media seront alors effective et quand il passera au-dessus, il ne les prendra plus en compte. Ici, l’image de la bannière augmentera à 170% et le paragraphe aura une taille de 2vw.**  */\*--------------------responsive banner---------------------\*/* @media (max-width: 800px) {  .banner img{  width: 170%;  }  .banner p{  font-size: 2vw;  }  }  **Ensuite, je créais les deux boutons demandés par le client. Comme il ne servent qu’à faire des liens vers d’autres pages, je préfère utiliser les balises <a>, <img> et <p> que la balise <button> pour un meilleur rendu visuel.**  <section class="bt\_accueil">  <a href="particulier.php">  <img src="img/particulier.jpg" alt="image pour les particuliers">  <p>Particulier</p>  </a>  <a href="professionnel.php">  <img src="img/professionnel.jpg" alt="image pour les professionnel">  <p>Professionnel</p>  </a> </section>  **Côté CSS, je vais d’abord mettre la section entière que j’ai mis en classe « bt\_accueil » en flexbox grâce display. Une flexbox est une méthode de mise en page permettant de disposer des éléments plus facilement. Ensuite je précise que les flexbox sont en ligne, j’ajoute également le retour à la ligne si elles n’ont pas assez de place avec « wrap ». Dans la page d’accueil, elle ne sert pas vraiment mais j’ai aussi utilisé cette classe pour des pages de transition où elle est utile. Je lui indique ensuite de les espacer avec « space-evenly ».**  **Je réutilisais comme tout à l’heure les positions afin de pouvoir placer le texte au centre de l’image.**  .bt\_accueil {  display: flex;  flex-direction: row;  flex-wrap: wrap;  justify-content: space-evenly; } .bt\_accueil a{  position: relative;  width: 40%;  margin: 30px; } .bt\_accueil img {  width: 100%; } .bt\_accueil p{  position: absolute;  transform: translate(-50%,-50%);  top: 50%;  left: 50%;  font-size: 3vw;  width: 85%;  … }  **Je rends ensuite cette section responsive. Tout comme tout à l’heure, j’utilise @media pour indiquer les modifications à apporter lorsque la fenêtre descend sous les 800px. Je change donc juste les tailles du paragraphe et du bloc « a » ainsi que je passe la section en direction vertical, en colonne. Pour optimiser le code, j’aurai pu mettre tout le code responsive dans le même @media, mais pour un soucis d’organisation pour celui qui reprendra la suite (qui est débutant), j’ai préféré ce système.**  */\*---------responsive index + transitions--------------------------\*/* @media (max-width: 800px) {  .bt\_accueil {  display: flex;  flex-direction: column;  }  .bt\_accueil p{  font-size: 5vw;  }  .bt\_accueil a{  width: 90%;  } }  **Je vais donc maintenant passer au pied de la page. Dans un premier temps le client à demander d’avoir la possibilité de s’abonner à une newsletter donc j’ai mis une balise <hr> qui est une rupture thématique afin de faire ressortir la différence entre la page et le pied de page. Je lui ai mis une classe afin de pouvoir modifier sa taille et son épaisseur. J’ai ensuite créé une section qui a pour classe «newsletter ». Dedans je lui ai mis un titre d’importance 5 avec la balise <h5>. J’ai ensuite créé un formulaire avec la balise <form> qui est en méthode « post », c’est-à-dire qu’elle envoie une information. L’action est pour l’instant vide puisque la partie back-end n’était pas encore mis en place. J’ai ensuite mis dans le formulaire une zone de texte à remplir avec « input » et l’attribut « type=** "**e-mail**"**» qui permet de sécuriser les envoie afin que seule une adresse mail puisse être envoyé. Dans cette zone, j’y place également un texte qui s’effacera quand on écrira dedans afin d’aiguiller l’utilisateur de ce qui doit mettre. Pour cela j’utilise l’attribut « placeholder ». Je mets ensuite un attribut « aria-label » qui permet de définir une légende non-visible. Je mets ensuite la balise <button> qui permet d’envoyer la donnée au serveur. J’ai placé une icon sur le bouton à la place du texte.**  <hr class="hrNewsletter"> <section class="newsletter">  <h5>Abonnez-vous à notre newsletter</h5>  <form class="input-group mb-3 group\_newsletter" action="" method="post">  <input type="email" class="form-control" placeholder="Votre adresse email" aria-label="email" aria-describedby="button-addon2">  <button class="btn btn-outline-dark" type="button" id="button-addon2" name="mail"><img class="icon\_mail" src="img/icon\_mail.png"></button>  </form> </section>  **Pour ce qui est du CSS personnalisé de la section newsletter, je n’ai réalisé que des modifications de mise en forme (couleur, taille, …), rien de plus que ce que j’ai déjà utilisé.**  **J’ai ensuite créé une un footer grâce à la balise du même nom ou se trouve des balises <div> avec à l’intérieur des paragraphes contenant les coordonnées du client et enfin une autre <div> avec des liens vers les pages légales :**  <footer>  <div class="first\_div">  <div>  <p>NOM</p>  <p>adresse</p>  <p>CP VILLE</p>  </div>  <div>  <p>Tél : 00 00 00 00 00</p>  <p>Mail : mail@boite.fr</p>  </div>  </div>  <div class="last\_div">  <a href="./mention\_legale.php"><p>Mentions legales</p></a>  <a href="./confidentialite.php"><p>Politique de confidentialité</p></a>  </div> </footer>  </body> </html>  **Pour ce qui est du CSS, j’ai utilisé les flexbox, les différentes mises en forme déjà utilisé.**  **Puisque les pages du site ont la même en-tête et pied de page, j’ai décidé d’utiliser php afin d’éviter de répéter le même code sur chacune d’elle. Pour cela, j’ai créé deux fichiers php que j’ai nommé respectivement header et footer. Dedans, j’y ai placé tout le début du code html jusqu’à la fin de la balise <header> dans le fichier header et la section newsletter et footer dans le fichier footer. J’ai ensuite mis le fichier de l’accueil en php et ajouter le code suivant à la place de ce que j’ai mis dans le fichier header :**  <?php require("template/header.php"); ?>  **J’ai fait de même pour le footer en changeant le nom du fichier.** | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| **J’ai utilisé les langages suivants :**   * **Html** * **CSS** * **PHP**   **J’ai également utilisé le framework bootstrap, le logiciel IntellJ et WAMP pour tester le site en local.**  **J’ai aussi utilisé la documentation en ligne de bootstrap.** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| **J’ai collaboré avec un collègue qui s’occupait de la partie back-end, du graphisme et du retour client.** | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***D2COM*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **stage** | | | | | |
| **Période d’exercice** | |  | **Du** | | **01/11/2021** | | **au** | 26/11/21 |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| * **J’ai pu réaliser un site en toute autonomie** * **J’ai pu pratiquer un peu de php** * **J’ai pu utiliser le CSS pour faire du responsive et du flexbox** | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **1** | | | **Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°3***  | | | | ***J’ai créé une application météo*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| **Dans le cadre d’un exercice, j’ai réalisé une activité météo qui récupére une API pour ensuite l’afficher.**  **Dans un premier temps, j’ai réalisé un model en fonction du Json envoyé par l’API. J’ai d’abord mis en variable les différents attributs. Je les ai mis en private afin qu’il soit sécurisé et que seul la classe peut les utiliser directement. J’implémente ensuite la classe de l’interface serializale. Cela permet de rendre un objet en séquence. Ensuite, je mets en place un constructeur vide et un autre personnalisé. Je créé ensuite les getter et setter des variables afin de pouvoir lire ou modifier la variable dans d’autres classes.**  public class WeatherBean implements Serializable{  private CoordBean coord;  …  @SerializedName("main")  private TempBean temperature;   public WeatherBean() {  }   public WeatherBean(String name, int visibility) {  this.name = name;  this.visibility = visibility;  }   public String getName() {  return name;  }   public void setName(String name) {  this.name = name;  }  **Je créais ensuite l’utilitaire « OkHttpUtils » qui va me permettre de faire une requête au serveur afin de récupérer le Json d’une API. Je créé donc la méthode sendGetOkHttpRequest qui prends une url en paramètre. J’utilise un Log.w afin de faire apparaître dans la console l’url. Je réalise ensuite la requête puis je vérifie que je ne reçois pas un code d’erreur de la par du serveur avec une condition qui lève une exception si le serveur revoit un code inférieur à 200 ou supérieur à 299 sinon il retourne le contenu.**  public class OkhttpUtils {  public static String sendGetOkHttpRequest(String url) throws Exception {  Log.*w*("tag", "url : " + url);  OkHttpClient client = new OkHttpClient();  Request request = new Request.Builder().url(url).build();  Response response = client.newCall(request).execute();  if (response.code() < 200 || response.code() >= 300) {  throw new Exception("Réponse du serveur incorrect : " + response.code());  } else {  return response.body().string();  }  } }  **Je créais ensuite l’utilitaire WSUtils qui va récupérer le Json d’une url particulière et le parser en objet. Après avoir créé une variable qui va être instanciée de l’url voulu. Je modifie la fin de l’url afin de pouvoir modifier la latitude et la longitude du lieux voulu. Je créais ensuite une méthode loadWeather qui retournera un objet WeatherBean et qui prend en paramètre la latitude et la longitude de l’utilisateur. Je créé la variable url qui va être instancier de la première variable créée en ajoutant la bonne latitude et longitude. Ensuite, je récupère le Json avec la méthode de « sendGetOkHttpRequest » de OkHttpUtils. Je la parse ensuite grâce à new Gson dans une variable data. Je vérifie ensuite qu’elle n’est pas vide ou null. Si elle l’est, je lève une exception sinon elle retourne la variable data.**  public class WSUtils {   private final static String *URL\_WEATHER\_WB* = "http://api.openweather..." + "&lat=%s&lon=%s";  public static WeatherBean loadWeather(double latitude, double longitude) throws Exception {  String url = String.*format*(*URL\_WEATHER\_WB*, latitude, longitude);  String json = OkhttpUtils.*sendGetOkHttpRequest*(url);  WeatherBean data = new Gson().fromJson(json, WeatherBean.class);   if (data.getWeather() == null || data.getWeather().isEmpty()) {  throw new Exception("La liste est vide");  }   return data;  }  }  **Je mets ensuite en place l’activité. Je commence par l’interface graphique (fichier xml).**    **J’utilise principalement du textView et de l’imageView. A la fin, je place une barre de progression qui n’apparaîtra que lors du chargement et un bouton de rafraîchissement des données.**  **Je vais ensuite sur la classe de l’activité que j’étends avec AppCompatActivity et j’implémente View.OnClickListener. Je mets ensuite en place le binding que j’ai d’abord configurer dans le fichier build.gradle du module. Je place sur le bouton de rafraîchissement un setOnClickListener. Je mets ensuite un place une condition qui vérifie si Intent n’est pas null alors on lance la méthode que je présenterais après.**  public class WeatherActivity extends AppCompatActivity implements View.OnClickListener {  private ActivityWeatherBinding binding;    @Override  protected void onCreate(@Nullable Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  binding = ActivityWeatherBinding.*inflate*(getLayoutInflater());  setContentView(binding.getRoot());binding.btLoad.setOnClickListener(this);   if ((WeatherBean) getIntent().getSerializableExtra("weather") != null) {  showWeatherBean((WeatherBean) getIntent().getSerializableExtra("weather"));  }  }  **Je mets ensuite en place la vérification de permission de localisation. Je mets une condition si on a la permission, on lance la méthode showWeather sinon on affiche la fenêtre de demande de permission.**  @Override  public void onClick(View view) {  if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*)  == PackageManager.*PERMISSION\_GRANTED*) {showWeather();  } else {ActivityCompat.*requestPermissions*(this,  new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*}, 0);  }   }  **Je m’occupe ensuite du résultat de la fenêtre de permission. J’utilise une condition si on a la permission on lance showWeather sinon je fais apparaître un toast.**  @Override  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] gr) {  super.onRequestPermissionsResult(requestCode, permissions, gr); *//On verifie la réponse* if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*)  == PackageManager.*PERMISSION\_GRANTED*) { *//ON a la permission* showWeather();  } else { *//On n'a pas reçu la permission* Toast.*makeText*(this, "Il faut acceptation de localisation pour utiliser Weather", Toast.*LENGTH\_SHORT*).show();  }  }  **Je créais ensuite la méthode shomProgressBar qui permet de faire apparaître ou disparaître la progressBar. J’utilise OnUiThread afin que l’élément graphique puisse être utilisé dans un Thread.**  private void showProgressBar(Boolean visible) {  runOnUiThread(new Runnable() {  @Override  public void run() {  if (visible) {  binding.pb.setVisibility(View.*VISIBLE*);  } else {  binding.pb.setVisibility(View.*GONE*);  }  }  }); }  **Ensuite je créais une méthode permettant de récupérer la dernière localisation. Je vérifie en premier temps qu’on a bien la permission de geolocaliser puis récupère la dernière localisation avec la meilleure précision qu’ensuite on retourne.**  private Location getLastKnownLocation() {if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*)  == PackageManager.*PERMISSION\_DENIED*) {  return null;  }  LocationManager lm = (LocationManager) getSystemService(*LOCATION\_SERVICE*);  Location bestLocation = null;for (String provider : lm.getProviders(true)) {  Location l = lm.getLastKnownLocation(provider);  if (l != null && (bestLocation == null || l.getAccuracy() < bestLocation.getAccuracy())) {  bestLocation = l;  }  }  return bestLocation;  }  **Je créais ensuite la méthode permettant d’afficher le message d’erreur.**  private void showErrorMessage(String errorMessage) {  runOnUiThread(new Runnable() {  @Override  public void run() {  binding.tvError.setVisibility(View.*VISIBLE*);  binding.tvError.setText(errorMessage);  }  });  } }  **Je créais après la méthode permettant de mettre à jour UI.**  private void showWeatherBean(WeatherBean data) {  String url = "https://openweathermap.org/img/wn/" + data.getWeather().get(0).getIcon() + "@4x.png";  runOnUiThread(new Runnable() {  @Override  public void run() {   binding.tvName.setText(data.getName());  Picasso.*get*().load(url).into(binding.ivTemperature);  binding.tvTemps.setText(data.getWeather().get(0).getDescription());  binding.tvTemp.setText(data.getTemperature().getTemp() + "°");  binding.tvTemp2.setText(getString(R.string.*string\_minmax*, "" + data.getTemperature().getTemp\_min(), "" + data.getTemperature().getTemp\_max()));  binding.tvWind.setText("" + data.getWind().getSpeed());  }  }); }  **Je peux donc maintenant créer showWeather. Je mets d’abord en place un thread puis un try et catch afin de pouvoir lever des exceptions en cas d’erreur.**  public void showWeather() {  showProgressBar(true);   new Thread() {  @Override  public void run() {  final WeatherBean weather;  try {  Location location = getLastKnownLocation();  if (location == null) {  throw new Exception("Pas de localisation");  }  showWeatherBean(WSUtils.*loadWeather*(location.getLatitude(), location.getLongitude()));    } catch (Exception e) {  e.printStackTrace();  showErrorMessage(e.getMessage());  }  showProgressBar(false);  }  }.start();  binding.tvError.setVisibility(View.*GONE*); } | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| **J’ai utilisé le langage Java et le logiciel IntellJ.**  **J’ai également utilisé la documentation en ligne de Java** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| **TP individuel** | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | **Cliquez ici pour taper du texte.** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | Cliquez ici pour taper du texte. | | | | | |
| **Période d’exercice** | |  | **Du** | | **31/08/2021** | | **au** | **09/09/2021** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| * **Ce tp m’a permis de m’exercer en Java** * **Il m’a permis d’utiliser une API et le Json** * **Il m’a permis d’utiliser les exceptions** | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **2** | | | **Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°1***  | | | | ***J’ai mis en place un Webservice*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| **Afin de tester nos connaissances globales, on nous a donné un exercice en groupe qui consistait à créer une application android récupérant deux API. Afin de pouvoir utiliser les APIssans restriction, nous devions mettre en place un webservice qui récupérer l’API donnant les stations de vélo sur Toulouse, la sauvegarder dans une base de données et ensuite la renvoyer.**  **Dans un premier temps, j’ai analysé le Json que l’on nous a fourni pour élaborer le MCD et le MLD.**    **J’ai donc créé une entité nommée STATIONS avec une propriété « id » en clé primaire. La clé primaire est unique à chaque enregistrement et on peut voir qu’elle est soulignée dans le MCD pour l’identifier. J’ai ensuite ajouté 7 autres propriétés avec leur typage.**  **N’ayant qu’une seule entité, nous n’avons donc aucune relation ni cardinalité. Le MLD est alors exactement identique au MCD.**  **Après la conception, j’ai donc créé la base de données et sa table avec ses attributs.**  **J’ai d’abord créé la base de données avec « CREATE DATABASE » et son nom. Ensuite, j’indique qu’on va l’utiliser avec « use » puis je peux enfin créer la table avec « CREATE TABLE » et le nom de la table ainsi que ses attributs. Pour l’attribut « id » qui est notre clé primaire, on lui instancie le « PRIMARY KEY » ainsi que le « BIGINT » son typage, « NOT NULL » car il ne peut pas être vide et « auto-increment » pour que l’id soit créé automatiquement à chaque enregistrement en ajoutant 1 à l’id précédent.**  **Une fois la base de données crée, je commence à mettre en place le serveur. Je créais le projet avec spring boot puis crée la classe Webservice qui est un controler. Je lie la base de données avec le projet et inscris les informations d’accès dans le fichier application.properties. Je créais ensuite les classes model. Je créais donc la classe StationsBean.**  @Entity @Table(name = "stations") public class StationsBean {  @Id  @GeneratedValue(strategy = GenerationType.*IDENTITY*)  private long id;  private String name;  ...  }  **J’utilise le @Table pour indiquer qu’elle table de la base de données elle doit utiliser. Les attributs sont mis en private pour éviter qu’elles soient modifiables directement. C’est pour cela que dans la suite du code, j’ai créé les getter et setteur lié à chaque attribut pour qu’on puisse les lire et les instancié. L’id est en génération automatique.**  **Ensuite, je créais l’interface StationsDao pour lire et modifier la base de données.**  @Repository public interface StationsDAO extends JpaRepository<StationsBean, Long>{ }  **Ici, je le laisse vide car je n’ai pas besoin de requête spécifique. Je n’utiliserais que les requêtes déjà intégrées. Je créais ensuite la classe utilitaire OkHttpUtils qui aura la méthode static getstation qui permettra de récupérer le gson de l’api et de le parser pour retourner une liste de StationBean.**  public class OkHttpUtils {  public static List<StationsBean> getStation() throws Exception {String url = "https://api.jcdecaux.com/vls/v1/stations?contract=Toulouse&apiKey=2a1b07b2a523f81188fe34e348206a57ffa6f2a7";  System.*out*.println("Url : " + url);  OkHttpClient client = new OkHttpClient();Request request = new Request.Builder().url(url).build();Response response = client.newCall(request).execute();if (response.code() < 200 || response.code() > 299) {  throw new Exception("Réponse du serveur incorrect : " + response.code());  } else {  Gson g = new Gson();  List<StationsBean> maListeStations = g.fromJson(response.body().string(), new TypeToken<List<StationsBean>>() {  }.getType());return maListeStations;  }  }  }  **Je mets l’url en variable puis fais un contrôle avec « system.out.println » pour l’afficher dans la console. Je créais alors la requête puis l’execute et enfin je fais analyser la réponse avec un « if » qui retourne une exception si le code et inférieur à 200 ou supérieur à 299 sinon il parse le gson et retourne la liste.**  **Après quelques tests pour vérifier que la base de données est bien reliée et que la méthode getStations fonctionne, je créais la méthode permettant de remplir la table de la base de données avec l’API. Comme je veux que la base de données soit mise à jour toutes les 10 minutes, je créais la classe ScheduleTasks.**  @Component public class ScheduledTasks {  @Autowired  private StationsDAO stationDao;   @Scheduled(fixedRate = 600000)  public void refreshData() throws Exception {  System.*out*.println("refreshData()");  List<StationsBean> listeStationBDD = OkHttpUtils.*getStation*();   if (!listeStationBDD.isEmpty()){  stationDao.deleteAll();  stationDao.saveAll(listeStationBDD);  }  else {  throw new Exception("Liste vide BDD non mise à jour");  }  } }  **Je créais la méthode refreshData dont le temps d’exécution est fixé à 600 000 millisecondes, soit 10 minutes. Je mets le contrôle console pour voir quand il s’exécute avec le system.out.println. J’appelle ensuite la methode getStation dans la variable listeStationBDD. Je teste ensuite si la liste n’est pas vide alors j’efface tout dans la base de données avec la méthode deleteAll() de l’interface stationDao puis on enregistre la nouvelle liste dans la base de donnée avec la méthode saveall() de l’interface stationDao. J’ajoute ensuite @EnableScheduling dans la classe application où se trouve le « main ».**  **Enfin, je créais la méthode getBikesInToulouse dans le controler Webservice. Je mets un contrôle console puis retourne tous les enregistrements de la base de donnée en format gson grâce à la méthode intégrer findAll() de l’interface StationDao.**  @GetMapping("/getBikesInToulouse") public List<StationsBean> getBikesInToulouse() {  System.*out*.println("/getBikesInToulouse");  return stationDao.findAll(); } | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| **Pour réaliser ses tâches, j’ai utilisé les langages JAVA et MySQL.**  **J’ai donc utilisé les logiciels suivants :**   * **Looping pour le MCD/MLD** * **MySQL Workbench et WAMP pour la Base de données** * **IntelliJ pour coder le webservice** * **Spring boot pour la mise en service du webservice**   **J’ai également utilisé la documentation en ligne de SQL et de JAVA.** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| **Nous avons réalisé ce TP en groupe de 4 mais nous nous sommes organisés afin que 2 d’entre nous s’occuper de la partie serveur tandis que les deux autres créer une application utilisant ce webservice.** | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***ADRAR*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **TP** | | | | | |
| **Période d’exercice** | |  | **Du** | | **13/12/2021** | | **au** | **17/12/2021** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| * **Ce TP m’a permis de mettre en pratique tous ce que j’ai appris le long de la formation.** * **J’ai également dû travailler en équipe. On n’a dû s’organiser et communiquer tout du long.** * **J’ai dû réaliser des tests à chaque étape et gérer les erreurs rencontrer.** | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** |  | | |  | | | | |
| **Activité-type** | **2** | | | **Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité** | | | | |
| ***Exemple n°2***  | | | | ***J’ai réalisé une base de données avec ses requêtes*** | | | | |
|  | | | |  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | | | | |
| **Pour réaliser cette base de données, j’ai d’abord réalisé un Modèle Conceptuelle de Données (MCD).**  **Cette base de données à 3 entités avec une clé primaire « id » qui est souligné sur le schéma pour l’indiquer et des propriétés de type varchar. Je créais une relation entre les entités responsable et locaux de type 1,1. Elles sont associées par le verbe « avoir » avec de chaque côté une cardinalité 0,1. Je le traduirais donc par les phrases :**  **« un responsable peut avoir aucun ou un local » et « un local peut avoir aucun ou un responsable ».**  **Je créé également une relation de type n, n entre les entités locaux et voitures. Elles sont associés par le verbes stationner et leurs cardinalités de type 0,n. Je les traduirais par :**  **« un local peut stationner aucune ou plusieurs voitures » et « une voiture peut être stationner sur aucun ou à plusieurs locaux. »**  **Ensuite, j’ai réalisé le Modèle Logique des Données. Il reprend en partie le MCD. Pour la relation de type « 1,1 », j’ai donc ajouté une clé étrangère « id\_responsonble » (la clé primaire de l’entité « responsable ») à la table « locaux ».**  **Pour la relation de type « n,n » , j’ai créé une table d’association nommée « stationner » où j’ai ajouté en clé primaire et clé etrangère les deux clés primaires des tables liées.**  **Après j’ai donc créé la base de données avec « create database » puis on indique qu’on va l’utilisée avec « use ». Ensuite, je peux donc créer les tables avec « create table ».**  create table responsables (  id\_responsable bigint not null primary key auto\_increment,  nom varchar(50),  …  id\_local bigint,  foreign key (id\_local) references locaux(id\_local)  );  **J’ai indiqué que l’id\_responsable est une clé primaire avec « primary key », qu’elle doit avoir une donnée avec « not null » et que la donnée est automatique créé en ajoutant au précédent avec « auto-increment ». Je créais ensuite la clé étrangère qu’on relie avec l’autre table avec « foreign key() references ».**  **Je créais enfin la table d’association « stationner », je mets les deux champs en clé étrangère que l’exemple d’avant.**  create table stationner (  id\_local bigint not null ,  id\_voiture bigint not null,  foreign key (id\_local) references locaux(id\_local),  foreign key (id\_voiture) references voitures(id\_voiture)  );  **Après avoir créé la table, je me rends compte que j’ai oublié de mettre mes deux champs en clé primaire. Je vais donc faire une requête pour la modifier avec « alter table » puis j’utilise « add primary key() » pour ajouter mes champs en clé primaire.**  alter table stationner  add primary key (id\_local, id\_voiture);  **Ensuite je réalise quelques enregistrements pour pouvoir faire des tests sur la base de données avec « insert into ».**  insert into locaux (adresse,ville,cp) values  …  ("7 rue Final","Fantasy",77777);  **Je fais également des enregistrements sur les autres tables. Après je modifie un des enregistrements grâce à « update » puis je précise le champ à modifier avec « set » et précise quel enregistrement il s’agit avec « where ».**  update voitures  set couleur ="marron clair"  where id\_voiture = "3";  **Ensuite je réalise quelques suppressions avec « delete » et « where ».**  delete from voitures  where couleur ="marron clair";  **Après j’ai réalisé une requête de selection où je sélectionne tous les champs de la table voiture**  select \*  from voitures;  **Je souhaite avoir une information plus précise. Je voudrais par exemple avoir le nom des responsables avec la ville de leur local, je ferais donc cette requête :**  select nom, ville  from responsables  inner join locaux  where responsables.id\_local=locaux.id\_local; | | | | | | | | |
|  | | | | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | | | | |
| **J’ai utilisé MySQL comme langage.**  **Pour les logiciels, j’ai utilisé Looping pour les diagrammes MCD/MLD, MySQL Workbrench et WAMP pour la création et l’hébergement de la base de données** | | | | | | | | |
|  | | | | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | | | | |
| **J’ai réalisé ce TP seule.** | | | | | | | | |
|  | | | | | | | | |
| **4. Contexte** | | | | | | | | |
|  | | | | | |  | | |
| **Nom de l’entreprise, organisme ou association**  | | | | | | ***ADRAR*** | | |
|  | |  | | | | | | |
| **Chantier, atelier, service** | |  | **TP** | | | | | |
| **Période d’exercice** | |  | **Du** | | **09/06/2021** | | **au** | **09/06/2021** |
|  | | | | | | | | |
|  | | | | | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | | | | | |
| * **Ce TP m’a permis de faire des requêtes sql varié** * **J’ai également réalisé un MCD/MLD ayant deux sortes de relations** | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **6** |  |  |

|  |  |  |
| --- | --- | --- |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| 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. |

|  |
| --- |
| **Déclaration sur l’honneur** |
|  |
|  |

*Jessica Anglade*

Je soussigné(e) [prénom et nom] ,

déclare sur l’honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l’auteur(e) des réalisations jointes.

*Pommevic*

07/01/2022

Fait à le

pour faire valoir ce que de droit.

Signature :

|  |
| --- |
| **Documents illustrant la pratique professionnelle** |
|  |
| *(facultatif)* |
| **Intitulé** |
| Cliquez ici pour taper du texte. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **Annexes** |
|  |
| *(Si le RC le prévoit)* |
|  |