

Semestre : 1 ☒ 2 ☐

Session : COVID

Module : Technologies web 2.0

Enseignants : UP Web

Classes : 3A28->3A49

Documents autorisés : OUI ☐ NON ☒

Nombre de pages : 5

Internet autorisée : OUI ☐ NON ☒

Date : 16/03/2022

Heure : 15h

Durée : 1h30

Partie 1 : QCM : (8points)

**NB : Une seule réponse est correcte**

1. L'objectif d'un contrôleur dans un projet Symfony est de renvoyer un objet :
  - A. Response
  - B. Request
  - C. Interface
  - D. View
2. Le Composer dans un projet Symfony, permet de :
  - A. Installer les librairies
  - B. Mettre à jour les librairies
  - C. Déclarer des librairies
  - D. Toutes les réponses sont correctes
3. Quelle est la commande Symfony qui permet de créer une base de données ?
  - A. php bin/console make:database
  - B. php bin/console doctrine:create:database
  - C. php bin/console doctrine:database:create
  - D. php bin/console doctrine:make:database
4. Le contrôleur frontal d'une application symfony est :
  - A. Index.php
  - B. Kernel.php
  - C. Preload.php
  - D. Bootstrap.php

5. Lors de l'affichage des produits au niveau du fichier showproduct.html.twig, on a obtenu cette erreur.

Neither the property "prix" nor one of the methods "prix()", "getprix()"/"isprix()"/"hasprix()" or "\_\_call()" exist and have public access in class "App\Entity\Product".

Cette erreur est due à un problème au niveau du :

- A. Contrôleur
  - B. Formulaire
  - C. Fichier twig
  - D. Entité
6. L'Objet Relational Mapping permet de :
- A. Augmenter la quantité du code
  - B. Interdire le changement de SGBDR
  - C. Créer une indépendance du code vis-à-vis du SGBDR utilisé
  - D. Déclarer une association entre la classe et la table
7. On peut effectuer le mapping au niveau de l'ORM grâce à :
- A. yml
  - B. Annotations
  - C. xml
  - D. Toutes les réponses sont correctes
8. Un formulaire Symfony :
- A. Est l'image d'un objet existant
  - B. Sert à alimenter un objet
  - C. Sert à récupérer des informations
  - D. Toutes les réponses sont correctes

## Partie II : Etude de Cas (12 points)

Nous proposons de créer une application Web pour le département de Ressources Humaines d'une Société, développée avec Symfony 4 et les données seront enregistrées dans une base de données de type MySQL.



- L'expérience est un entier indiquant le nombre d'années d'expériences.
- La spécialité peut être soit « Informatique » soit « Reseaux »

1. Compléter le code pour que l'attribut experience puisse être nul (0.5pts)

```
/**
 * @ORM\Column(type="string",length=255,[1])
 */
private $experience;
```

2. Complétez le code nécessaire pour afficher seulement les candidats avec spécialité Informatique. (1.25pts)

```
public function read(): [2]
{
    $candidats = $this->getDoctrine()->getRepository([3]::class)
        ->[4] ([5]=> [6]);
    return $this->render('candidat/read.html.twig', ['candidats'
=> $candidats]);
}
```

3. Complétez le code qui permet d'afficher les candidats comme le montre la figure suivante. (1.5pts)

## Liste des Candidats

Cin	Nom - Prenom	Experience	Specialite
25415	ALI -SALAH	5	Informatique
25478	ARROUCH -LAZHAR	21	Réseaux
154775	MED -MOUNIR	14	Informatique
455745	SAMI -SAMI	12	Informatique

```
<h1> Liste des Candidats</h1>
<table border="1">
    <tr>
        <th>Cin</th>
        <th>Nom - Prenom </th>
        <th>Experience</th>
        <th>Specialite</th>
    </tr>
    [7]
    <tr>
        [8]
    </tr>
    [7]
</table>
```

**NB : Le nom et le prénom doivent être affichées en majuscule.**

4. Pour chaque candidat, on souhaite ajouter un lien « planifier » comme le montre la figure suivante :

## Liste des Candidats

Cin	Nom - Prenom	Experience	Specialite	Action
25415	ALI -SALAH	5	Informatique	<a href="#">Planifier</a>
25478	ARROUCH -LAZHAR	21	Réseaux	<a href="#">Planifier</a>
154775	MED -MOUNIR	14	Informatique	<a href="#">Planifier</a>
455745	SAMI -SAMI	12	Informatique	<a href="#">Planifier</a>

Complétez le code qui permet d'ajouter le lien hypertexte, sachant que le lien nous redirige vers la route nommée « planifiercandidat » et qui prend en paramètre l'id du candidat. (1.25pts).

```
<a href="{{[9]([10] , [11])}}">Planifier</a>
```

5. En Cliquant sur le lien « planifier », le responsable va être redirigé vers l'interface de planification d'entretien comme la montre la figure suivante :

### Planifier un entretien à Ali salah

Type ☐ Technique N1 ☐ Technique N2 ☐ Ressources Humaine

Date

Heure  :

- a. Complétez le code pour qu'un seul type puisse être choisi, et que l'affichage soit sous forme de bouton radio (1.5 pts)

```
->add('type', [12]::class, [
  'choices' => [
    'Technique N1' => "TechniqueN1",
    'Technique N2' => "TechniqueN2",
    'Ressouces Humaine' => "Ressouces Humaine",
  ],
  [13]=> [14],
  [15] => [16]
])
```

- b. Complétez le code de la fonction qui permet de planifier un entretien à un candidat. (3.5pts)

```
/**
 * @Route("[17]" , name =[18]) */
public function add(Request $request, $id , [20]$rep)
{
    $e = [21] ;
    $c = $rep->[22] ;
    $e->setCandidat($c);
    $e->[23];
    $form = $this->createForm(EntretienType::class, $e);
    $form->handleRequest($request);
    if ($form->isSubmitted()) {
        $em = $this->getDoctrine()->getManager();
        $em->persist($e);
        $em->flush();
    }
    return $this->render("add.html.twig",
        ["form" => $form->[24], "c"=>$c]);
}
```

6. Pour chaque candidat, on veut afficher les entretiens qui lui sont planifiés comme le montre la figure suivante :

## Les entretiens programés pour le candidat Ali

Type	Date	Etat
Technique N1	20/10/2022	planned
Technique N2	20/01/2023	planned

Complétez la fonction Query Builder qui permet d'afficher les entretiens spécifiques à un candidat (2.5pts)

```
public function listByCandidat($id)
{
    return $this->[25] ('e')
        ->[26] ('e.candidat', 'c')
        ->[27] ('c')
        ->where('c.id=:id')
        ->setParameter('id',$id)
        ->[28]->[29];
}
```

**Bon travail**