

## TP03 – PROFIL CLIENT

### MANDAT

Dans ce troisième travail pratique, vous devez développer les fonctionnalités pour le profil client de votre site de commerce électronique. C'est dans ce travail que l'utilisateur pourra se créer un compte, se connecter et modifier ses informations personnelles.

### LIVRABLES

Votre site web complet doit être accessible sur le serveur **techinfo** dans un sous-domaine tp03.4d6 ne doit pas être modifié entre la date de remise et la remise des notes. Le certificat SSL doit être valide dans chrome à la date maximale de remise.

Votre code source devra être remis :

- Dans un dossier compressé dans Léa **avant 23h59 le lundi 17 avril 2023**.
- Dans une branche TP03 de votre repository GitHub partagé avec l'enseignant

N'oubliez pas que votre site Web doit fonctionner avec la version 8.2 de PHP et avec la version 6.2.5 de Symfony.

### SITE DE DÉMONSTRATION

Vous pouvez regarder l'exemple disponible à l'adresse ci-dessous pour regarder les fonctionnalités<sup>1</sup>.

<https://tp03.4d6.ycharron.techinfo-cstj.ca/>

Ce site n'a pas été testé de manière rigoureuse, si vous rencontrez des problèmes ou si vous avez des suggestions, vous êtes invité à les communiquer.

---

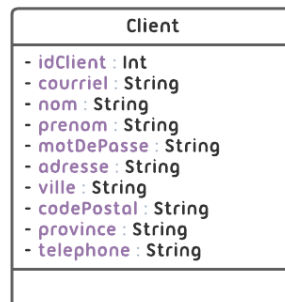
<sup>1</sup> Attention, j'ai ajouté certaines fonctionnalités que je ne vous demande pas.

## PONDÉRATION

Ce travail pratique compte pour 15 % de la note finale. Les points seront distribués en fonction des critères suivants :

Éléments de correction	%
Qualité de l'interface utilisateur	10
Configuration de la sécurité	2,5
Base de données - Clients	2,5
Classe – Client	10
Création d'un compte Client	15
Connexion d'un client	7,5
Affichage des informations du dossier	5
Modification des informations d'un compte	12,5
Modification du mot de passe d'un compte	15
Déconnexion	2,5
Déploiement sur le serveur de production	2,5
Respect du document de conception et qualité du code	10
Correction du TP02	5
Rétrocompatibilité des fonctionnalités du TP01 et du TP02	-10
Qualité de la langue	-10

## CLASSE CLIENT



Les noms des classes, des attributs, des méthodes et des routes sont à titre indicatif. Vous pouvez utiliser les noms de votre choix.

## VALIDATION DES DONNÉES

Attributs	Requis	Validation
<b>idClient</b>	Oui	Clé primaire de la table
<b>courriel</b>	Oui	Format de courriel, identifiant unique de connexion.
<b>nom</b>	Oui	Entre 2 et 30 caractères
<b>prenom</b>	Oui	Entre 2 et 30 caractères
<b>motPasse</b>	Oui	Doit être haché
<b>adresse</b>	Oui	Entre 5 et 100 caractères
<b>ville</b>	Oui	Entre 3 et 30 caractères
<b>codePostal</b>	Oui	Respectant le format canadien
<b>province</b>	Oui	2 caractères représentant les provinces et territoires canadiens <sup>2</sup>
<b>telephone</b>	Non	Obligatoire et respectant le format canadien ( <b>ex. 450-963-7453</b> )

## BASE DE DONNÉES

Une fois l'entité **Client** créée avec les validations nécessaires créer la table associée dans la base de données. Assurez-vous que les champs respectent les normes de base de données.

## SÉCURITÉ

Assurez-vous de configurer les paramètres nécessaires dans le fichier `security.yaml` pour mettre en fonction le hachage des mots de passe et la connexion utilisateur.

## INSCRIPTION

Créer un contrôleur et une vue pour permettre l'inscription d'utilisateur. La page consiste en un formulaire pour chacun des attributs de l'entité **Client**. Vous devez créer une classe de formulaire (**AbstractType**).

- La province doit être modifiable via une liste déroulante
- Le mot de passe doit être vérifié avec un second champ

Vous devez valider toutes les informations côté client et côté serveur, pour le code postal et le téléphone vous devez utiliser une bibliothèque pour réaliser des masques de saisie. Les formats des données ne doivent pas être sauvegardés en base de données.

Vous devez valider l'ensemble des données côté client et côté serveur.

<sup>2</sup> Dans un système non pédagogique, il faudrait penser faire une table et une relation entre les deux.

**ROUTE****/inscription****inscription**

Nous allons créer une nouvelle route `/inscription`, cette route doit afficher le formulaire. Voici les étapes à réaliser dans cette route :

1. Créer un formulaire
2. Si le formulaire est soumis et valide
  - a. Encoder et conserver dans l'entité le mot de passe
  - b. Sauvegarder l'entité client dans la base de données
  - c. Rediriger l'utilisateur à la page de dossier
  - d. L'utilisateur doit être authentifié après la création de son compte client
3. Il ne faut pas oublier la gestion des erreurs et les messages à afficher à l'utilisateur

**CONNEXION****VIEW**

Créer une nouvelle vue devant minimalement être composée d'un formulaire pour la connexion. La connexion doit se faire à l'aide du courriel et d'un mot de passe.

**/connexion****connexion**

Créer une route `/connexion`, celle-ci doit traiter l'authentification et afficher la vue `connexion.html.twig`.

**/deconnexion****deconnexion**

Créer une route `/deconnexion`, cette route doit être vide, elle doit seulement exister pour le fonctionnement du site.

## PROFIL

Pour permettre à l'utilisateur de modifier les informations de son profil, il sera nécessaire de créer une vue pour son profil. Cette vue devra offrir deux formulaires un pour la modification des informations du client et un deuxième spécialisé pour la modification du mot de passe.

## MODIFICATION DES INFORMATIONS

Vous devez créer une classe de formulaire (`AbstractType`) pour la modification des informations d'un client. Le formulaire doit contenir tous les champs d'un compte client à l'exception du courriel qui ne peut pas être modifié et du mot de passe qui sera modifiable dans un autre formulaire.

---

/profil

profil

Nous allons développer la route `/profil`, pour afficher les informations du compte client dans un formulaire. Voici un peu les étapes à suivre dans cette fonction.

1. Retrouver les informations du client authentifié
2. Créer le formulaire avec les informations du client
3. Si le formulaire est soumis et valide
  - a. Sauvegarder les modifications dans la base de données
  - b. Rediriger l'utilisateur vers la page de dossier

Attention, il faut rediriger l'utilisateur s'il tente d'accéder à son dossier sans être authentifié.

## MODIFICATION DU MOT DE PASSE

Vous devez ajouter un formulaire à la page du profil utilisateur pour lui permettre de modifier son mot de passe.

---

/profil

profil

Nous allons ajouter à notre route `/profil` un formulaire pour la modification de mot de passe<sup>3</sup> de l'utilisateur. Ce formulaire doit comporter trois champs :

- Le mot de passe actuel
- Le nouveau de mot de passe et une confirmation du mot de passe

Voici les étapes à suivre pour modifier le mot de passe :

1. Le mot de passe actuel doit être celui de l'utilisateur
2. Le nouveau mot de passe doit être différent du mot de passe actuel
3. Encoder le nouveau de passe
4. Sauvegarder le nouveau mot de passe

---

<sup>3</sup> Vous allez devoir prendre un peu de temps pour lire la documentation de l'interface [UserPasswordHasherInterface](#)

### INTERFACE UTILISATEUR

Faites en sorte d'ajouter les options pour l'utilisateur de se créer un compte, de se connecter et de se connecter aux endroits nécessaires.

Faites en sorte qu'il soit facile pour l'utilisateur de remarquer s'il est authentifié, par exemple dans la barre de navigation, à vous de voir.

### CONSIGNES SUPPLÉMENTAIRES

Il est important de bien valider le fonctionnement des éléments du travail pratique 1 et 2 avant de remettre votre travail.

### DÉPLOIEMENT DANS L'ENVIRONNEMENT DE PRODUCTION

Une fois toutes les fonctionnalités complétées, vous devez déployer votre site sur le serveur **techinfo** dans un sous-domaine sécurisé spécifique pour le TP03.