Modèle de copie :
Évaluation en cours de formation

Développeur Web et Web Mobile

Prénom : Michel
Nom : Almont

Lien Github du projet : https://github.com/NOA-FASHION/Trtconseil
Documentation Technique : https://github.com/NOA-FASHION/Trtconseil/tree/main/Documentation
Lien site web : https://backend-strapi.online/trt-conseil/
Attention ! Merci de bien classer vos documents dans votre Github ou votre drive.
URL du site (si vous avez mis votre projet en ligne) : ……

Description du projet

1. Liste des compétences du référentiel qui sont couvertes par le projet

- Maquetter une application.
- Développer une interface utilisateur web dynamique.
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.
- Créer une base de données.
- Développer les composants d'accès aux données.
- Développer la partie back-end d'une application web ou web mobile.
- Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

2. Résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

PROJET TRT-CONSEIL

Selon le cahier des charges l'application demandé, devra permettre aux acteurs de l'hotellerie de se confronté.
TRT Conseil est une agence de recrutement spécialisée dans l'hôtellerie et la restauration. Fondée en 2014, la société s'est agrandie au fil des ans et possède dorénavant plus de 12 centres dispersés aux quatre coins de la France.
Elle souhaite dévelloper un outil ou les recruteurs pourrons proposer des postes de travail et les candidats pourronts y postuler, tout cela sous l'administration des consultants de la boite.

Il y aura donc une interface pour chaque type d'utilisateurs.

Interface Admin
Pour accéder à l'interface d'un compte admin, il faudra obligatoirement posséder un compte avec le rôle admin.
Cette interface aura pour fonction de créer des comptes consultants.

Interface consultant
L'interface consultant aura la fonction de valider les nouveaux comptes de type recruteur et de type candidat, de valider les annonces des recruteurs et les candidatures des candidats.
Il faudra obligatoirement disposer d'un compte de type consultant que seul un administrateur peut créer.

Interface recruteur
L'interface recruteur permettra de rentrer les données de l'entreprise ainsi que de créer les annonces des postes à pourvoir.
Il donnera aussi la possibilité de voir les candidatures associées aux annonces qu'elle aura créé au préalable.
Pour accéder à cette interface, il faudra d'abord une inscription du recruteur sur le site puis une validation du compte par un consultant.

Interface candidat
L'interface candidat permettra de rentrer les données personnelles du candidat ainsi que son cv.
Il pourra aussi accéder aux annonces valider et y postuler.
Pour accéder à cette interface il faut d'abord que le candidat s'inscrive sur la plateforme puis il faudra que son compte soit activer par un consultant.

3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

Fonctionnalités de l'application

Se connecter

Utilisateurs concernés : Recruteurs, candidats, consultants, administrateurs

4 types d'utilisateur devront pouvoir se connecter :

Les recruteurs : Une entreprise qui recherche un employé.
Les candidats : Un serveur, responsable de la restauration, chef cuisinier etc.
Les consultants : Missionnés par TRT Conseil pour gérer les liaisons sur le back-office entre recruteurs et candidats.
L'administrateur : La personne en charge de la maintenance de l'application.
Chaque type d'utilisateur posséderons un compte pour accéder à leur espace personnel et respectif.

Espace spécifique

Utilisateurs concernés : Recruteurs, candidats, consultants, administrateurs

Chaque type d'utilisateur posséderons un espace type spécifique, qui sont l'espace candidat pour les candidats, l'espace partenaire pour les partenaires et l'espace consultant pour les consultants.
L'espace recruteurs donnera accès aux annonces qui lui appartiennent.
L'espace candidats donnera accès aux annonces activés.
L'espace consultants donnera accès aux annonces, aux comptes candidats et aux comptes recruteurs.
L'espace administrateur permettra la gestion des comptes de type consultants.

Créer son compte

Utilisateurs concernés : Recruteurs, candidats

Les recruteurs et les candidats auront la possibilité de créer leur compte qui sera par défaut désactivé.
Pour les comptes des consultants c'est l'administrateur qui aura le droit de les créer.

Activation des comptes

Utilisateurs concernés : consultant

Le consultant pourra activer les comptes des candidats et des recruteurs

Compléter son profil

Utilisateurs concernés : Recruteurs, candidats

Les candidats pourront préciser leur nom, prénom ainsi que transmettre leur CV (obligatoirement au format PDF).
Les recruteurs pourront préciser le nom de l'entreprise ainsi qu'une adresse.

Publier une annonce

Utilisateurs concernés : recruteurs

Un formulaire devra demander l'intitulé du poste, le lieu de travail et une description détaillée (horaires, salaire, etc.).

Pour chaque offre qu'il a transmise, une liste des candidats validés par TRT Conseil et qui ont postulés à cette annonce sera visible par le recruteur.

Postuler à une annonce
Utilisateurs concernés : candidats

Depuis la liste de toutes les annonces disponibles sur l'application, un candidat peut postuler à une offre en appuyant sur un simple bouton.
Si c'est approuvé, le recruteur concerné recevra un email avec le nom/prénom du candidat ainsi que son CV.

Activation annonce et candidature
Utilisateurs concernés : consultant

Un consultant doit valider l'annonce d'un partenaire avant qu'elle soit visible pour les candidats.
Un consultant doit approuver la candidature d'un candidat

4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Spécifications techniques du projet :

Maquette/Illustration/Logo/Image

Pour la partie maquettage j'ai utilisé **Figma**.

Diagrams.net est l'outil que j'ai utilisé pour réaliser le diagramme de classe ou, diagramme de cas d'utilisation et le diagramme de séquence

Canva.com est l'outils que j'ai utilisé pour la documentation.

Environnement de travail

L'environnement de développement tournait sous **MacOs**

Pour consulter, modifier et tester la base de données, j'ai utilisé l'IDE **Datagrip.**

Vscode est mon éditeur de texte préféré.

Technologie front-end

Pour coder rapidement une interface responsive et efficace j'ai choisi l'outils **Bootstrap** couplé au thème **Bootswatch**.

Le Template **Twig** est le moteur de gabarit que j'ai utilisé pour la gestion des vues.
Le moteur de Template open source **Twig** facilite le développement, la sécurisation et la maintenance d'applications web PHP.
JavaScript est un élément indispensable pour le bon fonctionnement de Bootstrap.

Technologie Back-end

Base de données

Le système de gestion de base de données est **POSTGRES.**
Ce système multi-plateforme est largement connu et réputé à travers le monde, notamment pour son comportement stable et pour être très respectueux des normes **ANSI SQL.**

Symfony

Le Framework PHP Symfony est le partenaire que j'ai choisi pour interfacer le SGBD POSTGRESS.

Pourquoi Symfony ?
Il contient tout ce dont a besoin pour créer un site web professionnel et sécurisé :

- un moteur de gabarit
- un ORM
- un client de test

Symfony est un Framework qu'on pourrait qualifier de modulaire.
Pour utiliser toute la puissance de ce Framework il faut installer et activer plusieurs modules indispensables pour réaliser un site web sécurisé et professionnel.
Les composants qui ont été installé et activer dans ce projet sont décrit succinctement dans les prochaines lignes.

Doctrine

Doctrine est un ORM de Symfony
Un ORM est un ensemble de classes permettant de manipuler les tables d'une base de données relationnelle comme s'il s'agissait d'objets.

En base de données chaque éléments ou objet d'une application est représenté par une table.
En programmation ces éléments sont des objets.

Un ORM est une couche d'abstraction d'accès à la base de données qui donne l'illusion de ne plus travailler avec des requêtes mais de manipuler des objets.

Mail

Symfony possède son propre composant nous permettant d'envoyer des e-mails depuis la version 4.3.C'est le composant Mailer.

L'installation du composant Mailer se fait via composer

Forms

Les formulaires sont des éléments indispensables pour récupérer les informations de nos objets.
Symfony propose un système puissant et flexible qui permet d'unifier et de simplifier la génération et le traitement de formulaires.
A lui seul, il gère les éléments suivants :

- Rendu du formulaire dans la page
- Gestion de l'envoi du formulaire
- Validation des données
- Normalisation des données
- Protection CSRF

L'installation de ce module se fait par l'intermédiaire de composer

Github

Pour la gestion du travail collaboratif, du versionning et de l'hébergement des dépôts du projet le tandem GIT/GITHUB a été utilisé.

Déploiement

Le déploiement a été effectuer sur un serveur VPS Ubuntu.
Le serveur web utilisé est l'outils NGINX

Sécurité

SecurityBundle

Authentification avec le SecurityBundle de Symfony

Le contrôle de la sécurité sous Symfony est très avancé mais également très simple. Pour cela Symfony distingue :

- L'authentification,
- L'autorisation.

L'authentification, c'est le procédé qui permet de déterminer qui est votre visiteur. Il y a deux cas possibles :

- Le visiteur est anonyme car il ne s'est pas identifié,
- Le visiteur est membre de votre site car s'est identifié

Sous Symfony, c'est le firewall qui prend en charge l'authentification.

Régler les paramètres du firewall va vous permettre de sécuriser le site. En effet, vous pouvez restreindre l'accès à certaines parties du site uniquement aux visiteurs qui sont membres. Autrement dit, il faudra que le visiteur soit authentifié pour que le firewall l'autorise à passer.

L'autorisation intervient après l'authentification. Comme son nom l'indique, c'est la procédure qui va accorder les droits d'accès à un contenu. Sous Symfony, c'est l'Access control qui prend en charge l'autorisation.

Prenons l'exemple de différentes catégories de membres. Tous les visiteurs authentifiés ont le droit de poster des messages sur le forum mais uniquement les membres administrateurs ont des droits de modération et peuvent les supprimer. C'est l'accès control qui permet de faire cela.

Sécurisation de l'environnement

Concernant la sécurité du serveur, j'ai installé le tandem Fail2ban le firewall UFW pour filtrer les paquets et limiter l'ouverture des ports du serveur.

Un pare-feu est essentiel lors de la configuration du VPS pour limiter le trafic indésirable sortant ou entrant dans votre VPS.

L'outil fail2ban permet de surveiller l'activité des logs de certains services, tel que SSH ou Apache. Lors d'un trop grand nombre d'authentifications ratées fail2ban va générer une règle IPTables, cette règle aura pour but d'interdire pendant une durée déterminée les connexions depuis l'adresse IP susceptible d'être un attaquant.

5. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

Les bonnes pratiques de sécurité d'un site PHP

Vulnérabilités PHP

La dernière étude du Laboratoire de Threat Intelligence F5 montre que PHP est le langage web où il y a le plus de vulnérabilités. En effet, sur l'ensemble du trafic malveillant en 2018, 81 % étaient liés au langage PHP. La hausse des vulnérabilités PHP est de 23 % par rapport à 2017.

Des vulnérabilités PHP à nuancer et qui peuvent être empêchées

PHP, le language web le plus utilisé

Tout d'abord, il faut savoir que PHP est utilisé pour la création de plus de 80 % des sites web mondiaux. Il est donc logique que les hackers recherchent plus facilement les vulnérabilités de ce langage car les informations à dérober sont plus nombreuses. De plus, par sa facilité de compréhension, PHP est utilisé par beaucoup de débutants en informatique. Ces derniers n'ont pas encore l'expérience nécessaire pour sécuriser les applications.

L'utilisation de Symfony pour sécuriser les applications

Le Framework permet d'empêcher aisément de nombreuses failles de sécurité : grâce à Twig, les failles XSS sont bloquées. Le principe de cette faille est d'injecter un code malveillant en langage de script dans un site web vulnérable, par exemple en déposant un message dans un forum qui redirige l'internaute vers un faux site (phishing) ou qui vole vos informations (cookies).

Un autre exemple : la gestion des formulaires de Symfony embarque un outil permettant de gérer les erreurs CRSF. Il s'agit d'effectuer une action visant un site ou une page précise en utilisant l'utilisateur comme déclencheur, sans qu'il en ait conscience.

Vulnérabilités PHP : Injection SQL

Doctrine l'ORM de symfony, permet d'empêcher nativement les failles d'injection de la plupart* des requêtes.
D'autre fonctionnalité peut être activé pour renforcer la sécurité de votre site.

- Le filtrage des urls par .htaccess
- Un firewall http
- Le changement des noms des tables. (prefixe wp_)
- L'utilisation des fonctionnalités de sécurité de Symfony

Vérification des vulnérabilités de sécurité sous symfony :

Le symfonybinaire créé lorsque vous installez Symfony CLI fournit une commande pour vérifier si les dépendances de votre projet contiennent une vulnérabilité de sécurité connue :

symfony check:security

Une bonne pratique de sécurité consiste à exécuter cette commande régulièrement pour pouvoir mettre à jour ou remplacer les dépendances compromises dès que possible.
Le contrôle de sécurité se fait localement en récupérant la base de données publique des avis de sécurité PHP, ainsi votre composer.lockfichier n'est pas envoyé sur le réseau.

6. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

Sur la fin de mon projet j'ai mis en place la fonctionnalité pour uploader un fichier PDF.
Le candidat doit au moment de compléter son profil uploader son cv au format PDF.
J'ai obtenu les informations pour coder cette fonctionnalité sur la documentation symfony qui est en anglais.

Le lien est le suivant :
https://symfony.com/doc/current/controller/upload_file.html

7. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes).

Instead of handling file uploading yourself, you may consider using the VichUploaderBundle community bundle. This bundle provides all the common operations (such as file renaming, saving and deleting) and it's tightly integrated with Doctrine ORM, MongoDB ODM, PHPCR ODM and Propel.

Imagine that you have a Product entity in your application and you want to add a PDF brochure for each product. To do so, add a new property called brochureFilename in the Product entity:

TRADUCTION

Au lieu de gérer vous-même le téléchargement de fichiers, vous pouvez envisager d'utiliser le bundle communautaire VichUploaderBundle. Cet ensemble fournit toutes les opérations courantes (telles que le changement de nom, l'enregistrement et la suppression de fichiers) et il est étroitement intégré à Doctrine ORM, MongoDB ODM, PHPCR ODM et Propel.

Imaginez que vous ayez une entité Produit dans votre application et que vous souhaitiez ajouter une brochure PDF pour chaque produit. Pour ce faire, ajoutez une nouvelle propriété appelée brochureFilename dans l'entité Product :

```
// src/Entity/Product.php
namespace App\Entity;
use Doctrine\ORM\Mapping as ORM;
class Product
{
    // ...
    #[ORM\Column(type: 'string')]
    private $brochureFilename;

    public function getBrochureFilename()
    {
        return $this->brochureFilename;
    }
    public function setBrochureFilename($brochureFilename)
    {
        $this->brochureFilename = $brochureFilename;
        return $this;
    }
}
```

Note that the type of the brochureFilename column is string instead of binary or blob because it only stores the PDF file name instead of the file contents.

The next step is to add a new field to the form that manages the Product entity. This must be a FileType field so the browsers can display the file upload widget. The trick to make it work is to add the form field as "unmapped", so Symfony doesn't try to get/set its value from the related entity:

TRADUCTION

Notez que le type de la colonne brochureFilename est une chaîne de caractère au lieu d'un fichier binaire car elle ne stocke que le nom du fichier PDF au lieu du contenu du fichier.

L'étape suivante consiste à ajouter un nouveau champ au formulaire qui gère l'entité Product. Il doit s'agir d'un champ FileType pour que les navigateurs puissent afficher le widget de téléchargement de fichiers. L'astuce pour que cela fonctionne est d'ajouter le champ de formulaire comme "non mappé", afin que Symfony n'essaie pas de définir sa valeur à partir de l'entité associée :

```
// src/Form/ProductType.php
namespace App\Form;
use App\Entity\Product;
use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\Extension\Core\Type\FileType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolver;
use Symfony\Component\Validator\Constraints\File;
class ProductType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            // ...
            ->add('brochure', FileType::class, [
                'label' => 'Brochure (PDF file)',

                // unmapped means that this field is not associated to any entity property
                'mapped' => false,
                // make it optional so you don't have to re-upload the PDF file
                // every time you edit the Product details
                'required' => false,
                // unmapped fields can't define their validation using annotations
                // in the associated entity, so you can use the PHP constraint classes
                'constraints' => [
                    new File([
                        'maxSize' => '1024k',
                        'mimeTypes' => [
                            'application/pdf',
                            'application/x-pdf',
                        ],
                        'mimeTypesMessage' => 'Please upload a valid PDF document',
                    ])
                ],
            ])
```

```
        // ...
    ;
  }

  public function configureOptions(OptionsResolver $resolver)
  {
    $resolver->setDefaults([
      'data_class' => Product::class,
    ]);
  }
}
```

Now, update the template that renders the form to display the new brochure field (the exact template code to add depends on the method used by your application to customize form rendering):

TRADUCTION

Maintenant, mettez à jour le template twig qui traduit le formulaire et affiche le nouveau champ 'brochure' (le code pour customiser votre Template dépend de la méthode utilisée par votre application) :

```
{# templates/product/new.html.twig #}
<h1>Adding a new product</h1>
{{ form_start(form) }}
  {# ... #}
  {{ form_row(form.brochure) }}
{{ form_end(form) }}
```

Finally, you need to update the code of the controller that handles the form:

TRADUCTION

Enfin, vous devez mettre à jour le code du contrôleur qui gère le formulaire :

```
// src/Controller/ProductController.php
namespace App\Controller;
use App\Entity\Product;
use App\Form\ProductType;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\File\Exception\FileException;
use Symfony\Component\HttpFoundation\File\UploadedFile;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\String\Slugger\SluggerInterface;
class ProductController extends AbstractController
{
  #[Route('/product/new', name: 'app_product_new')]
  public function new(Request $request, SluggerInterface $slugger)
  {
    $product = new Product();
    $form = $this->createForm(ProductType::class, $product);
    $form->handleRequest($request);
    if ($form->isSubmitted() && $form->isValid()) {
      /** @var UploadedFile $brochureFile */
```

```
        $brochureFile = $form->get('brochure')->getData();
        // this condition is needed because the 'brochure' field is not required
        // so the PDF file must be processed only when a file is uploaded
        if ($brochureFile) {
            $originalFilename = pathinfo($brochureFile->getClientOriginalName(), PATHINFO_FILENAME);
            // this is needed to safely include the file name as part of the URL
            $safeFilename = $slugger->slug($originalFilename);
            $newFilename = $safeFilename.'-'.uniqid().'.'.$brochureFile->guessExtension();
            // Move the file to the directory where brochures are stored
            try {
                $brochureFile->move(
                    $this->getParameter('brochures_directory'),
                    $newFilename
                );
            } catch (FileException $e) {
                // ... handle exception if something happens during file upload
            }
            // updates the 'brochureFilename' property to store the PDF file name
            // instead of its contents
            $product->setBrochureFilename($newFilename);
        }
        // ... persist the $product variable or any other work

        return $this->redirectToRoute('app_product_list');
    }
    return $this->renderForm('product/new.html.twig', [
        'form' => $form,
    ]);
  }
}
```

8. Autres ressources

Now, create the brochures_directory parameter that was used in the controller to specify the directory in which the brochures should be stored:

TRADUCTION

Maintenant, spécifier le paramètre' brochures_directory' qui a été utilisé dans le contrôleur pour spécifier le répertoire dans lequel les fichiers brochures doivent être stockées :

```
# config/services.yaml
# ...
parameters:
  brochures_directory: '%kernel.project_dir%/public/uploads/brochures'
```

9. Informations complémentaires