

Projet ITI4

Objectif du projet

L'objectif du projet est de mettre en pratiques les compétences développées dans les cours de DevOps, de Développement WEB et d'Agilité avec SCRUM. Le sujet est libre. Les seules limites sont celles de la légalité et du bon goût.

Organisation

Groupe de travail

Le projet est à réaliser par groupe de 4.
La constitution des groupes est libre.

Répartition des séances

- **Sprint 1** : du 05/11 au 18/11 avec des séances encadrées :
 - lundi 05 novembre 08:00 à 12:45
 - mercredi 07 novembre de 08:00 à 12:45
 - mercredi 14 novembre de 08:00 à 12:45
- **Sprint 2** : du 19/11 au 25/11 avec des séances encadrées :
 - lundi 19 novembre 08:00 à 12:45
 - mardi 20 novembre de 08:00 à 12:45
 - mercredi 21 novembre de 08:00 à 12:45
- **Sprint 3** : du lundi 26/11 au 02/12 avec des séances encadrées :
 - lundi 26 novembre 08:00 à 12:45
 - mardi 27 novembre de 09:45 à 12:45
 - mercredi 28 novembre de 08:00 à 12:45
- **Présentation finale** :
 - lundi 03 décembre de 8:00 à 12:45
 - mercredi 05 décembre de 08:00 à 12:45

Consignes

Fonctionnel

Le site final doit contenir (à minima) :

- Un système d'authentification qui contient :
 - Une fonctionnalité d'authentification
 - Une fonctionnalité de changement du mot de passe pour l'utilisateur
 - Une partie de l'application accessible uniquement pour les utilisateurs authentifiés
 - Une fonctionnalité pour se déconnecter de l'application
- Un système de gestion des utilisateurs (accessible uniquement pour les administrateurs) qui contient :
 - Une fonctionnalité pour lister la liste des utilisateurs
 - Une fonctionnalité pour la création d'un utilisateur
 - Une fonctionnalité pour la suppression d'un utilisateur avec la contrainte que l'utilisateur courant ne peut pas être supprimé
 - Une fonctionnalité pour définir le rôle de l'utilisateur
- Une gestion du métier de votre application (sujet libre) qui contient (à minima)
 - Une fonctionnalité pour lister des éléments
 - Une fonctionnalité de recherche d'éléments
 - Une fonctionnalité pour ajouter un élément respectant les contraintes suivantes :
 - Validation des données côté client
 - 4 types de champs (<input>) différent
 - Une fonctionnalité pour supprimer un élément
 - Une fonctionnalité pour modifier un élément (avec vérification de la cohérence des données saisies avant envoi au serveur)

Un utilisateur doit avoir à minima :

- Un login
- Un mot de passe (hashé en Argon2)
- Un rôle (Administrateur, Utilisateur)

Agilité avec SCRUM

La réalisation de votre projet doit mettre en oeuvre la méthodologie étudiées lors du cours d'agilité avec SCRUM. Nous attendons donc d'avoir :

- Une mise en place des sprints (cf Organisation / Répartition des séances) ;
- Le découpage du projet en stories avec les actions :
 - de priorisation des stories ;
 - d'estimation (en charge) pour chaque story ;
 - d'associer les stories aux différents sprints ;

Développement Web

La réalisation de votre projet doit mettre en oeuvre les différentes technologies étudiées lors du cours de développement web. Nous attendons donc d'avoir :

- du **HTML5** et du **CSS3** pour la réalisation de l'interface ;
- du **Javascript** pour réaliser les actions :
 - de manipulation du DOM ;
 - de gestion des évènements ;
 - de requêtes AJAX ;
- du code **JAVA** pour réaliser les actions :
 - de génération des pages HTML via **Thymeleaf** ;
 - de communication avec la base de données via **JDBC** ;
 - d'interaction avec les requêtes AJAX via les **servlets** ;

L'architecture du projet doit se composer :

- d'un apache tomcat pour déployer l'application web
- d'une base de données MariaDB pour gérer les données de l'application

La base de données doit être composé de :

- Au moins 1 table pour la gestion des utilisateurs ;
- Au moins 2 tables pour la gestion du métier de l'application avec une relation entre les tables (de type clé étrangère)

Vous ne devez pas utiliser de framework CSS tels que Bootstrap qui permettent de faire du positionnement sans coder de CSS. Des bibliothèques spécifiques de type FontAwesome sont néanmoins autorisées, n'hésitez pas à demander si vous avez un doute.

DevOps

La réalisation de votre projet doit mettre en oeuvre les différentes technologies étudiées lors du cours de DevOps. Nous attendons donc d'avoir :

- Une gestion des logs via **LogBack** ;
- Une construction du projet via **Maven** ;
- Une gestion du code source via **Git** (Gitlab) ;
- Des tests unitaires sur l'ensemble des services. *La couverture de code des services doit être supérieur à 90%*. Les tests unitaires doivent utiliser les technologies suivantes.
 - **JUnit** ;
 - **AssertJ** ;
 - **Mockito** ;
- Une *intégration continue* via **Gitlab-ci**

Conseils

Design du site

La beauté graphique de votre application ne rentre pas dans la notation. Néanmoins, l'originalité sera appréciée par le professeur qui corrige son 20ème application. Si vous avez des doutes sur vos capacités de designer web, restez dans la sobriété.

Charge de travail

Vous pouvez faire ce que vous voulez mais restez raisonnable. La qualité sera préférée à la quantité. De même, si votre site présente 50 pages avec des fonctionnalités similaires, cela n'a pas un grand intérêt pédagogique.

Qualité

Plus votre code est lisible, mieux c'est. Donc, faites attention à la qualité de votre code :

- Indentation du code ;
- Noms des variables, des méthodes et des classes ;
- Tests unitaires ;
- Contrôle des données : l'internaute peut saisir n'importe quoi dans les champs.

Code HTML

Certaines erreurs reviennent souvent dans le code HTML :

- N'abusez pas de la balise **
**, elle est faite pour passer à la ligne au sein d'un paragraphe. Elle ne doit pas être utilisée pour « faire de l'espace » entre les éléments de votre page.
- La balise **<table>** sert à afficher un tableau dans votre page HTML. Si vous utilisez une balise **<table>** mais que vous n'affichez pas un tableau sur votre page, c'est qu'il ne fallait sans doute pas utiliser cette balise.
- Les balises ****, **<i>** et **<u>** sont toujours interprétées par les navigateurs mais ne devraient pas être utilisées. Préférez utiliser ****, **** ou **** avec des styles CSS appropriés.

Rendu

Démonstration de fin de sprint

A chaque fin de sprint, une démonstration rapide (05 à 10 minutes) de ce qui a été réalisée pendant le sprint est à réaliser.

Présentation des stories du sprint

Au début de chaque sprint, un document récapitulatif des stories qui seront à réaliser pendant le sprint est à fournir.

Pour chacune des stories les informations suivantes sont à renseigner :

- Description
- Personne en charge
- Nombre de points
- Priorité

Présentation finale

Ce rendu a pour objectif de présenter le travail réalisé dans le cadre du projet. Elle doit être sous la forme de présentation powerpoint et doit durer une quinzaine de minutes.

Les sujets suivants doivent être abordés au cours de la présentation :

- But de l'application qui a été réalisée (Besoin)
- Fonctionnel qui a été mis en place pour répondre au besoin
- Architecture de votre application
- Organisation au sein de l'équipe
- Bilan de chaque Sprint et actions en conséquences
- Difficultés rencontrées
- Démonstration

Code source de l'application

A la fin du projet, le code source doit être rendu sous la forme d'un lien Gitlab qui doit être fournis aux encadrants. Le dépôt Git doit contenir :

- Le code source
- Un script d'initialisation pour la base de donnée
- Votre fichier de configuration pour la mise en place de l'intégration continu

La dockerisation de votre application est un plus qui sera apprécié à sa juste valeur :-)

Notation

La note finale est décomposée de la façon suivante :

- 20% sur le respect des règles fonctionnelles
- 20% sur la présentation finale
- 10% sur le suivi de la méthode agile pendant la réalisation du projet
- 50 % sur le code (code propre, respect des exigences de tests et des technos)