HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

STI

Sécurité des Technologies Internet

Projet 1

Application de messagerie

Aspect fonctionnel

Professeur
Abraham Rubinstein
abraham.rubinstein@heig-vd.ch

Assistant
Stéphane Teixeira Carvalho
stephane.teixeiracarvalho@heig-vd.ch

septembre 2021 – février 2022

Nom, prénom : _____

Nom, prénom : _____

Table des matières

1. Objectif

L'objectif de ce projet est de s'assurer que l'étudiant ait les connaissances en développement Web afin d'assimiler les connaissances en sécurité s'y rapportant. Ce projet consiste donc pour les étudiants à développer par eux-mêmes une application Web **très simple**.

Il s'agit de concevoir une application Web permettant, dans le cadre d'une entreprise, d'envoyer des messages de texte entre les collaborateurs. **Aucun protocole de communication devra être implémenté**. Les messages sont tout simplement échangés utilisant une base de données SQLite.

Ce travail sera réalisé par **groupes de 2 étudiants**.

2. Technologies à utiliser

- Docker
- PHP ou le langage de votre choix (PHP **hautement conseillé**… vous pouvez choisir des vieilles versions de PHP/autres pour simuler des conditions adverses aux développeurs)
- SQLite (éventuellement MySQL, mais SQLite est **hautement conseillé** pour simplifier le travail)
- Si d'autres librairies ou technologies doivent être utilisées, elles doivent être validées par le professeur (Bootstrap et autres sont autorisés pour **embellir** l'application. Pourtant, seul l'aspect fonctionnel aura un impact sur la note). En principe, aucun framework pour gérer l'authentification ni le contrôle d'accès ne sera autorisé.

3. Rendu

3.1. A rendre (sur un repo GitHub)

- Code de l'application (fichiers source, html, php, images, etc.)
- Base de données (si nécessaire), fichier SQLite ou dump
- Scripts ou tout autre moyen pour aider au déploiement et start-up de l'application. Si des librairies ou autres ressources doivent être installés, votre projet doit automatiser ce processus
- Manuel permettant l'installation/lancement/utilisation de l'application sur Docker (un simple README peut suffire). Ce manuel ne doit pas contenir une longue liste de libs à installer et de modifications manuelles à faire sur la base de données, etc. Votre projet doit être « clés en main » ; prêt à être utilisé

3.2. Échéance

L'ensemble du travail doit être rendu au plus tard le **Jeudi 14 octobre 2021 à 23h59**. Le rendu se fera à l'aide d'un email destiné au professeur et à l'assistant indiquant le repo Github

4. Cahier des charges

4.1. Définition globale

L'application doit permettre la mise en œuvre d'une messagerie électronique au sein d'une entreprise. Cette messagerie sera une application Web uniquement se basant sur une base de données (pas de SMTP ou autres).
La personne chargée du développement de l'application a juste reçu une liste de fonctionnalités et peu de temps pour la réalisation. Le budget ne permet donc de réfléchir sur la sécurité. Il est

attendu qu'une deuxième équipe évalue la sécurité de l'application et la modifie quelques mois plus tard.
Même l'environnement dans lequel l'application tourne risque d'être mal sécurisé.

4.2. Authentification

Une authentification simple sera nécessaire afin d'accéder à l'application. Vous devez implémenter vous-même votre propre système d'authentification.

- **Vous ne pouvez en aucun cas utiliser des modules/paquets/frameworks/etc vous permettant de simplifier la gestion d'accès et l'authentification.**
- **Seule la page de login sera accessible sans être authentifié**.

4.3. Rôles et authentification

L'application devra proposer deux rôles différents :

- Collaborateur,
- Administrateur.

Un mécanisme d'authentification simple (utilisateur – mot de passe) devra permettre d'accéder aux fonctionnalités. Pour pouvoir se connecter, un utilisateur devra être défini comme « actif ».
Les fonctionnalités détaillées pour chaque rôle sont définies plus loin dans ce document.

4.4. Navigation

Il devra être aisé de naviguer d'une page à l'autre, via des liens ou boutons.
Vous devrez implémenter votre propre système de navigation. **Vous ne pouvez en aucun cas utiliser des modules/paquets/frameworks/etc vous permettant de simplifier la navigation.**

4.5. Fonctionnalités

Un **collaborateur** aura accès aux fonctions suivantes :

- Lecture des messages reçus : une liste, triée par date de réception, affichera les informations suivantes :
  - Date de réception
  - Expéditeur
  - Sujet
  - Bouton ou lien permettant la réponse au message
  - Bouton ou lien permettant la suppression du message
  - Bouton ou lien permettant d'ouvrir les détails du message
    - Devra permettre l'affichage des mêmes informations/options que ci-dessus, avec le corps du message en plus
- Ecrire un nouveau message : rédaction d'un nouveau message à l'attention d'un autre utilisateur. Les informations suivantes devront être fournies :
  - Destinataire (unique)
  - Sujet
  - Corps du message
- Changement du mot de passe : afin de pouvoir modifier son propre mot de passe

Un **administrateur** aura accès aux fonctions suivantes :

- Doit avoir les mêmes fonctionnalités qu'un Collaborateur, en plus des suivantes
- Ajout / Modification / Suppression d'un utilisateur : un utilisateur est représenté par :
  - Un login (non modifiable)
  - Un mot de passe (modifiable)
  - Une validité (boolean, modifiable), actif ou inactif

- Un rôle (modifiable)

5. Plateforme

Vous devez préparer votre image Docker avec toutes les dépendances nécessaires. Il faudra documenter exactement comment la déployer et l'utiliser.

5.1. Image Docker

Si vous voulez vous épargner le travail de créer votre propre image Docker, une image a déjà été préparée. Elle contient un serveur Nginx, PHP et SQLite. Vous pouvez l'utiliser, surtout parce qu'elle utilise de vieilles versions de logiciels, ce qui peut potentiellement rendre l'application plus vulnérable. En pratique, il existe un nombre assez important de serveurs qui tournent sur internet sans mises à jour depuis des années !

Si vous décidez d'utiliser l'image fournie, la commande suivante télécharge l'image, lance un conteneur nommé `sti_project`, relie de manière dynamique le répertoire « `site` » local avec « `/usr/shar/nginx` » dans l'image, relie le port 8080 de votre ordinateur hôte vers le port 80 sur votre conteneur et renomme l'hôte virtuel du conteneur comme « sti ». Le conteneur est lancé en mode daemon :

```
docker run -ti -v "$PWD/site":/usr/share/nginx/ -d -p 8080:80 --name
sti_project --hostname sti arubinst/sti:project2018
```

Attention : certaines options peuvent varier en fonction de votre OS hôte. Référez-vous à la documentation.

Vous pouvez changer le port 8080 par un autre port s'il est déjà occupé par un service tournant sur votre machine.

Le répertoire `site` doit contenir un sous-répertoire `html` pour les fichiers de votre application et un sous-répertoire `databases` pour les bases de donnés :

Ensuite, pour lancer les services web et PHP, utiliser les commandes suivantes :

```
docker exec -u root sti_project service nginx start
```

```
docker exec -u root sti_project service php5-fpm start
```

Si vous avez besoin d'utiliser le conteneur en mode interactif, vous pouvez lancer un shell avec la commande suivante :

```
docker exec -it sti_project /bin/bash
```

Attention : l'utilisateur par défaut est `labo`, mot de passe `labo`. La commande `sudo` est active pour cet utilisateur.

Pour lancer un shell directement comme root :

```
docker exec -it -u root sti_project /bin/bash
```

Pour sortir du shell :

```
exit
```

Finalement, pour arrêter le conteneur :

```
docker stop sti_project
```

6. Evaluations

Chaque rendu sera évalué sur la base des critères suivants :

- Qualité du rendu
  - Respect des consignes (délai, archives, noms des fichiers, etc)
  - Présence de tous les éléments,
  - Installation/utilisation aisée,
  - etc.
- Le manuel
  - Qualité du contenu : complet, précis
  - Qualité rédactionnelle (présentation, structure, clarté, orthographe, etc).
- Les aspects fonctionnels de l'application
  - Fonctionnalités du cahier des charges,
  - Appréciation du code.

Ces critères sont donnés à titre indicatif. Ils peuvent être modifiés (ajoutés, supprimés, modifiés) et leur pondération peut être adaptée.