PROJET PORTFOLIO

Table des matières

1 L'APPLICATION ATTENDUE

Il est demandé le développement d'une application de type portfolio. Celle-ci va servir de support de présentation de votre travail lors de l'épreuve E6 et constitue l'un des deux projets de l'épreuve E4. Ce portfolio doit permettre de bénéficier d'un outil qui présente les compétences mobilisées au travers de différentes situations professionnelles rencontrées en complément d'une présentation de vous-même et de votre CV.

1.1 Notion de portfolio

Un portfolio est un dossier personnel dans lequel les acquis de formation et les acquis de l'expérience d'une personne sont définis et démontrés en vue d'une reconnaissance par un établissement d'enseignement ou un employeur. En France, on voit parfois le terme de portefeuille de compétences.

Le portfolio diffère du curriculum vitæ dans le sens où les renseignements qu'il contient sont articulés en fonction d'un objectif, par exemple une demande d'emploi, et qu'il doit aussi présenter des preuves des acquis de la personne.

Source : https://fr.wikipedia.org/wiki/Portfolio

1.2 Notion de situation professionnelle

La notion de « situation » est centrale dans le projet à réaliser. Il convient de bien comprendre de quoi il s'agit.

Une situation concerne les réalisations qui répondent à un ou des besoins exprimés par une entreprise, une personne, etc. Une situation peut être fictive ou réelle.

Les situations rencontrées lors de votre formation initiale (à l'école) sont appelées dans ce projet « situations initiales » et les situations rencontrées en stage sont appelées dans ce projet « situations de stages ».

En première année de BTS, vous avez travaillé sur différentes situations fictives (liste presque exhaustive) :

- Application web de gestion de contacts stockés en bdd (CRUD, upload d'images, accès par authentification, gestion du cache du navigateur) (travail réalisé sans POO)
- Utilisation et manipulation de contrôles winForm en C#
- Création d'un bloc note en C#
- Création d'un éditeur de texte riche (rtf) en C#
- Initiation à GIT et mise en œuvre
- Application web « Dictionnaire » : (Mise en œuvre de la POO, CRUD, requêtes asynchrones, DOM)
- ...

En seconde année de BTS :

- Gestion des paiements et des expéditions de commandes client (Mise en œuvre de la POO, CRUD, requêtes asynchrones, format JSON, DOM)
- Implémentation de contraintes de gestion dans le SGBD (les procédures stockées et les triggers)
- Gestion des accès à une base de données (les privilèges) et accès distant au serveur
- Mise en place d'un ORM dans un projet et génération de la base de données
- Réalisation d'un projet mobilisant les bases du framework Symfony (vue, modèle, contrôleur, formulaire, moteur de template, orm)
- ...

En stage, vous avez travaillé sur des situations réelles :

- C'est à vous de lister la ou les situations rencontrées (les notes de synthèse reprennent ces situations si elles ont bien été réalisées)

Maintenant vous savez ce qu'est une « situation ».

Important :

Les situations initiales de 1ère année et de 2ème année ne concernent pas seulement les matières « informatiques ». En économie-droit-management, en SI7, etc, vous avez travaillé sur des situations qui vous ont conduit à valider certaines compétences.

Il convient de les ajouter au portefeuille de compétences

1.3 Caractéristiques d'une situation

Une situation se caractérise par les éléments suivants :

- Une image d'illustration
- Un titre
- Un nom court qui sera utilisé dans le menu de l'application
- Un slug (qui va être affiché dans l'url)
- Une description (besoin exprimé, solution proposées, etc)
- Une information (que l'on peut voir comme un complément à une description)
- La liste des technologies mobilisées (IDE, framework, librairie, ...)
- La liste des compétences mises en œuvre
- Une période de réalisation
- Son statut de situation de stage ou non

2 FONCTIONNALITES

Côté back office :

- Lister toutes les situations professionnelles
- Ajouter / modifier / supprimer une situation professionnelle

Côté front office :

- Consulter une situation professionnelle
- Consulter le CV, la page d'accueil

3 PRODUCTION ATTENDUE :

- Une application fonctionnelle
- La documentation technique (exemple de contenus : diagrammes de classes, MCD, diagrammes UML autres, schémas et explications divers, liste des commits, manuel utilisateur, ...) doit être fournit dans un seul et même fichier au format PDF.

4 CAHIER DES CHARGES

Environnement technique et technologique

- L'application doit être développée avec l'IDE Phpstorm.
- Le framework PHP Symfony doit être utilisée dans sa version 5
- Les frameworks CSS et JS Bootstrap doivent être mobilisés afin de produire facilement un site adaptatif avec une mise en forme minimale de base.
- Le projet doit être versionné avec GIT
- Le SGBD utilisé doit être MariaDB. La bdd aura pour nom `NOM_portfolio`
- L'hôte vituel à utiliser aura le format suivant : `nom-portfolio`
- Le dossier contenant le projet aura pour nom `NOM_portfolio`

Accès à la base de données

- Seul un utilisateur spécifique avec un mot de passe pourra s'y connecter et l'exploiter. Il correspondra à la chaîne suivante `NOM_portfolio`

Accès au back office

- L'accès au back office doit être sécurisé. Ce point ne sera pas implémenté sauf si vous avez le temps et l'envie de le faire. La gestion de la sécurité des accès est l'un des points indispensables à connaître dans Symfony mais assez complexe à appréhender.

Architecture

L'architecture MVC doit être constamment respectée.

- Pour la couche « Model », Doctrine doit être l'ORM utilisé
- Pour la couche « View », le moteur de template Twig doit être utilisé

Formulaire

Le ou les formulaires doivent être générés à l'aide du composant Symfony dédié à cet usage :

https://symfony.com/doc/current/forms.html

Affichage

- L'intégralité du site doit être adaptatif.
- Toutes les pages doivent hériter d'un template de base qui lui-même inclut le menu
- Tous les boutons utilisés dans les formulaires ainsi que les liens de navigation devront avoir strictement le même rendu. Cela doit être réalisé grâce à une macro twig.

  Exemple de rendu :

  Lorsqu'il s'agit d'un simple lien mais qui permet de se déplacer dans le site (en dehors du menu), le lien doit être mis en forme de façon à ressembler à un bouton.

  Lorsqu'il s'agit d'un bouton qui sert à soumettre des données (pour créer ou mettre à jour des données), le bouton sera vert.

  Si le formulaire a pour objectif la suppression d'un élément, alors le bouton de soumission sera rouge.

Documentation technique

Le code doit être documenté avec les tags de « Phpdocumentor » ceci afin de permettre l'impression ultérieure de la documentation du code. https://www.phpdoc.org/

- L'architecture de l'application doit être documentée à l'aide d'un diagramme de classes. Le stéréotype `entity` sera utilisé lorsqu'une classe concerne des objets persistés en base de données. Le stéréotype `repository` sera utilisé lorsqu'une classe concerne est `repository`
- La liste de tous les commits doit être fournie une fois le projet terminé

La documentation technique (exemple de contenus : diagrammes de classes, MCD, diagrammes UML autres, schémas et explications divers, liste des commits, manuel utilisateur, ...) doit être fournit dans un seul et même fichier au format PDF.

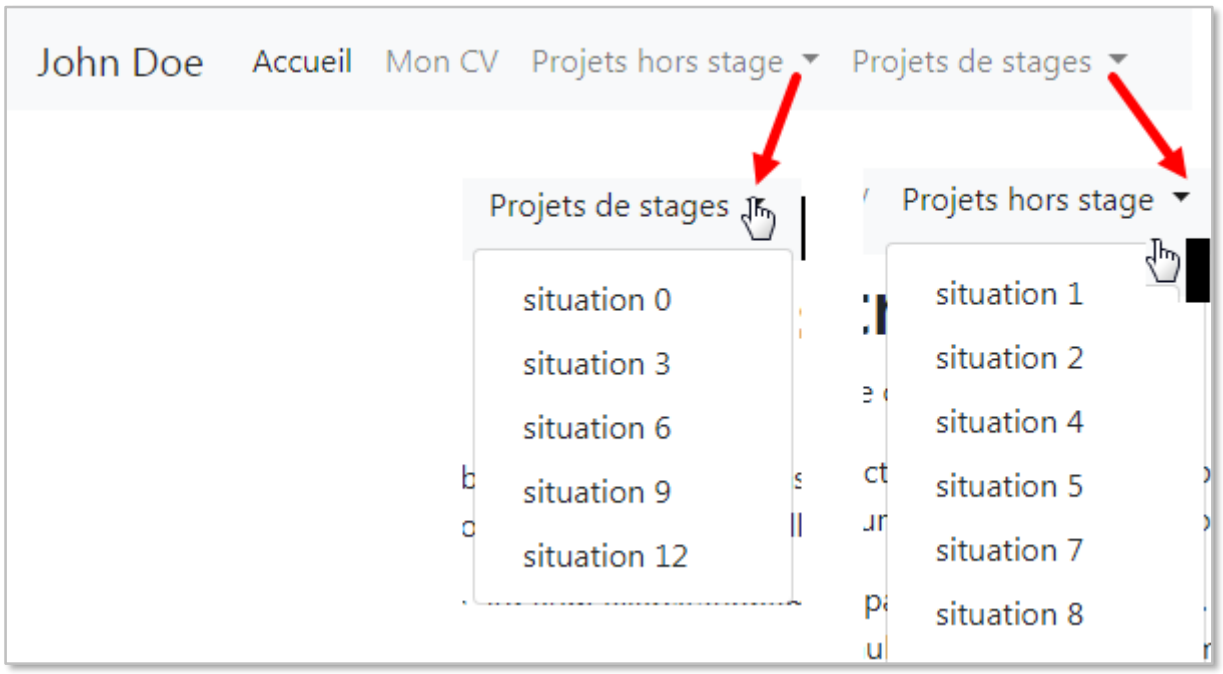
Dépendances du projet

- Pour les dépendances du projet PHP, Composer doit être utilisé afin de permettre la mise à jour des paquets et dépendances de l'application

Navigabilité

Le menu (unique) doit proposer les items navigables suivants :

```
Accueil, Mon CV, Projets hors stage, Projets de stages :
```



Les situations listées sont celles qui sont persistées en base de données. Ici, leur nom n'est pas du tout significatif.

L'item de menu en gras est l'item dont la page est affichée (ici, il s'agit de l'item « Accueil ». Si la situation 6 est affichée, l'item « Projets hors stage » doit être en gras tout comme l'item « situation 6 ».

Caractéristiques techniques d'une situation

Caractéristique	obligatoire	remarques
nom de l'image qui sert d'illustration à la situation	oui	seules les images png sont autorisées. Le nom du fichier extension comprise ne doit pas dépasser 255 caractères.
titre court qui va être affiché dans le menu	oui	chaîne de 30 caractères maximum. Chaque titre court doit être unique. Ce titre est affiché dans le menu.
titre (intitulé long de la situation)	oui	Le titre doit être suffisamment explicite sans dépasser 255 caractères. Il doit être unique.
slug basé sur le titre de la situation	oui	un slug ne peut être constitué que des lettres a à z, des chiffres 0 à 9 et de tirets (-). Pas de caractères spéciaux ni de lettres avec accents. C'est une technique très utilisée pour écrire de belles url. Chaque slug doit être unique car une situation doit pouvoir être identifiée à partir de son slug. Le slug doit être généré automatiquement au moment d'enregistrer ou de mettre à jour une situation professionnelle. Une écouteur (listener) doit être utilisé pour gérer cela. https://www.doctrine-project.org/projects/doctrine-orm/en/latest/reference/events.html#entity-listeners-class

<table>
<tr><td></td><td></td><td>Attention : il faut comprendre que des titres différents peuvent conduire à générer le même slug :

<table><tr><th>titre</th><th>slug généré</th></tr><tr><td>titre 1</td><td>titre-1</td></tr><tr><td>titre-1</td><td>titre-1</td></tr><tr><td>titre$$$$$ 1</td><td>titre-1</td></tr><tr><td>titre 1</td><td>titre-1</td></tr></table>Si chaque titre est bien unique, les slugs ne le sont pas !
Dans ce cas, l'utilisateur doit être informé que le titre utilisé conduit à générer un slug qui existe déjà et qu'il doit modifier le titre (long) proposé</td></tr>
<tr><td>description de la situation</td><td>non</td><td>Besoin exprimé, solution proposée, etc.

Texte plus ou moins long.
Chaque nouvelle phrase est saisie sur une nouvelle ligne. L'affichage doit restituer ces retours à la ligne</td></tr>
<tr><td>informations sur la situation</td><td>non</td><td>Complément à la description.

Texte plus ou moins long.
Chaque nouvelle phrase est saisie sur une nouvelle ligne. L'affichage doit restituer ces retours à la ligne</td></tr>
<tr><td>technologies utilisées</td><td>non</td><td>Texte plus ou moins long dans lequel sont énumérées les technologies mobilisées.
Chaque nouvelle technologie est saisie sur une nouvelle ligne. L'affichage doit restituer ces retours à la ligne.
(il n'est donc pas attendu une entité spécifique pour gérer les technologies)</td></tr>
<tr><td>compétences mises en œuvre</td><td>non</td><td>Texte plus ou moins long dans lequel sont énumérées les compétences mobilisées.
Chaque nouvelle compétence est saisie sur une nouvelle ligne dans la même zone de saisie. L'affichage des compétences d'une situation doit restituer ces retours à la ligne. (il n'est donc pas attendu une entité spécifique pour gérer les compétences)</td></tr>
<tr><td>date de début de la période de réalisation</td><td>non</td><td rowspan="2">Il ne peut y avoir de date de fin s'il n'y a pas de date de début. Il peut y avoir une date de début sans date de fin. La date de fin ne peut pas être antérieure à la date de début.
Cela doit être géré par une contrainte de validation personnalisée.
La date doit être saisie dans un input classique (sans liste déroulante). Pour cela, voir la documentation suivante :
https://symfony.com/doc/current/reference/forms/types/date.html#rendering-a-single-html5-text-box</td></tr>
<tr><td>date de fin de période de réalisation</td><td>non</td></tr>
<tr><td>statut indiquant si la situation est une situation de stage ou non</td><td>oui</td><td>par défaut, une situation n'est pas une situation de stage.</td></tr>
</table>

Les contraintes de validation seront définies avec des annotations directement dans les entités :

https://symfony.com/doc/current/reference/constraints.html

Attention, il existe des annotations à spécifier au niveau d'une propriété et des annotations à spécifier au niveau de la classe. Si un besoin de validation n'est pas couvert par les contraintes proposées par Symfony, il vous appartient de créer une contrainte de validation personnalisée.

Format des urls

Les urls doivent être les suivantes en fonction des pages affichées :

Page affichée	url
Page d'accueil	NOM_portfolio/
Page du CV	NOM_portfolio/mon-cv
Page d'une situation professionnelle	NOM_portfolio/9/creation-site-internet-lycee-paul (avec 9 le numéro de la situation suivi du slug correspondant au titre de la situation)

Design

Le design est libre mais doit mettre en valeur le portfolio