

Projet : Gestion d'un coffre-fort numérique

Nous souhaitons réaliser un coffre-fort numérique. L'utilisateur peut y déposer des données confidentielles (sous forme de textes ou phrases par exemple : « **Mes codes perso pour la Banque** »), mais pour y accéder, il doit avoir en sa possession 3 clés déterminées à l'avance (2 saisies dans des inputs et 1 clé générée automatiquement au clic lors de la soumission du formulaire).

Les 3 clés seront stockées en base de données.

1. Les pages du projet

Le projet est constitué de **2 pages** : (2 templates + 2 controllers)

a. Page d'ajout de données (URL à ajouter au routeur)

Une page d'ajout de données avec laquelle, grâce à un **formulaire**, le client peut **ajouter des données** avec les informations suivantes :

- Adresse email du client
- Données confidentielles (dont la saisie doit se faire à l'aide d'un champ « *textarea* »)
- Clé personnalisée numéro 1
- Clé personnalisée numéro 2

NB : Les clés personnalisées sont des **champs saisissables par le client**.

Les **informations entrées** ne doivent pas être sauvegardées dans les cookies du navigateur mais **seront enregistrées en base de données**.

Une fois que le client aura cliqué sur le bouton de soumission du formulaire « **Enregistrer** », une troisième clé aléatoire sera générée automatiquement (*nombre positif de type INT, intervalle au choix, exemple : entre 0 et 100 ou entre 1000 et 9999*) et elle sera affichée au client, avec un message lui demandant de noter cette clé.

Elle sera en effet nécessaire ultérieurement pour consulter les données du coffre-fort.

b. Page de requête (URL à ajouter au routeur)

Une page permettant de **récupérer le détail des données** précédemment enregistrées à partir des informations suivantes :

- Adresse email du client
- Clé personnalisée numéro 1
- Clé personnalisée numéro 2
- Clé personnalisée numéro 3

Si ces informations correspondent à une information en base de données, l'ensemble des informations confidentielles doivent s'afficher. Sinon, une page expliquant qu'aucune information n'a été trouvée devra être affichée au client. (*Exemple de message d'erreur : Email incorrect etc...*)

2. Contraintes techniques du projet

Vous avez des contraintes techniques à respecter dans l'implémentation du projet.

1. Le model (entité, classe) doit se nommer **Information**
2. **L'architecture MVC doit être respectée**
3. Le dossier des vues doit s'appeler **templates**
4. Le dossier des modèles doit s'appeler **Entity**

3. Conseils

Vous ne disposez pas de visuel illustrant les pages, ni d'indications supplémentaires. Il faudra concilier ce que vous avez appris en formation avec la demande ci-dessus et retrouver les mécanismes qui vous intéressent pour les implémenter. Vous ne serez pas notés sur l'aspect visuel du site internet.

Pensez à bien démarrer votre projet dans le dossier public (cd public) avant de lancer le serveur.