|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Nom de naissance* |  | ABDOUN |
| *Nom d’usage* |  | ABDOUN |
| *Prénom* |  | Walid |
| *Adresse* |  | 708 boulevard République François Mitterand  59240 Dunkerque |
|  |  |

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

|  |  |
| --- | --- |
| **Présentation du dossier** | |
|  | |
|  | |
| Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l’emploi.**  Le DP appartient au candidat. Il le conserve, l’actualise durant son parcours et le présente **obligatoirement à chaque session d’examen**.  Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.  Il est consulté par le jury au moment de la session d’examen.  **Pour prendre sa décision, le jury dispose :**   1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l’entretien professionnel ou de l’entretien technique ou du questionnement à partir de productions. 2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle. 3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d’un parcours de formation 4. de l’entretien final (dans le cadre de la session titre).   *[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels*  *du ministère chargé de l’Emploi]*  **Ce dossier comporte :**   * pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ; * un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d’un titre, d’un diplôme, d’un certificat de qualification professionnelle (CQP) ou des attestations de formation ; * une déclaration sur l’honneur à compléter et à signer ; * des documents illustrant la pratique professionnelle du candidat (facultatif) * des annexes, si nécessaire. | |
| *Pour compléter ce dossier, le candidat dispose d’un site web en accès libre sur le site.* | |
|  | [**http://travail-emploi.gouv.fr/titres-professionnels**](http://travail-emploi.gouv.fr/titres-professionnels) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sommaire** | | | |
| **Exemples de pratique professionnelle** | | |
| **Développer la partie front-end d’une application web en intégrant les recommandations de sécurité.** | | **p.** | **5** |
|  |  Maquettage  p. | p. | 5 |
|  | JAVASCRIPT p. | p. | 7 |
|  |  HTML/CSS/BOOTSTRAP p | p. | 10 |
|  |  |  |  |
| **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | **p.** | **13** |
|  |  Projet Walidify  p. | p. | 13 |
|  | MCD p. | p. | 28 |
|  |  MVC/CRUD p | p. | 50 |
|  |  Connexion/Inscription  p | p. | 75 |
|  |  Commentaire p | p. | 86 |
|  |  |  |  |
| **Titres, diplômes, CQP, attestations de formation** *(facultatif)* | | **p.** | 93 |
| **Déclaration sur l’honneur** | | **p.** | 94 |
| **Documents illustrant la pratique professionnelle** *(facultatif)* | | **p.** | 95 |
| **Annexes** *(Si le RC le prévoit)* | | **p.** | 96 |

**Exemples de pratique**

**professionnelle**

|  |  |  |
| --- | --- | --- |
| **Activité-type 1**  **Développer la partie front-end d’une application web en intégrant les recommandations de sécurité.** | | |
| ***Exemple n° 1*** ***Maquettage*** | | |
|  | | | | |
|  | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | |
| J ’ai maquetté une page sur le thème de l’Irlande » dans 3 formats :  PC/Tablette/Télephone | | | | |
|  | | | | |
|  | | | | |
|  | | | |
|  | | | | |
| **2. Précisez les moyens utilisés :** | | | | |
|  | | | | |
| J’ai utilisé FIGMA pour la maquette afin d’agencer l’emplacement des éléments selon le support. | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | |
|  | | | | |
| J’ai effectué ce travail seul. | | | | |
|  | | | | |
|  | | | |
|  | | | | |
| **4. Contexte** | | | | |
|  | |
| Nom de l’entreprise, organisme ou association | |
| *AFCI Saint Pol Sur Mer*  Chantier, atelier, serviceCentre de formation |
| Période d’exerciceDu : *10/10/2022*au :*12/10/2022* | | | | |
|  | | | |
|  | | | |
| **5. Informations complémentaires** *(facultatif)* | | | |
|  | | | | |
|  | | | | |
|  | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Activité-type 1**  **Développer la partie front-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 2*** ***Javascript*** | | | |
|  | | | |
|  | | | |
| J’ai développé un quiz en HTML et Javascript        C:\Users\Mon PC\Downloads\Desktop - 1 (4).png | | | |
|  | | | |
|  | | | |
|  | | | |
| **3. Avec qui avez-vous travaillé ?** | | | |
|  | | | |
| J’ai effectué ce travail seul. | | | |
|  | | | |
|  | | |
|  | | | |
| **4. Contexte** | | | |
|  | |
| Nom de l’entreprise, organisme ou association*AFCI Saint Pol Sur Mer* | |
| Chantier, atelier, service  *Centre de*  *formation* |
| Période d’exerciceDu : *13/02/2023*au :*18/02/2023* | | | |
|  | | |
|  | | |
| **5. Informations complémentaires** *(facultatif)* | | |
|  | | | |
|  | | | |
|  | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **1** | **Développer la partie front-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 3***  | | ***Projet WALIDIFY*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| J’ai développé une page d’accueil en prévision de mon projet de site.    CSS :  https://lh6.googleusercontent.com/kNxCbXcx-yl86QcE32oNee4HyvbM17UG3gnjLJTCw3rvjGPb7jeFAeuoTROCDMngMjiHdBwFjyANHMpsHctSQmNKonu5oGCjofQq4H7Oa-5S6j81hwXCgRbCWdDbGpFSjVtDKgC6ZpZhVSjnOZOuz5A  HTML :  https://lh6.googleusercontent.com/PXqbPTnpd-z6CA1qnCrQdwaM75dxQfl3CHoQR7eb9J3CM5TJ6alhzBwrM0hGT_83BHriejQMiIuY-rL-uDsAo8enSH_1Ru9kpj8NmpyErOkRefOmVYu_AQMdDXcXpNuwrZ_GuG4DqTTo7o0LVTdI1J4 | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| HTML/CSS/BOOTSTRAP. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| J’ai travaillé seul. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI Saint Pol Sur Mer* | |
| Chantier, atelier, service | | | *Centre deformation* | | |
| Période d’exerciceDu : *20/02/2023*au :*24/02/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 1***  | | ***Projet WALIDIFY*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| En premier lieu, j’ai commencé à maquetté l’interface du site.  La V1 de la maquette a pour objectif d’afficher chaque page artistes/albums/playlists de cette manière.  https://lh4.googleusercontent.com/lbuvrtKVvVxrpIn7HDr6JlQvUYOGNRVKKKuyDFoko_3TfwwCL_EmC0raaeDNTFT9EF_2dDKb0rOXgWhsaq4Y1AeoB6zNznbqwihWkQlp9vxxiV--bM7MzkQc7lU9mwcTQNvXtSpr7dgUjC7C5YZb4ic  Affichage d’un album :  https://lh3.googleusercontent.com/6buLn5jVPNHQum9XVYWBVwgW-62ftGDkDjLQIMyl9YsrrSmD75B9XYN0ViGDyzxMgi1iB2I4-Ptcm9KgolrTafQOQvAgdxQEK7xzwnXdwykcDdwKoiz_HRnd05oHiAEGlLZI01O32lkbXXs3DTTvTDw  Page de connexion et d’inscription :  https://lh4.googleusercontent.com/hkaUbOsnoc51REFz9aChncvIgZ3Xih427xeDfvnaUhX-c4PaJ20Q2qlPU81OECLxaaemQL7TvoNeVsrU8IuCG0Qdw-0jWYhQE6N_Tp_wucfklvcC9Uql7y5GivGRKAYuSh9x95stxuYeG6JfS5ISD0M  Page Admin :  https://lh5.googleusercontent.com/2BfS6VoGdxBAWMcj7vxVKobvqatDHHc4kyBGRNEzKHsIHDnKIxNCdVFauc-Hg9PYOHqK4HKfsz3eKpWQ3AwEbX9xzUZDa8GzcCdnHSn9_oDaLP5Us08Cii37o18tociVliHIp1c6FOgHb9D8uwKjgcY  En me basant sur les conceptions de ces maquettes j’ai donc édifié une base de développement pour l’aspect HTML/CSS et BOOTSTRAP.  Création du dossier Walidify dans le dossier WWW de wamp64 et ouverture de celui-ci dans VSC.  https://lh6.googleusercontent.com/DE0XF5YVop4dK-niaQOszikY3L7jT0WJO3IqhsAwb-10l9yABoz4e9qW5UsU4sX74QtMfLe0MJGfRKa7Cb03uZQonKKcJBIBMEnxddJCEfhnrzAhxAAqgPCr5k42FBpAB7BpZ2Md6sLXyba3TKz9DF4https://lh4.googleusercontent.com/Te4OilwZGh133Ll6KXxD4sERItKEcZbGU8G_Pg61QIWfyoNZ5cSoRsO1FnHA4f9Q0iuHyIUs79Y5YePj5fm9x_sqHPOvGYbg5SU7gyQ_arLScdxEk1dT8gEN6AQuJy3z-aX1y4E1bOEJKNatjVEi8MM  Création du site sur Visual Studio Code :  HTML :  https://lh4.googleusercontent.com/KL5Cww6aDXZFuP80EA68cgcOpZWGxOlx5r-waoLH8RQRe4kMV54Tht6A5mY060cyzL_V7BFS5Ijv_LUM1sbQVnDFQyHmt1saaf_YR5aLuqCoV2Vq4FOy3TNyTixvJ-vHT2Phcyil3SmBKPbxREcBfFc  https://lh4.googleusercontent.com/TTCgLl7yv_LN9eDu2ocEaqSZLYlbgrb5vR-Yi6O74COGLdMVtlykEFKJaMnFx_EiDy-OlWakvUGTSiL3xTrBDZMwL5w59Q0ypAc_rWae6Iu7xUTaG8BW48xIPEyuy_eGP4AkPG7TQdGbJQQDS42KeP0  CSS  https://lh4.googleusercontent.com/sXVLVjHoUxvjfbd6tPoQnQa_8EDgN3gBVFjBWyBIKdLoyumHNPTSV2Bn8laPWPctkVezZqNaG1KXqpI-BJcbQSmDLFOKRkUeTf8Kxa5vG3RMpQdZQaMoF6iP-2PcNB1vggOwhz6JnuRbxBCKx8u2CZ4  CSS pour la vidéo en fond :  https://lh6.googleusercontent.com/kNxCbXcx-yl86QcE32oNee4HyvbM17UG3gnjLJTCw3rvjGPb7jeFAeuoTROCDMngMjiHdBwFjyANHMpsHctSQmNKonu5oGCjofQq4H7Oa-5S6j81hwXCgRbCWdDbGpFSjVtDKgC6ZpZhVSjnOZOuz5A  JavaScript pour le message :    Page d’affichage des albums :  https://lh5.googleusercontent.com/Nc7nLzpJ-nTXH9_aAGe_Mhe0PZ0eY0wLSEKNHq9Ctv9culIepdZ51W2grK8cbDLhXvizVib0mPoKvHz_VD-YpKM-hskWtCBEe5Ja6J8dWrv3Dx_shcOZlSMXXCLHtJPHjdZIQvsV-0lG_1yK7u4Ep8E  CSS :  https://lh4.googleusercontent.com/JH83CR3b0hYbHuS2Sv6nQ6ctfS_TX0UJGHN91h1ju52srywBZZrp_dU8nFIA0TEhtV3OjePE4Elm5q_iD3JtPYA3UfZoD54qUu1XEXaN2BvTU4FofW5jkF21o9HF1WYVFmHUP5v7z_1o7Gsz88blNLY    Page d’affichage d’un album :  https://lh3.googleusercontent.com/5921CT5Qijlf9gMuTIiFOzC_H3EeQzvhWJwL4p50kNVI3po4-OBjB0lcCTEqCJ0iUPjwqkc1t72wVDRq1gizgBYRODyfd9ttVpa1_1ZmaimIO4nJ7I4JHlO0O9L-bt_becrBFJKUpQfEJ_Ik3Zp6OVQ    HTML :  https://lh6.googleusercontent.com/5UmFhTZNYDn_bcuKj9oO5qoGrre42-fC8WvTXPG4gtK9nugSqDeTqHxrmsVPA8v8BdnKcRpjD3fyo-CEpk1w6VzwyJ122VDw-U-XOhVuQ2XFRaoPsZRIytuf8iwv4VA9JuKjZtLbIcNL0aM9d3JooK8  Page de connexion/inscription :  https://lh3.googleusercontent.com/PbQRgTtq_ftXC1jx7CtmvvwDAgMqb-ECicGbnsH1tKwyPnWaHn_mDTYXT6dx195qBaaY4HMcQu9kKxSRdap5zZxhI-prUVhHM6UyJFyEfCExWCWtAuKoycH4G1J2-0hJESOGAFe5d-KBO-PF4Aboo6k  https://lh5.googleusercontent.com/NERV5FR8lM3eOVhcvYy5EE2Wgacy9uzCzAcqaxkhyzWP7Sb7whBMiIkINiNVkMTHGs0eEkHQ642p3kG1cYhJ5PD0csZbdL05-SGOTXMYPp6u4JXPBv0NAq8C6hOcqKkt5PqkDMCepotVJFgZk6FJdgk    HTML :    https://lh3.googleusercontent.com/Ur9SpcIHXwb9cuQIMwrRa9DAuqxreeQaf9LRb1pyPuXyld3S8C6ibG-N-5TMQyOeVGiOsio9ASE6f2Z6mYtfpHfc-l-zjrksipNVvdGnW8Bq5LVWEQ3qjCu0JYNNqgQnSyOIq59oPBjYz8WD-rRIOQM  CSS :  https://lh5.googleusercontent.com/zzW2uT_PDXck-5RQ9M3XGvWoNwRPo04vphi4sV5SH-LB8psUNLpuvzV5a5UWYX7o0EvK_qmGv-fnpL8-h3x_ie1Dh_gE9hqXoEBCShZYQPZPTT3WJhOETc3V-qDRT2pWo6cBVjxM5LMJRH7CaQSCbZU  Page des actualités :  https://lh6.googleusercontent.com/Xuia0ZVf-5n7w0HvewG4AIYP1l4vBAB4nWcddTFgX57GnXfbGzuKfbeURqGZC7ADVbnWicHpDz-XYzUxKoI66aarSel64U63-f1qcVoDji-ICpiDk2vIMSw8mSgjZGtcFW1WcYavNIs-tD1s13Y-RJc    HTML :  https://lh4.googleusercontent.com/sYtB_9d2B72Z0X1YcsfJEGVrsU27Z0OaffXFpnYBuOHhe05ORJHIwDkPO71w8AtnIbckoHdsej_a7Ea95qS4DFCzMu8oHS9pMxzGRB6xNZ8wv1PpnGgDdGxwMS5gzIU79zOt0Ldsx1Gy5YWR_u-5C44 | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé Figma pour le maquettage.  Par la suite, HTML, CSS, BOOTSTRAP ainsi que JavaScript | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI* | |
| Chantier, atelier, service | | | Atelier | | |
| Période d’exerciceDu : *06/03/2023*au :*02/06/2023* | | | | | |
|  | | | | |
|  | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 2***  | | ***MCD*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
| **J’ai effectué le Modèle Conceptueur de Données pour mon site de cette manière.** | | | | | |
| C:\Users\Mon PC\Downloads\mcd.png  C:\Users\Mon PC\Downloads\mcd2.png  C:\Users\Mon PC\Downloads\mcd3.png | | | | | |
| Et voici le script SQL que j’intègre dans ma base de données:  C:\Users\Mon PC\Downloads\Frame 1 (1).png | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| J’ai utilisé AnalyseSI pour la conception du MCD, Visual Studio Code et MySQL pour la lecture et l’intéfration du script SQL. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI* | |
| Chantier, atelier, service | | | Atelier | | |
| Période d’exerciceDu : *06/03/2023*au :*02/06/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 3***  | | ***MVC/CRUD*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Définition de la structure MVC  1. Modèle  Dans la structure MVC, le Modèle est responsable de l'interaction avec les données. En utilisant PHP et une base de données (en l'occurrence ici MySQL via wampServer), le modèle traite toutes les demandes d'information, mises à jour et modifications.    public function chargementPlaylist()      {          $req = $this->getBdd()->prepare("SELECT \* FROM playlists ORDER BY id ASC");          $req->execute();          $mesplaylist = $req->fetchAll(PDO::FETCH\_ASSOC);          foreach ($mesplaylist as $playlist) {              $play = new Playlist(                  $playlist["id"],                  $playlist["playlist"],                  $playlist["link"],                  $playlist["images"],              );              $this->ajoutPlaylist($play);          }          $req->closeCursor();      }    2. Vue  La Vue est responsable de l'affichage de l'information à l'utilisateur. C'est la partie de votre application qui contient les fichiers HTML, CSS et une partie de JS.  3. Contrôleur  Le Contrôleur fait le lien entre le Modèle et la Vue. Il traite les requêtes de l'utilisateur, interagit avec le Modèle pour récupérer les données, puis transmet ces données à la Vue pour être affichées.  class PlaylistsController{      private $playlistManager ;      public function \_\_construct(){      $this->playlistManager = new PlaylistManager() ;      $this->playlistManager->chargementPlaylist();      }  // Fonction afin d'afficher la liste des playlists dans la page playlists      public function afficherPlaylist(){          $playlists = $this->playlistManager->getPlaylists() ;          require "views/playlist.view.php" ;      }    J'ai conçu une base de données relationnelle qui stocke les informations des artistes, albums, playlists, les identifiants des utilisateurs et les commentaires. J'ai utilisé PHP pour interagir avec cette base de données, en créant des requêtes SQL pour insérer, récupérer, mettre à jour ou supprimer des données.  Voici un extrait de mon modèle d'utilisateur, qui permet de récupérer une playlist par son identifiant :  Création de la base de données Walidify et des tables dans la base de données :  https://lh3.googleusercontent.com/MUyjPwaUSAlXvsW7zcYoEI9oopMhwUZO_OXbGcYKd7CE8n1zvlnaP5Pn8mwbMGj4TtKblHIDvgu4dHrU-8Qr1PffHh9EmUFRM2HDhSnsZUIK6k-Z-PyGpWsG8mzXNNAbtvQQF2vlpteFofE9vYpHQTg  Insertion des tables dans la BDD Walidify :  https://lh3.googleusercontent.com/2tZC5IBYAfrEG67qVTWt05JIqRZxURgBYyYMRdtQVZPVR6PHYmQT8czg9PAZhkrdhUnA4NoU2JhJXd2xl712zE66clF76coN1TEj7WCv2SR9jG-G9tbPhGyAv1BPyd7WsIe7NeOFYXjIZBp7wA5esjw      Chaque table possède plusieurs colonnes pouvant contenir des informations, par exemple la table album :  https://lh6.googleusercontent.com/F4NxjxvYBGSQ74RRCkRY1AtssuDiMZCLsZdari9Ejq-lfztT-ki3g7qkK8i_3eRapFLWT34u3dXVgzmzzI0QNJ4yJT9zY1TKRDfbMNJOVY7XlAFoKkVNJQW9QZnM-2tIdm9DIqia1qNRzIIX8XmQUaw    L’id :  “PRIMARY KEY”, autrement la clé primaire, permet d'identifier chaque enregistrement. Il est auto-incrémenté à chaque enregistrement. Peut être utilisée en tant que clé secondaire.  Nom/album/tracks/durée/date/statut :  contiennent des informations à tout moment modifiables et sont relatifs à chaque artiste ou album.  Visualisation finale de la table après l’ajout manuel des informations de l’album :  https://lh4.googleusercontent.com/EX9FNrcPZuagMle7GKZWpUFfKwa9JUK-lWGj7G_WRqyDRe7uJJSKj8eaF9HaY8felUKptU50Kew7ua4SZDu0koGXPWhWSvPEMFfsrJxLa_qhv22otpTZU71MyBMHefteTzKyBg4jslFIUL2qcp8dsDQ  6. Connexion à la BDD :  Création de la classe Model permettant de créer la connexion à la BDD par l’intermédiaire de PDO.  La classe est abstraite car elle ne sera jamais instanciée directement.  https://lh5.googleusercontent.com/xE_kN906_Z2pDHVjybn0b6SQevduUtcirvGmfmO1wJJGcUCuQDk8-yhatvHdpvN-4hDICZVzVrYvSRuhBqJjGobnrd_kajYHOurHeKxyKnZNopLqp044G7R1pYpNHsxFN8W7NqFPcUioSGsfD7zipCE  // Class Model va permettre de gérer  la connexion à la BDD  abstract class Model{  // Définit l'attribut en static afin qu'il soit accessible par toutes les class qui héritent de la class Model      private static $pdo;      private static function setBdd(){  // Définit le chemin afin d'accéder à la BDD          self::$pdo = new PDO("mysql:host=localhost;dbname=walidify;charset=utf8", "root", "");          self::$pdo->setAttribute(PDO::ATTR\_ERRMODE,PDO::ERRMODE\_WARNING);      }  // Protected = accessible par les class filles mais pas par des algorithmes tiers      protected function getBdd(){  // Dans cette fonction, je test si j'ai déjà une instance de PDO qui existe/ si mon attribut static PDO est vide ou non  // Si c'est null, il faut que je génére une 1ère connexion          if(self::$pdo === null){              self::setBdd();          }          return self::$pdo ;      }}    Ce code définit une classe abstraite PHP appelée Model. Cette classe est destinée à gérer une connexion à une base de données en utilisant PDO (PHP Data Objects), qui est un ensemble d'interfaces de bases de données en PHP.  Routeur :  “index.php” doit router les demandes des utilisateurs vers les pages correspondantes. Pour cela, le fichier va regarder les urls et choisir le bon canal pour réaliser les actions.  Exemple :  <?php  session\_start();  define("URL", str\_replace("index.php", "", (isset($\_SERVER['HTTPS']) ? "https" : "http") .      "://$\_SERVER[HTTP\_HOST]$\_SERVER[PHP\_SELF]"));  try {      if (empty($\_GET['page'])) {          require "views/accueil.view.php";      } else {          // Découpage de l'URL pour accéder aux routes          $url = explode("/", filter\_var($\_GET['page']), FILTER\_SANITIZE\_URL);          switch ($url[0]) {              case "accueil":                  require "views/accueil.view.php";                  break;                  case "news":                      require "views/news.view.php";                      break;  En résumé, ce code permet de charger différentes pages du site en fonction de l'URL demandée par l'utilisateur.  Fichier .htaccess :  Permet de réécrire dynamiquement l’url en transformant l’affichage php pour devenir un affichage standard :  https://lh6.googleusercontent.com/uDmJiwwPvQUGHxUmVBaORvuFS3kOSjOWrFV0SMLx_nE_17u3GkXF_b6XKgB4PF1ulsby2mru-LYZ18X1thZDQe23iUrsehtXi21or8h02wVTkTgtkolZGdIbxULYbzDhKGotr_upSfIQpdOiymBx894  Mise en place du MVC pour la page Albums :   1. La class Albums en POO situé dans Models :   https://lh4.googleusercontent.com/iUDUenHVy792PtW-tDvvXzQTpkP0iKZ3WVZId9IP3qtHfc3MTpDmYUihPr9GmIUFm7NCXZpRw_eZQ4b2IzP2wgyPTZv4cWf7eyt5Z_CAt5uhdSSE38LtS_CNn1llafW82GwKlBF6pETClbUMzol4XUQ    class Album {      private $id;      private $nom;      private $album;      private $statut;      private $duree;      private $tracks;      private $dateSortie ;      private $image ;        public function \_\_construct($id , $nom, $album , $statut , $duree, $tracks, $dateSortie, $image){          $this->id = $id;          $this->nom = $nom;          $this->album = $album;          $this->statut = $statut;          $this->duree = $duree;          $this->tracks = $tracks;          $this->dateSortie = $dateSortie;          $this->image = $image;      }  // Getter & Setter à utiliser dans la page Albums      public function getId(){return $this->id;}      public function getNom(){return $this->nom;}      public function getAlbum(){return $this->album;}      public function getStatut(){return $this->statut;}      public function getDuree(){return $this->duree;}      public function getTracks(){return $this->tracks;}      public function getDatesortie(){return $this->dateSortie;}      public function getImage(){return $this->image;}      public function setNom($nom){$this->nom = $nom;}      public function setAlbum($album){$this->album = $album;}      public function setStatut($statut){$this->statut = $statut;}      public function setDuree($duree){$this->duree = $duree;}      public function setTracks($tracks){$this->tracks = $tracks;}      public function setDatesortie($dateSortie){$this->dateSortie = $dateSortie;}      public function setImage($image){$this->image = $image;}  }  La class Album a huit propriétés privées : id, nom, album, statut, duree, tracks, dateSortie et image. Ces propriétés stockent les détails de l'album.  Ainsi, ce code définit essentiellement comment un album est représenté dans le cadre de cette application, y compris les données qu'il stocke et les opérations que l'on peut effectuer sur ces données.   1. Le manager d’Albums dans Models :   https://lh3.googleusercontent.com/hUNSvSNVqPfWOq1mD-5H1fXndLbkGbFzUbZGlJPzQJOPiRkoQASmKV8fMJaXZ8CQg5MU-xIxXr4hccUeRY9VQ7T7hDtrvAwRk-nJveDnurPQan0O9298Y5Fpg1HolPwDgGILcpunx4mdWiMCSPsm2GE  Il permet de gérer les albums sans passer par des attributs et fonctions “static”. Par conséquent, je crée un manager d’albums.  require\_once "Model.class.php";  require\_once "album.class.php";  // ajout du tableau albums dans la class Manager  class AlbumManager extends Model{      private $albums ;  // tableau d'albums  // L'objet de type $album est transféré en paramètre de fonction dans le tableau $albums[] qui est dans la class album (self)      public function ajoutAlbum($album){          $this->albums[] = $album ;          return $this->albums ;      }      public function getAlbums(){          return $this->albums ;      }  // Je crée la requête pour récupérer les albums dans ma BDD      public function chargementAlbum(){          $req = $this->getBdd()->prepare("SELECT \* FROM albums ORDER BY id ASC");          $req->execute();  // fetchAll permet de récupérer toutes les lignes/ FETCH\_ASSOC permet d'éviter les doublons\ $mesalbums =/ attribut $albums          $mesalbums = $req->fetchAll(PDO::FETCH\_ASSOC);          $req->closeCursor();  // Parcourir le tableau $mesalbums grâce au foreach et récupérer chaque élément          foreach($mesalbums as $album){              $al = new Album($album["id"],$album["nom"],$album["album"],$album["statut"],              $album["duree"],$album["tracks"],$album["date"],$album["image"]);              $this->ajoutAlbum($al);}}  Ce code définit une classe PHP appelée “AlbumManager”, qui hérite de la classe Model que nous avons vue précédemment. Cette classe est destinée à gérer un tableau d'albums et à interagir avec une base de données pour récupérer les informations des albums.  Dans l'ensemble, la classe “AlbumManager” fournit des méthodes pour gérer un tableau d'albums, y compris la récupération des albums à partir d'une base de données et l'ajout de nouveaux albums à la liste.  Grâce un à un echo <pre>, nous pouvons voir que la requête afin de récupérer les albums dans le tableau albums s’est exécutée avec succès :  https://lh5.googleusercontent.com/cz-9fBhKJGMm53EeQye4o6sWat61MJS0BA9BoBAqc3lHKeIXA8YRu_ekj8Q3Glkb3kDviBFxOY1ZddCI6dOGzgTAxTTRTygFohYs085C-20gFSRymvR4gDc7mOxvWrbHsNtktztLQRqpnKAAvaoL15A   1. Le contrôleur d’albums dans Controllers :   Le contrôleur d’albums pilote la récupération de données des autres fichiers et du lancement des fonctions pour renvoyer un résultat cohérent à l’utilisateur sur la demande de routage :  https://lh6.googleusercontent.com/k4hj4EpYPTU_MuW1ZN8xb4OhLwqQ24iGOLCif0Lyi5kuEoVLnYkhqNydL8LBRiQVdU1k41mDGE0yzptkBZFf9CJK_t2TFE4V7jiOo12l1PLmNghdKyUxYi1c2x1iYOnMJx7Ar0_scyDM3yzGwFoWscM  <?php  require\_once "models/albumManager.class.php" ;  require\_once "CommentController.controller.php" ;  class AlbumsController{      private $albumManager ;      private $commentairesController;      public function \_\_construct(){      $this->albumManager = new AlbumManager() ;      $this->albumManager->chargementAlbum();      $this->commentairesController = new CommentaireManager();      }  // Fonction afin d'afficher la liste des albums dans la page albums      public function afficherAlbums(){          $albums = $this->albumManager->getAlbums() ;          require "views/album.view.php" ;      }  Ce code définit une classe PHP nommée “AlbumsController”. Cette classe du contrôleur est utilisée pour gérer les interactions entre les modèles de données pour les albums et les commentaires dans un second temps, et les vues qui affichent ces données à l'utilisateur.     1. La view de la page albums :   https://lh4.googleusercontent.com/G6JE1fB--MaVOP7CN0JCtQMLw-s-Ale0HiUW_XKWyHhnlJNdXPg9fZlTBM4MlEFvsU5_khapbRsV1pVXCNbSADf_sBsvBgFP-qIkDYfkSMOB9gsucF1_wOYieZFp9xZc1XlhkKGvtAI-kRHOlLywYAU  <div class="content container-fluid my-5">  <!-- Boucle FOR afin d'afficher chaque élément souhaité de l'album grâce aux GETTER & SETTER -->  <div class="row row-cols-1 row-cols-md-2 row-cols-lg-3 g-4">          <?php for ($i = 0; $i < count($albums); $i++) { ?>  <div class="col">      <div class="card custom-card">          <div class="d-flex justify-content-center">  <img src="public/images/<?= $albums[$i]->getImage(); ?>" class="card-img-top fluid w-50" alt="Image de l'album <?= $albums[$i]->getAlbum(); ?>">          </div>      <div class="card-body">  <h5 class="card-title"><a href="<?= URL ?>albums/l/<?= $albums[$i]->getId(); ?>"><?= $albums[$i]->getAlbum(); ?></a></h5>  <p class="card-text"><small class="text-muted">Par <a href="<?= URL ?>artistes/l/<?= $albums[$i]->getId(); ?>"><?= $albums[$i]->getNom(); ?></a></small></p>  <p class="card-text"><?= $albums[$i]->getDatesortie(); ?></p>  <p class="card-text"><?= $albums[$i]->getStatut(); ?></p>      </div>   </div>  </div>  Ce code HTML et PHP génère une grille d'albums sur la page albums. Chaque album est affiché dans une carte Bootstrap, avec une image, un lien vers les détails de l'album et de l'artiste, la date de sortie, et le statut.    La boucle PHP “for” parcourt chaque album dans le tableau $albums défini auparavant dans le contrôleur :  $albums = $this->albumManager->getAlbums()  Tous les albums sont récupérés et à disposition dans la variable $albums.  Pour chaque album, il génère un bloc div avec une carte Bootstrap.  L'ensemble de ces cartes albums est affiché en fonction de la taille de l'écran, en une, deux ou trois colonnes grâce aux classes Bootstrap spécifiques.        Définition du chemin dans la barre de navigation :  https://lh6.googleusercontent.com/Fjt5yUkpaMsaUi06Ihw3VfglekSRSlyX1B0QHkwpEjV7qdwoEALPijc22MNmDDabo-yIGdfXdvqvhgCx5mIG5rdhvolb4_1U-opVliqASBrwymOCCoLVSWJCOGnhxuKGxNLrIDb0UqXQREmO3j9e9bE  Indication et redirection du chemin dans le switch de l’index.php :  https://lh4.googleusercontent.com/IJWYM94UIU4z5XPIonkVWv4KiXMh-R8SjCqUF85Zh2F5HA4K__3rx4GRLc4AP_kH7l8H3FOIy3dj1nay3xUNLmbKvfn1EcqF1AT-gduESrBc4LZY-w8I-S0bGwXkGOutjRWP6j5C0e7TIsJU36irRUk  J’instancie mon contrôleur d’albums et j’appelle la fonction “afficherAlbums()” présent dans le contrôleur d’albums.  Résultat sur le site :  https://lh6.googleusercontent.com/uyfruSOwsJifiy6Y4NxzoQ0RMCR7lUudJfVMOt-8zdsq9vol9VbfwW7exvLybcg2hsppNtpAop1cqvslnF7LoNs5avqJjtdKotvgVe-2zoFeCFyXKyDnA_HdyjKYaPxhTCr_4RNfPw5NgWCM1J2ib_w  Cliquez sur “albums” ->  https://lh4.googleusercontent.com/rae18P6acA12E3CtnBdAH46UjzYa82bLHEe_FuJro7vp9riPpAZrUoGkwUqqK1iyQMJAAJJmhbzbXduKPUkM_iX6TewOhArmTeLvI9RKQi4LS_trKxsHN5sOrL7jxPN6sbIg4BtBQBp3orfZXkG1MVg    Nous pouvons donc constater que le code est fonctionnel dans sa structure MVC. Grâce au modèle, le chargement des éléments vers la BDD s’est executé, ces données ont été récupérées par le contrôleur et transmises à la vue de la page qui les a affichées dans une card Bootstrap les unes à la suite des autres grâce à la boucle for.  https://lh5.googleusercontent.com/Le3CZocT2UV0epphyxDVzXctu9Af-a0I94RRmrT4HzZlMa6fBwSaZpxJ7u2XMBeNx3ydPmQMWi1FL_V9YtdhIS5nrc1c2uG5X0YzUv58sG-_ash98o6Lrg4R_ROvzRElAQYHK2naw_o-g5s680p-h8Q  Par ailleurs, la réécriture de l’URL s’est bien exécutée.  Nous pouvons réécrire le code et le modifier afin de l’adapter pour l’affichage des pages Artistes et Playlist.  Playlists :  https://lh4.googleusercontent.com/quWwhSVMzszA3JI2W_aXuM2_jfvDGvvI9R8ZrTAfwIvsybdZaofjAM4IX-lV9gJgpOYUV0_UTk2u2dOaWhdDd1BX4kOwAActSbGPzWYebOFYntp_oOGWkgWGUIw6f5s-OiwIEaatkrlXIzWQBud6tmk  Artistes :  https://lh5.googleusercontent.com/L2lNkoSsLYC-Fd6V78frfNmRdfiuK_mOClIRhko0Z8_PSNmGMtiU_srXdbo_xj68jrNLTNMGJl2Y6JXXf-ZgpxUgBosKvSvI79S_9lRejqZxE1OgOuEPr0Q9yW-_eryVpw3ef9txRhp2yBSg9yE0llU | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| PHP, HTML, CSS, BOOTSTRAP | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI* | |
| Chantier, atelier, service | | | ATELIER | | |
| Période d’exerciceDu : *06/03/2023*au :*02/06/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 3***  | | ***CRUD*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| **CRUD**  Voici comment j’ai intégré le CRUD à mon site afin d’ajouter, extraire, mettre à jour et supprimer des données de mon site  J’ai commencé par créer les routes menant vers les pages permettant les différentes fonctionnalités.   switch ($url[0]) {              case "accueil":                  require "views/accueil.view.php";                  break;              case "albums":                  if (empty($url[1])) {                      $albumController->afficherAlbums();                  } else if ($url[1] === "l") {                  // affichage d'un album}                    else if ($url[1] === "a") {                  // ajout d'un album                   } else if ($url[1] === "m") {                  // modification d'un album                  } else if ($url[1] === "s") {                  // suppression d'un album}    **Affichage d’un album :**  Le 3ème élément de l’url doit être l’id de l’album à afficher. Je crée donc une fonction dans le manager et le contrôleur qui permet de récupérer l’id.  Je n’ai pas besoin de faire de requête en BDD car la liste des albums est présente dans l’objet et sera actualisée à chaque instant dans le tableau d’albums disponible dans l’attribut portant ce nom.  Il me suffit donc de parcourir le tableau et de rechercher l’album qui dispose du même id que celui envoyé en paramètre de la fonction. Une fois trouvé, il retourne l’album au programme.  Une fois l’album récupéré au niveau du contrôleur, j’appelle la vue permettant d’afficher l’album.  https://lh3.googleusercontent.com/Uy3AqkIFPlhkcpsEfYCi845M2O3Xrcq7ZIl7eiLKiPsTE55HpVK2oEF6aei_WJVdgG0Az9-lvLg5cRZLjldRLdaw7dE77DJ_oXY95L6VhvNem_F81hjXjTYng4rXO2Oa6SpsE3iI21bcFsyKS77K6aE  // Fonction qui permet de récupérer un livre selon l'id      public function getAlbumById($id){          for($i=0; $i < count($this->albums); $i++)          {              if ($this->albums[$i]->getId() === $id){                  return $this->albums[$i] ;              }          }      }  Je parcours ici mon tableau d’albums dont les informations en BDD sont déjà pré chargées.  https://lh6.googleusercontent.com/WyNsCMWD_jkgqXABr54lYDACdRe4o-zAUaSqYCjhup9fYTBOFq_H_lg7sjNwap42en8tL-71mwFkDcsJEd_o0fGlNVbsw-vPnqUu31-xV8MliGPxycYAg2721aoMDtLkukZR9AXM2kvda7-w3glCDVs  // Fonction afin d'afficher un Album dans une page      public function afficherAlbum($id){         $album = $this->albumManager->getAlbumById($id);         require "views/affichage/afficherAlbum.view.php" ;      }  Mon contrôleur dispose donc de l’id trouvé dans sa variable $album et appelle la vue afin d’y ajouter les informations de l’album.  https://lh5.googleusercontent.com/jOPPeHPnWb01uNuQQMCNcKdYf2Hp51_HXUXfujR9GLKiSWOQ3mNERBGiL5Qc-ATwmqCQYbIefPYoela39aG1_nCA619b5CI99oI3S-suNHg5VD7_kLepnS9mmONrFZ5mqeOYZtaySfsCN2llM-Uc-84  https://lh3.googleusercontent.com/Ee0GiZrXBHtATUjP7kS1UjX6_dMIEsHTOk4u-rfAeikOOKzqIqDaFJvODcXrN9FZvq1UD_ESWaLm1vyufOUMPliz4FfTsa4W3mfn5P8kVYZwni9BjZdwJirMrRSNcytDiLAeCoed0x_hXbaneNeDyn8    Dans la première colonne, une image est affichée. L’image est obtenue à partir de l'objet $album via la méthode getImage().  Dans la deuxième colonne, plusieurs paragraphes sont affichés. Chacun contient des informations spécifiques sur l'album, obtenues à partir de l'objet $album.  Chaque information est précédée d'un texte de description pour indiquer à l’utilisateur ce que chaque donnée représente.    Ecriture de la route dans la view des albums et dans index.php :  <p class="card-text"><small class="text-muted">Par <a href="<?= URL ?>artistes/l/<?= $albums[$i]->getId(); ?>"><?= $albums[$i]->getNom(); ?></a></small></p>  <p class="card-text"><?= $albums[$i]->getDatesortie(); ?></p>    else if ($url[1] === "l") {                  // affichage d'un album                      $albumController->afficherAlbum($url[2]);                  }    Résultat dans l’URL :  https://lh5.googleusercontent.com/cF2lw8AmCOdOuBlLRxupRDTTEdo3y0g2KJCCb_71M4EZfr9Jdple1dkcCUGvJFy1w74iM2EFUX-SCgj9mEKlHEFGYSAzbUkJuCNX_C2_6jW3ipm2-4fb_t6HfaPJHXSibh5Uzmlx9KaXgwQPI034e68    Résultat à l’affichage :  https://lh4.googleusercontent.com/uTT-h8CImUCo4OqmFgwcrIL2VT3rNYc9_w2E1Jblu3EOf2wYeHY6wZ489ISeN_7ZprrJWwsvbp6_oegmeNUxQObH43I7YeKV_DXzG6j6tW5pSpD1hnkYJiSI94VxEnYoNcQe3D2cmkD9RXcNvSrUgkU    **Ajout d’un album :**  https://lh3.googleusercontent.com/aTOGiuEd6vEElNvSpNV2xacJL2wdSBw_QaCLiGdmmPN_yzS4w4NQE5oDz8RY88XouqAX1qYCzevnPFm2RzVno2NL9innXG7MXoKZQgmuk4O3FrHrHO6txQ1Zx3s-2jErYXJU04LZF2X-8J6fk_RFp6k  https://lh5.googleusercontent.com/x-xZxivP1UPrlvHepj-IMu_o3UssXmZvapw0mm3RqzkusUtoXnatISYhQJTnSH5jv_uqQ1_T_b2RDlDrgbnQ1jlItCDJZ3bQr17PYt2lFRJqXi61Y2G-Acq9euzms4c1FJzlhpq-BA8HRyY3C8lybIs  J’ajoute l’URL qui amène vers la page d’ajout lorsque l’utilisateur clique dessus.      Création de la vue correspondante :  https://lh5.googleusercontent.com/Mm_fM2uwR4R8Ym5baFkGiRja6J9oODh7xGUuGkoZIcRmEIJT4FJi3Bg4jWPwd_MW8wnPB1Fdc08Y_F0rsST5p2hEiYKLK1CDwpgi6QdjHTKpWZZjS1hHof8iVj7P7UCB5K1eNJjysdB-QPAvpwlAXt4  https://lh6.googleusercontent.com/2Ei5Uq5iBiWpbYQHKYUn5uHJodezqd2c3q7VhApHYJKYyi7lsSjCMrqj22th0JlsfSOAHQ2-JsVKh3tLnjYtU7TonBz-2n7XpetEXMjtqfmPGa36JNMMl3R_NkgGIWHup-az6MLcCZ3SYd6JncnLthM  https://lh3.googleusercontent.com/sIEs3fgPH3wBhhdmg8jOmxvTyNQS4XZxAahF3t2RwOgYJtNMwu2AZVujjtzqHubscNggdc0XtVl4D2Xk3DPhMFIIDNNM9M2XONzbE-nlnj79-ASl_jqQISBsfhOSfj3XqhTEtA8fpb19SIQfvjkpmVU  Résultat dans l’affichage :  https://lh6.googleusercontent.com/JbQwyvVzy0_hMfsNh9a4oaINJ86ErlWLlG-NQDvO1BjmZoQ3DZp13jT5f7WAv5oBRUXNBhYAknwtA06whEHdr2Kuep2j29gmdK6N63oL2liX-uF5vtXEfeF4t4X1wA_qLeFEF1ghjX6exyp8i1pCq8Q  Ajout de la fonction “ajoutAlbum” dans le contrôleur qui va permettre le lancer l’affichage du formulaire d’ajout d’un album en BDD.  // Fonction afin d'ajouter un album dans la liste      public function ajoutAlbum(){                      require "views/ajout/ajoutAlbum.view.php" ;  }  public function ajoutAlbumValidation(){          $file = $\_FILES['image'];          echo "<pre>";          print\_r($file);          echo "</pre>";  }    Ajout dans index.php de la fonction d’ajout du contrôleur :   case "albums":                  if (empty($url[1])) {                      $albumController->afficherAlbums();                  } else if ($url[1] === "l") {                  // affichage d'un album                      $albumController->afficherAlbum($url[2]);                    } else if ($url[1] === "a") {                  // ajout d'un album                      $albumController->ajoutAlbum();                  }                  else if ($url[1] === "av") {                      $albumController->ajoutAlbumValidation();                  }  https://lh3.googleusercontent.com/TgQAtlI8JBw9OgNb9RQAVwmgKw7oTETLUB3QGodSqx8_eWBCLO7YQAZPro0tPkxF3g0M98LIzyHB6jc_YeYZxjpbANRg-yLcLyJQDYPCPd13BKQINg42H1I6N0Fj5UaFhtodLzfZ4kQAD7FE3ivijiQ  https://lh5.googleusercontent.com/0DWvc2MNe_XJ4-JWGwW-WWGht0qsZRGBT3quRDRGpwGe3jfmliN3UfHxOO7KHFeMG_mMpKXEnvj_7-MsaxH1TE6RPXpVVmIcUxlQFiQuRmUiSFFrxlWEEkus2gvaHjw4RtydEQOF_-bd2m8ReB_wEG4  Nous pouvons constater que les informations de l’image sont bien enregistrées, nous pouvons par conséquent les manipuler en créant une fonction dans le contrôleur.  https://lh6.googleusercontent.com/nuDm3P0679x5pmBzcDMk0_Pv7vlalFCk0Al1ziFFbe4An6WVcjzV89IGovqNqqgEfgH1GDLy-JpqYezAXGZ5OBpdvAQxL1OXpMQ1O_ysxrYuRnVgG6jmCzW907EWx_pTYRRTQpo0nkPUXpMdm25SSrM  La fonction “ajoutImage()” est utilisée pour uploader une image sur le serveur. Elle vérifie plusieurs choses :  Si un fichier a bien été fourni et si le répertoire cible existe (sinon, elle le crée).  Si le fichier est bien une image et si son extension est valide (jpg, jpeg, png, gif).  Si le fichier n'existe pas déjà dans le répertoire et si sa taille n'est pas trop grande.  Finalement, elle déplace le fichier vers le répertoire cible.  Si toutes ces conditions sont respectées, elle renvoie le nom du fichier image nouvellement créé, sinon elle génère une exception expliquant le problème rencontré.      Dans Models je crée la fonction “ajoutAlbumBd” afin d’y intégrer les requêtes en BDD.  https://lh4.googleusercontent.com/vTYTm7O8IhWM9lL1jIXBpyxIPv-mPya4lF95pqccEBHpmUMcKwaoTAzzS-TkZk54aWrNIidCw10KEdkza2yP4_CH40VUQFFUZ94lAqK9bU9ER-gzHu2KEicua1X566oIDNF4hXwz9VMyyZuQs4aTims  La fonction “ajoutAlbumBd()” est utilisée pour ajouter un nouvel album dans une base de données.  En résumé, cette fonction ajoute un nouvel album dans une base de données et, si l'opération réussit, ajoute également cet album à la liste d'albums gérée dans le tableau par “AlbumManager”.    Nous pouvons tester la bonne réalisation du code d’ajout :  https://lh6.googleusercontent.com/7Vlbt4UK97r5zoabtVJP6aXn1Gqy6T7BTmv6UVxEopWnXWL70Gz5eNCDxbOtTWmDAClex27p-HPVDnjEQW29oXSAOh-GHDjUyk6C4wr5TPbFbX_RJyOvbBwj-LP0YoFgAbI_eLj0xiWWwG8GXlGoTHg  Résultat sur la page d’albums :  https://lh4.googleusercontent.com/44fOPRF4eGJDTKNpliY6QwF8tgWBMRMnPXQ8gGvNQ9zVlL449LC2mSs4l5lWt6BcwJGTMF67-RN4Ia7VVeb_8No67_G6VHaxc9w45HvAp8pVwjWYn--DAe2BrJUBlFj7Ev-Me6-SVvIYpq54GxSnAgA    Résultat sur la page d’affichage d’un album :  https://lh5.googleusercontent.com/r1R5FQuJSIDy40VliauBSpe0taBgQQ9W7bpD3EavADiuOW5FDLRuoGwoCm9uCHJmd6ZQna6LvrlLqR4azYFfnl_I1fHpLxN4-Hsjlo93AeMkmwyj9dGPkwkDzSVAlFkJ0HoO4iRAFaucxndi--3uvMw    Résultat dans la base de données :  https://lh3.googleusercontent.com/DHHoSHfSSZLb7UpSKs5nsu96EmOCZmPgyiV806n0VHS4zxoJ9DO9yJ753dFXwNurn0GcyKMrT9q6-Ybr5ObTTO9kU8HFoGRvSRlGw25csww5PuRtkzNdX2onmWXtDgQnjld3MqJR5wAman9P0azK9yE      **Suppression d’un album :**  https://lh4.googleusercontent.com/mMSvESSzYkJ8Oge1vpCTr9-b7ITyx8nUxuqRLvbbsHuCrmmcwnwstYyv5hSMLJ81SWFD7L1K7ubyi1hsfc3mUAFL9qaarmhKJQqdWrt5-flfpvlTIU_X-O-F1CkYiEgaVNmUhMAH6Oj58dWmqFCkUQU  https://lh4.googleusercontent.com/YVML4KWl-ydhm7y4q7iToZVVnjqHrv0AJle9XDM5b4N4zU3ylvNXJAI4zOi2TtRWdnDo__P_RF--ztwde0wzW3NXspMpPe0XakXhrJraVyihGYYzF6eK3BDfhtOkfGuNC9wCI9kNNHFu9AHGTkeHnm4  Je transforme le bouton en mini-formulaire qui redirige vers l’URL de suppression qui prend en compte l’id de l’album concerné. J’ajoute également une confirmation JS “onSubmit” de suppression pour l’utilisateur.  Ajout dans index.php de la fonction de suppression du contrôleur :    case "albums":                  if (empty($url[1])) {                      $albumController->afficherAlbums();                  } else if ($url[1] === "l") {                  // affichage d'un album                      $albumController->afficherAlbum($url[2]);                    } else if ($url[1] === "a") {                  // ajout d'un album                      $albumController->ajoutAlbum();                  }                  else if ($url[1] === "s") {                  // suppression d'un album                      $albumController->suppressionAlbum($url[2]);                  }else {                      throw new Exception("La page n'existe pas.");                  }                  break;    Ajout de la fonction dans le contrôleur :   public function suppressionAlbum($id){          $nomImage = $this->albumManager->getAlbumById($id)->getImage();          unlink("public/images/".$nomImage);          $this->albumManager->suppressionAlbumBd($id);          $\_SESSION['alert'] = [              "type" => "success",              "msg" => "Suppression réalisée avec succès"          ];          header('Location: ' . URL . "albums");      }  Cette fonction supprime l'image d'un album du serveur, supprime l'album de la base de données, informe l'utilisateur que la suppression a été réussie et le redirige vers la page des albums.  Création de la fonction “suppressionAlbumBd” dans Models :      public function suppressionAlbumBd($id){          $req = "          Delete from albums where id = :idAlbum          ";          $stmt = $this->getBdd()->prepare($req);          $stmt->bindValue(":idAlbum",$id,PDO::PARAM\_INT);          $resultat = $stmt->execute();          $stmt->closeCursor();          if($resultat>0){              $album =$this->getAlbumById($id);              unset($album);          }    }  La méthode “suppressionAlbumBd($id)” supprime un album et prends en paramètre de fonction l’id correspondant à l’album sélectionné :  En résumé, cette méthode prépare et exécute une requête SQL pour supprimer un album de la base de données en utilisant l'ID fourni. Si l'opération est réussie, l'album est également supprimé du code PHP.    Suppression d’un album sur le site :  https://lh3.googleusercontent.com/lpDvvzCweeAsOSlNs_Wbu0-DSmEUrQ8XZwWQyGit7bdz-B3XxrBSJKWrTJ5ginFx64YV6mr1PvB1sZnzWvX8feUuM5uzNqR4PZAY_gtKXWCvwzGbQ5k1fPIKYMMfZax4Dq49VlWfhP1lUEkEM-Il1Ao  Cliquez sur supprimer du test que je viens de créer.  Je supprime donc l’album et voici le résultat sur l’affichage :  https://lh4.googleusercontent.com/eVwzVRpWw87l8sl8sdUGOIdmjE1Gw0TW9cXQUacB5kYAjpWPt3g377l829YpbkHK1yHe3q-lhxvMuWtbV2apzgEblCIOr70IyTz0p3OAwv37m5autqlsP_wUpUHg0Cl1wtdkYQd1S0WJUlClqhhx1ys  Résultat sur la base de données :  https://lh4.googleusercontent.com/0biDFtoGftx5VuhLPFCD8NzSD6kxvlIbc04stITp6oXmZzxRxs1F4LdK5eA-NGOc-jRGiSTDJjN2sVCco7WQscb6u8eknvldw4hbdvuiclk_9KmFjR8GSy6nQBiKLj_LTfWNTLdLwT8u36jaSVXvDyM      **Modification d’un album :**  Je lance une nouvelle route pour le bouton Modifier :  https://lh5.googleusercontent.com/ztOzHLCBEyea3iK-Fm5z1eyb6l_N2nc5iEmwh6Ado0ubKnLJO76WN-TNR17hko9bXHeBEIQM_0lrEEY3FOFdhPzlmL8ebekbYzK8woL-K_BPl_snrm86WloqVLTsf-SS91RcCCRKyi-Bbg-nEqOWLKM  Ajout de la fonction “modificationAlbum” permettant d’afficher le formulaire de modification d’un album.   else if ($url[1] === "m") {                  // modification d'un album                      $albumController->modificationAlbum($url[2]);                  }  La fonction doit récupérer un album afin de l’afficher dans le formulaire de modification.   public function modificationAlbum($id){          $album = $this->albumManager->getAlbumById($id);          require "views/modif/modifierAlbum.view.php";      }  View de la page de modification d’album.  https://lh5.googleusercontent.com/4MIL0Ctvpcwf5dOXBA2FLNTb9tHrCGUV1m74N3GrQO3ptD9CU0Mpz1etkhCbsHxizbrixPGiEzig6lqpPrhMhTZbZ0_g-mZpYE71qFCVyJ70fcuWq2ZE7CVdxCseOWtN-9CXIj90DHQyNlOFMzQ5aP0  https://lh6.googleusercontent.com/bBtlueSFR77Cf9hFDAE_BHXHMglUgO3AgkoMbfHOlem-tDMHXGw6AUa1tPrdW4lHPiyA64MxBWI83QVDICOC_tw2RzIEbAM046EIvg_tv1kLPfTgjKakxt2e_gSIrQoUqNdfEPzj83mH715betpyx4Y  https://lh4.googleusercontent.com/vpCQxzmA0T4cz70P0Y1UZYikZKW_wUOBFORZmqLG0oyllEqv-8bFLTCPwjWSCzURa2rSoy9Y2QjcqA8seDhOXhNdXEtdzfOysrh7ulqaKvyq6mh358IRzowMkWY_A3GgrNgUjFexAjo9dZH6fSMAhXU  La route "<?= URL ?>albums/mv" dans index.php renvoie vers la fonction suivante :    else if ($url[1] === "mv") {                      $albumController->modificationAlbumValidation();                  }  Visuel sur le site et dans l’URL :  https://lh3.googleusercontent.com/D4r1MCJMmHfS1oAUMngWYY_4aM239EOagm1cbhV07eae2G49IQKjomtDWvyjF-Ik1O0VbBthatoVCD0dSf7iOEd6WjgrEUTFXF6CytMl45yBP8tbVznxtKC4nHTt83LZXa7DoKaduyvBGAXZ0GOPr-I  https://lh3.googleusercontent.com/fTCnnjtl97HkR17UMYMYnGOr_uucCmVgY3yg587adxuC7pgU74Z27hGUNzLZXH12-kuRM8b2EOUD8gM6opI0TsCYSEKKrO17LK65IXtN4P4tH9MDvjsKTxU9SyofgMYV5LtW90PSekhAdZTYwRNZZ2w  Nous pouvons constater que la récupération de l’id dans l’URL et des éléments de l’album dans les values des inputs se sont bien affichées.  Dans le contrôleur, je peux commencer à créer la fonction “modificationAlbumValidation”.    public function modificationAlbumValidation(){          $imageActuelle = $this->albumManager->getAlbumById($\_POST['identifiant'])->  getImage();          $file = $\_FILES['image'];          if($file['size']>0){              unlink("public/images/".$imageActuelle);              $repertoire = "public/images/" ;              $nomImageToAdd =  $this->ajoutImage($file,$repertoire) ; // upload de l'image          } else{              $nomImageToAdd = $imageActuelle ;          }          $this->albumManager->modificationAlbumBd($\_POST['identifiant'],$\_POST['artiste'],  $\_POST['album'], $\_POST['tracks'],$\_POST['duree'],$\_POST['dateSortie'],  $\_POST['statut'],$nomImageToAdd);            $\_SESSION['alert'] = [              "type" => "success",              "msg" => "Modification réalisée avec succès"          ];          header('Location: ' . URL . "albums");      }    En résumé, cette méthode gère le processus de modification d'un album, y compris le remplacement de l'image de l'album et la mise à jour des informations de l'album dans la base de données.    Voici la fonction “modificationAlbumBd” dans Models qui contient la requête :  https://lh4.googleusercontent.com/mULN_NqhvOnJ5FDaMjnNRyW7o9hLk0rzpMgCNDUofmT0oHduDVIafqx4xZdysv136Vo7cYwNS4Qpi2J0w-UmAmTtgRcpxa4hE-yMhssnf074NI-SVQ4mpd7BYsdB1cwenjauHQ9VGrNLDpkcE2X5xL8  https://lh4.googleusercontent.com/p0QBWGwJQLZ7NqDI8UkWKSZ_igzWTXpUp1D1f51qgETMa57koliP497iEz2lWXVWBVutl74npDcTcIyipXLUtrCKDQZiE5bXzlz8xstRALA8Wkp1duO6m6HWSmJGqJzar_WDv4ZmGmehXyWKnUn49Mg        Résultat sur le site :  https://lh6.googleusercontent.com/dU5me7ChfG3gq_DKSFuLFRnt48MneNr8w3dkPx618ynnVU92UTXy5TJVvaC5CV9sdmQ97Q4yuml_o_Uv8eYYwRv9MGjHDhPjW9DF_jo8KCkB9UlOt7KnhVwNdis_dHLWZ-_bSmWzl3122AgXGWnQksM    https://lh5.googleusercontent.com/adrmLbGIuog4ROZ6DcPc2OdIML_kiaPBY9ovUWTIkowzwJcSmsYY2s4V13Wmdz_JNrNnZUPwekBuYY6iPwHOsPOYbQWwfakRKt2k2gSHjI4gPbpZJ5RkfFnCnd4kZILbOgUYAaFfyzFvQOvTlfJdnVI  https://lh6.googleusercontent.com/-bDaXnsdyNvoN7tDsRjQjMazxlbzG5e9XBeZI8O5c-aNUvaNuxi3K4Hu1wXaHmJYslynaViZ7hxqK2IRrpNp3aBfK6c5GFfhUptD5Bxu71h9MLOTrtt8eO7I8LfOzDW_eP08eHjBAtSpqrLazdUD9bE  Voici comment j’ai développé le CRUD pour les principaux éléments qui composent le site. | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| PHP, Requêtes SQL, HTML, CSS. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI* | |
| Chantier, atelier, service | | | Atelier | | |
| Période d’exerciceDu : *06/03/2023*au :*02/06/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 4***  | | ***Connexion/Inscription*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| D’autres part, j’ai ajouté des fonctionnalités de connexion, d’inscription et d’ajout de commentaires pour apporter de la personnalité à mon site et de la proximité avec les utilisateurs.  Voici les éléments du code qui composent ces fonctionnalités.  Connexion Inscription :    J’ai commencé par créer les pages html correspondantes et les ai stylisées avec du css :  https://lh5.googleusercontent.com/8cmpZ2Uxt6fI8qZFrTZxGk49LRAw34322Sk60JClpxS45fPLndkwWV3iMFGhiWBWK7MVxb87_LppJw66hTcJjBZcdble8F9qmC82gr9iuGipdPlhFGXUlNKIVkQIjFcxqC4AIv9SCz5D0H-vZZnpyus  https://lh6.googleusercontent.com/wh9llUFTnj1Rdsw7ZFV6BEq5I-pKyzgPh5lt6cbrilmlxalu474pnr4Q9Ge7oOJ4JjMCeSwBkHH7od6WWkeBV8ogJhjq7BUSqqyM-X6f_wyj6Ik0Cn_8pzJXPDxr1Jsu1YnrOzTkH1G3oVUPtGZEYa0  https://lh5.googleusercontent.com/r32ZTEjVQva_BdEjf9MgbZ95dGo6nxPg0tV9j9lTq3S5LeCPD-q8rFJd4yIu-7Vqe9alz8r4VSS4-dvQt5MU9KbNXzqux4x1-ozyMZpz2JiDiU-mL6eSY7C-wbkoUlDGwXyJ9DCv5z3lJlQuyi2w0WE    https://lh4.googleusercontent.com/1fB7Gf2GYXQR2LjjDMfbEpT-UaUDNUq96Go_tb1IQZB7QDG3OK601qSxogeSs8Ym5mplkyI0WChmg5wdqBVpC74izbzgtJQ38LKwJgl0KME3Gw2OejYmj7Gd-3BFlJKtlHIBOs0ys2qMjIBXoFT2pGo  https://lh6.googleusercontent.com/bvfPb6V2UhthMDOqlg-GxXAWV0z2Fw0rYNOePjhaLM55rZTbqW09U1QeV_IIOBxwrHs4q0WJbC9vXg_CwOZ6oR46sVN6LcTMlmsbAe6ngNnhYn4sJCyFalkk_QNKMNd2K6szHtFhFNBiBv1MO0AHBp4  https://lh3.googleusercontent.com/NnItSEBFAPW-7YYNQm1ZGskzGtG7pZveBbZvbZ6J8Wu_FD0M5yRMGFByAlwzuQqit5s3wtYZBpTCMgptKFFPrPP9QOg3eY2bUUXrSPV7RSee0mneyYp7HPygxd13iPP2QjhmUss4rAZJLsmKxeF-ZXI    Class et model pour les requêtes de base de données :  https://lh6.googleusercontent.com/XCSh1KjndCPqjXv-b6UHhKr2o9P8diz8_Irlm_YNBpKow-c2j3oXH319RjWiKmlAVLh2ZBztjAFlnK0hb14JEm0mDl0yai1nQMjeS1Mh-xlhlSrl_MpEsDzmU3EDWpt1YNK-m3p-0dZQzVRcqyhz0no  https://lh5.googleusercontent.com/ER2uAhJ-Zt5DCWH323T2qvp7fzT1xGKW2H5CKRaplOwZfLl-sLw5zkACLKftnmTus19M2CG2adzwz2vNZ7MPMw9W7WQQvgpCHCrxwKDTjnF11iNfKR3v4et2LwPBDxdRpmsyr5QApHdQl_BjbXLg0kU  Contrôleur de connexion :  https://lh4.googleusercontent.com/jEugKlkp9gqMIQe7xw6bkjQNGpDVj2TdkQ2hAzLzyURfHLbUxZbX7hrgJHtXVJGyXan_7LdZrJeBhf4K8Ia-U-4mqtKvqIlL0lB2Lh6kSW8cHiLePfPATRryUA7klSencR6z9cJ8f9Nsgu2dsjz3XLI  https://lh5.googleusercontent.com/CJxZjQCLyKAVWnPg_uE9zzQWl9KiNP2ByRzpAkL9q9dTxZu2frhkKA4-uEJjxU8Fc7Dsvpq3cwtySFcreiT2G5yyHWlt_bmAhS4UI2kMX1FWh4203i494hKUy6hWx0GH7t6XnPlyLGqe5eaybSBh1UI  J’ajoute également une fonction de déconnexion :  https://lh4.googleusercontent.com/srXLawkIF7njmBsL-vzV12kPUxPIteUUWFsEDwx1wIL5yLOyU0uKCa4gp4ptStZ9XPNanPmKeic695sP8TgjK4gDP0ovUHZrtn7sGne_Fxn8hkkMKzr-R6x7GuYOZSUs85is10Tce0_Z6u6Pc4IXxgA    Voici le routeur qui appelle donc les fonctions du contrôleur :  https://lh3.googleusercontent.com/IfzvYEvYUB_ylvZQB3G8KoKdJ73xUNQvAJX8Cfr9ASd1lIVN64dToEhLpcOp7XMSp9lYiK3WHWIPNxPYn7EO2u5aZAUVztSvsDu1QSdIrgkUVlzOe-bOKEMaFw5n0WoYsgxbKW2u0NcczmHuq1ANrK0  Test et résultats dans la base de données :  https://lh4.googleusercontent.com/uqdvUbFgPpnNH6b9_1JXzXrFfxfH61tdiFObF4epzQeSmdpDdLIWJoFHhfzpXFX72-U6jP_slvoHv32TH9WYlSaqd1rlJYIZZCb_PiwvlQPHlPMScXsZsE5sOp5MquxuSc6aS1lnSXee1qeAwL1sE0E  https://lh4.googleusercontent.com/TdjZ0VDWG2xz5N8SkbM2ccIXcohsvI0Ei9GG522CQQFXDDhEcGWNstXeOySRnw2nJdL3CyDXeKnBkAKgJxMtc1s7KlavAQnIpd975Ahxw-Lcm4CB7gskU0yyhBDWSgAFFF0CaVdL4aWyyFkGjHY09so  https://lh6.googleusercontent.com/KfIJCdltj9zD0zJKnR-obOuIULF0tCnuS5yduCLM-cD2E6YXEuSf4-0g_cgb6wOY_TnEVQtBZEKlHesKrIxoT8R6frWLxH0DAUM8l76QivLhrxkfIZvvP8OlmRl1GDOkASCfY19UOJmw_tFe_iTRyOQ  Tentative de connexion avec les mêmes infos :  https://lh5.googleusercontent.com/I34fP4ZA3mJmu4JYZYSD5BWmYYWJSB4wcsUOREwe6viX2Y8rQ0BBWLugPeAMT8QQkRmvJwTnmf6duIyHFEvynTAoyCuKF0QvKyn_rkWOcBJz-_1lqAu2naKaZl08-LTxlWZbqA6vJt0ao-dR05NRWNo  https://lh4.googleusercontent.com/vBnNI-IRAzMuzoJhg928eB9W_ijcv8fMMP2Z6wJrsAxJi8l2iqg_-1HNx1E4gRy5OAtxrR9yIdOlidFfhNKeVNccRYPh95YWSPapFFos7sJ3rt-NakczYcCn_FVhjVDC0TnJbgY5oA2YUBxBd6tPx9w  Les informations sont bien reconnues en base de données. Le programme récupère bien le pseudo dans la variable de session.  Par ailleurs, j’ai également fait en sorte d’ajouter une option de connexion pour les admin afin que seulement ces derniers soient en mesure d'interagir avec le CRUD.  J’ai donc réitéré le même code en faisant quelques modifications d’affichage en fonction du rôle (user ou admin).  Exemple dans la page listant les albums :   <div class="card-footer d-flex">  <?php if (isset($\_SESSION['role']) && $\_SESSION['role'] === 'admin') : ?>  <a href="<?= URL ?>albums/m/<?= $albums[$i]->getId(); ?>" class="btn btn-warning">Modifier</a>  <form method="POST" action="<?= URL ?>albums/s/<?= $albums[$i]->getId(); ?>" onsubmit="return confirm('Voulez-vous vraiment supprimer cet album ?')">              <button class="btn btn-danger" type="submit">Supprimer</button>  </form>   <?php endif; ?>       </div>     1. La condition   <?php if (isset($\_SESSION['role']) && $\_SESSION['role'] === 'admin'): ?>  Vérifie si le rôle de l'utilisateur actuellement connecté est 'admin'. Si c'est le cas, les éléments suivants seront affichés.      Résultat si l’admin n’est pas connecté et s’il l’est :  https://lh5.googleusercontent.com/_HlESF2J547y_TuwkSNynrSfGXvO4ddLka---KLVlQRYznS0Uibes8akK6BpH0h25ePNeHUqV2hI8290Pk0JflkBgtYn_tWItqtKtEGzBOGD0W9AICSEpHv-yFHBC6HsFDYFARSnuEiNcxzZqjrgBbo  https://lh4.googleusercontent.com/MWnhL3YuJMoBwl_YPguvqYau5HBZVqI6cgRCCflqvVnU9q_NPqjddLn5sdkoVHljqKw-tfrPPWMXmNxTHQUejjRJNW9Z2d1Keo4ENqgkKKSJ759PxvKALozBcErietyH9Is06izCY1YqJYYca67wsA4 | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| PHP, Requêtes SQL, HTML, CSS. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI* | |
| Chantier, atelier, service | | | Atelier | | |
| Période d’exerciceDu : *03/03/2023*au :*02/06/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activité-type** | **2** | **Développer la partie back-end d’une application web en intégrant les recommandations de sécurité.** | | | |
| ***Exemple n° 5***  | | ***Espace commentaire*** | | | |
|  | | | | | |
|  | | | | | |
| **1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :** | | | | | |
|  | | | | | |
| Commentaires :    Pour la section commentaires j’ai choisi de l’appliquer à chaque page d’album pour que les utilisateurs puissent partager leurs avis et analyses.    Voici comment la section se présente :  https://lh3.googleusercontent.com/lBkdQIlBiaGbJxiuipVayXG4F0j5kUFPTQR1wnWMkBrL4Io_H64iFxU-DwjTXKuZj00zAkjp5n-M5zRLdJzJ-6FnryFSFV0uJx6Q1c5WXNj8D8AjIYNoAoFQ61QyZ8xGS4Umv17KCB3dYcJD4e3PS-E  https://lh6.googleusercontent.com/NpXtCNbU5skysKaDBQyQzSJBA69eu_kobiY5YaP5PCtmy80kuqPWkLYSU4EheRJjsfN6PN3C6vFm3n-6-DMNZizSPgE06QQpsfLnR66R8ONrGHt7BOWyZ8ON5eMbXCvNSCJiEnRlRgQyp7stntp0Su4    Le code html se trouve donc dans le dossier suivant :  https://lh4.googleusercontent.com/zcD87S_hvLDDazQ-Rm5YVc2DhtpMBtBNF4z2y-CrBjhNDfyWThNu7bl_OsJRsGobdSMU2hsd250obVNwOBf9bq_ZkKNLxS-OKIrVOInZu0Omg_XZeAbyc1TdVZvVL8BtWZWCVlpK-MsyzmZdQXx6Kf8      Class et Model :  https://lh5.googleusercontent.com/jUlO1REc89iHcLHVHKW7UKPDkyTCQ_BoSKDWzYULY-6k3kuD6siTcBt4bxtVU9nbkmLMEfF6Olji1xBiGfeOss3mx4VN8bpIziDXQUXCNL6UzX6bxABxHvRmXsLj8kR7WJ5EtWIEegSLZKikBww2P0Y  https://lh6.googleusercontent.com/igoksHuGWFALm0DdrsFX79yzXroj6osnq3QOdjBrUzUZJkljGbxNPuNckCVx_wmOKf53OqvvdXk6HQqNn3PSkMC360BRjSE2cofKKKnSgmwEn-P8tmASx8x-Xt-fb_h1N_Db18NnDjH8YsmWKHYXEag  Contrôleur :  https://lh5.googleusercontent.com/ylHcT_MjUg9cKtrDueZ3ie3b_d0igm-n_KFNf3d5Hav_nfOtIsxh2h1F-Z-_QHxxNEWEfQRn2ux05O0spxTgxETyasCV0AUU9kRn2HiqNKPZAAIO2Cm9tXMG-RAX44mNlkdwmXZTjbQCVmH3Bq2zIbk  La fonction “addCommentaire”gère donc l’intégration et l’envoi du commentaire vers la base de données.  La fonction “getCommentaire” récupère les commentaires afin de permettre de les afficher.  J’ajoute la fonction dans mon routeur, précisément à la page d’affichage d’un album :  https://lh5.googleusercontent.com/o4JU8CWeAI-1h_gbUKJ1sQqnVPzptmf99RdC_4ijP1UrOZUeqGigoPPhwRDBqRNrHGXZcFHo-5ohLx4yygQFIKX3F5Y0iCfbqezoAX7J64EB6w1Fe0zf6HP5ZEzidvL-bokljd2JPxrPZvpaV93maYw  J’en ai profité également pour concevoir le code afin que seuls les utilisateurs inscrits et connectés puissent poster. Sinon, je les invite à s’inscrire afin qu’ils aient cette possibilité.  Voici donc le résultat :  https://lh5.googleusercontent.com/qYoQ_aTsAZby4ynbSnh_univW-uwK8SDyl_hZ7O-kPHC_AZMA8gyGsGiz19EyTX0ZQs3Obq4SKnc-Z8MSZefGsl49stbvHRPBEhCLhZnw6p_VuzzuvQxRfOPP_xBFO0RkJ80Xde2ZKoxKfk6D2BYCmM  https://lh4.googleusercontent.com/KEkKuYtl-mdW6cgR8ourEjqFUWhWA988gZnOFmUwXT-De5cnGJ5hUAzXqXwtndpAi3gURgxlQb1fYdHjmr6csw38pHNiqMfA8fhd6hiQE3o7DdrdbCmthcUSJqb_qkJEos-DFHd2yYA4FEQwHF6ABi8  https://lh6.googleusercontent.com/L1IIsHdE7GKliN_yRtQjepe5TyV5YwLIdgTqzh__udoqMOOikRXavpOP-kShVWJpJY8ad28hkkLVdVG0cDh07ST1vMT4dVHxkkiFV2-FXaQHIFF37T3naCd5t0j405iaWoP6Wg5kGQXydkQ9MSAPbY8    Résultat dans la base de données :  https://lh6.googleusercontent.com/TEY_mGpoqt05q7C_46zDqcE46b2pJJrMj-9jT42ryGWHdK5loDJgqWt3YAZR4iZonlvs56Uk8h7PqjjH3KHii4yxY7QDsel9lvUPrCGz91tFrvgL177Gxpsmxx1qqwpNAmkCuoDeDe5JV7BA4DEWFT0 | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **2. Précisez les moyens utilisés :** | | | | | |
|  | | | | | |
| PHP, Requêtes SQL, HTML, CSS, JavaScript. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **3. Avec qui avez-vous travaillé ?** | | | | | |
|  | | | | | |
| Seul | | | | | |
|  | | | | | |
|  | | | | |
|  | | | | | |
| **4. Contexte** | | | | | |
|  | | | |  | |
| Nom de l’entreprise, organisme ou association | | | | *AFCI* | |
| Chantier, atelier, service | | | Atelier | | |
| Période d’exerciceDu : *06/03/2023*au :*02/06/2023* | | | | | |
|  | | | | |
|  | | | | |
| **5. Informations complémentaires** *(facultatif)* | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |

|  |  |  |
| --- | --- | --- |
| **Titres, diplômes, CQP, attestations de formation** | | |
|  | | |
| *(facultatif)* | | |
| **Intitulé** | **Autorité ou organisme** | **Date** |
| **Baccalauréat** | **Lycée Louis Bascan** | 2017 |
| **BTS MUC** | **ISIFA PLUS VALUES** | 2020 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

Walid Abdoun

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.

Dunkerque

09/06/2023

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)* |