

Module : **Technologies Web**

Enseignants : **Equipe Web**

Classes : 3A19->3A24

Documents autorisés : OUI ☒ NON ☐

Calculatrice autorisée : OUI ☐ NON ☒

Date : 09/01/2018

Heure: 13h.30

Nombre de pages : 4

Internet autorisé : OUI ☐ NON ☒

Durée : 1h30

Problème :

Une école doctorale souhaite gérer les sujets de thèse ainsi que les différents candidats. Nous proposons alors de créer une application Symfony pour gérer l'affectation des sujets.

Travail demandé :

Soit le diagramme des classes suivant :



Figure 1 : Diagramme de classes

- 1- Créer le bundle « **TheseBundle** » (0.5 pt)
- 2- Créer les entités « **sujet** » et « **candidat** » qui correspondent aux classes de la **Figure 1.** (2 pts)
- 3- Créer la base puis générer le schéma correspondant. (0.5 pt)
- 4- Remplir la table « **sujet** » avec des valeurs au choix. Voir **Figure 2** (0.5 pt)

reference	titre	description	nbpoints
C785696321	La réactivité des hydrazones revisitée	Les hydrazines et leurs dérivés sont une classe im...	15
R147852369	Réseaux Hybrides pour l'IoT	Réseaux: Internet of Things ou internet des objets...	25
R741852745	Optimisation des ressources des réseaux IoT	Les réseaux IoT ont la particularité de mettre en ...	27

Figure 2 : Contenu de la table «sujet»

- 5- Créer l'interface d'ajout d'un candidat comme indiqué dans la **Figure 3. (1 pt)**

Cin	<input type="text"/>
Nom	<input type="text"/>
Prenom	<input type="text"/>
Score	<input type="text"/>
Mot Clé	<input type="text"/>
<input type="button" value="Ajouter"/>	

Figure 3: Formulaire d'ajout d'un candidat

- 6- En cliquant sur le bouton « **Ajouter** »,
a. les données saisies dans le formulaire sont insérées dans la table « **candidat** ». (2.5 pts)
b. Une redirection se fera vers la page « *afficher.html.twig* » que vous devez créer et qui affiche la liste des candidats. Voir **Figure 4. (2.5 pts)**

Said-Azer

- Cin: 12345678
- Score: 24
- Mot clé: réseau

[Simulation](#)

Asmi-Asma

- Cin: 98765432
- Score: 20
- Mot clé: Chimie

[Simulation](#)

Figure 4: Liste des candidats

- 7- Ajouter un lien pour chaque candidat appelé « **Simulation** » (Voir **Figure 4**).
- 8- En cliquant sur le lien « **Simulation** » d'un candidat, la liste des sujets adéquats à ce candidat apparaît (Voir **Figure 5 + Annexe 1**). Ecrire le code nécessaire pour ce traitement sachant que les sujets adéquats à un candidat sont les sujets où :
- le nombre de points d'un sujet – score candidat inférieur ou égal à 3
 - le mot clé du candidat apparaît dans la description du sujet.
- (4 pts)

Liste des sujets adéquats au candidat : Said Azer

- R147852369 - Réseaux Hybrides pour l'IoT
- R741852745 - Optimisation des ressources des réseaux IoT

Associer un sujet

Figure 5: Liste des sujets par simulation pour le candidat Said Azer

9- Ajouter un bouton « Associer un sujet » comme indiqué dans la **Figure 5**. (0.5 pt)

10- En cliquant sur ce bouton et en utilisant **jQuery** : (Voir **Figure 6** + **Annexe 2**)

- un lien « affecter » apparaît devant chaque sujet de la liste
- le bouton « Associer sujet » disparaît
- un nouveau bouton « Cacher les liens » apparaît.

(1,5 pts)

Liste des sujets adéquats au candidat : Said Azer

- R147852369 - Réseaux Hybrides pour l'IoT [Affecter](#)
- R741852745 - Optimisation des ressources des réseaux IoT [Affecter](#)

Cacher les liens

Figure 6: Actions jQuery suite au click sur « Associer un sujet »

11- En cliquant sur le bouton « Cacher les liens » est en utilisant **jQuery**, la liste des sujets reprend l'état initial c'est-à-dire:

- Les liens deviennent invisibles
- Le bouton « Cacher les liens » disparaît et le bouton « Associer un sujet apparaît ».

(1,5 pts)

12- En cliquant sur le lien « Affecter », le sujet correspondant est associé au candidat en question et une redirection se fait vers la liste des candidats. Voir **Figure 7** (3 pts)

- Said-Azer
- Cin: 12345678
 - Score: 24
 - Mot clé: réseau
 - Sujet: Réseaux Hybrides pour l'IoT [Simulation](#)
- Asmi-Asma
- Cin: 98765432
 - Score: 20
 - Mot clé: Chimie
 - Sujet: [Simulation](#)

Figure 7

Annexe 1 : pour envoyer plusieurs paramètres dans l'URL :

Routing.yml => path : /nomPath/{param1}/{param2}/.../{paramn}

Twig => <a href={{
path('route',{'param1':val1,'param2':val2,...,'paramn':valn})
}}

Annexe 2 : Pour insérer la bibliothèque **jQuery** placé sous le dossier « **/web** » du projet, il faut ajouter la ligne suivante:

<script src="{{ asset('jquery.min.js') }}"></script>