Application de banque en ligne sécurisée

Cahier des charges

Vous allez mettre en place une petite application web à destination des clients et des employés d'une banque, et implémenter des fonctionnalités basiques que l'on peut trouver dans une telle application.

On part du principe que c'est le même site qui est utilisé par les différents profils d'utilisateurs : clients de la banque et employés de la banque.

La nécessité de sécuriser cette application est assez évidente et les attaques potentielles faciles à imaginer. Les actions malveillantes ou non autorisées peuvent dans notre cas venir de différentes sources : un pirate complètement extérieur à l'application, un client de la banque, ou même un employé de la banque !

1. Environnement technique

L'application sera développée dans un environnement Apache/MySQL/PHP. Vos pages seront donc développées en PHP/HTML5 et votre base de données sera implémentée sous MySQL. Pour conserver l'intérêt pédagogique du devoir, vous n'utiliserez aucun framework (CSS, JavaScript, PHP).

2. Base de données

Créez une base nommée devoir-sr03 et ajoutez les tables suivantes dans cette base :

- USERS :
    - id_user
    - login
    - mot_de_passe
    - profil_user ('CLIENT', 'EMPLOYE')
    - nom
    - prenom
    - numero_compte
    - solde_compte

- MESSAGES :
    - id_msg
    - id_user_to (identifiant du destinataire du message)
    - id_user_from (identifiant de l'expéditeur du message)

- sujet_msg,
- corps_msg

3. Mettez en place les pages suivantes :

- **connexion** : elle comporte au minimum les champs 'login'et 'password' et un bouton de connexion

- **accueil** : page sur laquelle arrive un utilisateur qui a réussi à se connecter; on affiche sur cette page toutes les informations de l'utilisateur connecté; on dispose également des liens suivants sur la page : *messagerie* (pour tous les utilisateurs), *effectuer un virement* (pour tous les utilisateurs), *fiche client* (pour les employés de banque uniquement)

- **messagerie** : page qui liste tous les messages reçus par l'utilisateur connecté et qui permet de lire un message reçu ; on peut également envoyer un message depuis cette page : un employé peut envoyer un message à n'importe qui, un client ne peut envoyer un message qu'à un employé

- **virement** : page qui permet d'effectuer un virement d'un compte vers un autre; cette page reçoit en paramètre le numéro de compte qui sera débité. Attention : un client ne peut effectuer un virement que de son propre compte vers un autre compte.

- **fiche client** : page qui donne accès à liste de tous les clients; quand un client est sélectionné dans la liste, on affiche toutes ses informations ainsi qu'un lien *effectuer un virement* qui appelle la page de virement avec le numéro de compte de ce client en paramètre

4. Sécurisez votre site :

a. Testez votre site par rapport aux failles vues en cours/TD
b. Ajoutez les correctifs nécessaires en cas de faille détectée

Livrables et date limite

Date limite : 25/05/2021

Vous devez déposer sur Moodle un seul fichier zip (NomsBinomesDevoir3.zip) avec deux livrables :

- Code sources : un fichier codesource.txt qui contient un lien vers le dernier commit sur Gitlab de votre projet devoir (le contenu du commit sera constitué de vos fichiers PHP, HTML, CSS, JS, etc. en conservant l'arborescence de votre projet)

- Rapport : un fichier NomsBinomesDevoir2.pdf contenant deux sections :
  - la liste des failles que vous avez contrôlé
  - la liste des éléments et actions que vous avez mis en place pour sécuriser l'application