28/04/2022

# BE Web IENAC21 (séance 2)

***Objectifs :***

**Equipe 1 (2 personnes max) :** Authentification des utilisateurs

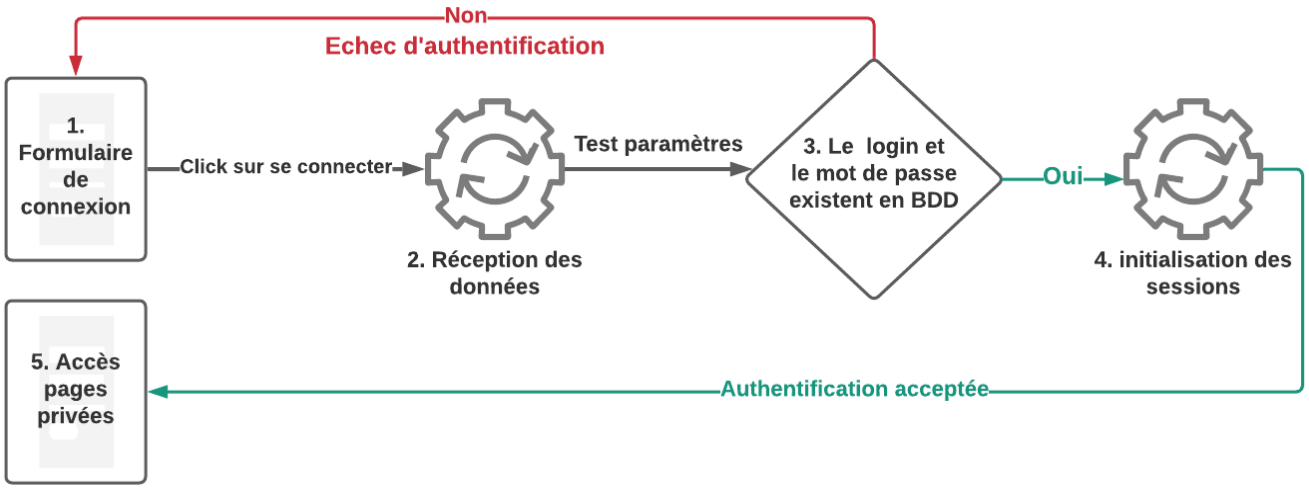
**Equipe 2 (2 personnes max) :** Création des comptes utilisateurs

# Equipe 1 :
## *Authentification des utilisateurs*

Lors de cette séance, vous allez :

- Utiliser le principe des sessions.
- Mettre en place la fonctionnalité d'authentification des utilisateurs. L'internaute authentifié pourra ainsi accéder aux parties dites « privées » du site.

***Rappel du principe d'authentification :***




## TRAVAIL A REALISER

### a. Réception des données du formulaire et création des sessions

Rappel, le formulaire d'authentification de la page login.html a été créé lors de la séance 1 du BE.

***Dans views.py :***

1) Réceptionner les données du formulaire d'authentification.
2) Appeler la fonction verifAuthData(login , motPasse) qui sera créée juste après dans bdd.py



Karine PUECHMOREL – Z134
Karine.puechmorel@enac.fr

## b. Test d'authentification

1) Importer le fichier bdd.py depuis E-campus section cours, puis enregistrez-le dans le répertoire model.

***Dans bdd.py :***

2) Créer la fonction verifAuthData(login , mot_passe)
   - Qui vérifie si le login et le mot de passe existent dans la table utilisateur - cf. *slide 135*
   - Qui renvoie un message d'échec ou de succès ainsi que les données de l'utilisateur si l'authentification a réussi.

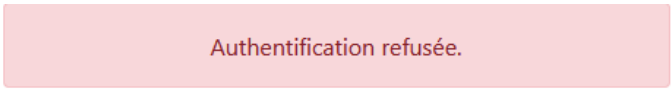
## c. Traitement de la réponse du serveur de BDD - *cf slide 135*

***Dans views.py et/ou fonction.py :***

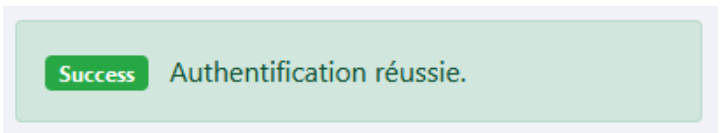
1) Si l'utilisateur est authentifié:
   - ✓ initialisez les variables de sessions:
     - session['nom']
     - session['prenom']
     - session['idUser']
     - session['login']
     - session['statut']
     - session['mail']
     - session['avatar']
   - ✓ Redirigez l'internaute vers soit la page d'accueil, soit une des pages privées en fonction du statut de la personne (ex : vers administration.html si l'internaute a un statut administrateur)
2) Si l'authentification est refusée, Redirigez l'internaute vers la page login.html

## d. Gestion des messages de succès ou d'échec - *cf slide 136*

1) En cas de refus d'authentification, faire apparaître dans la page « se connecter » :




2) En cas d'authentification réussie, faire apparaître dans la page de redirection :




+ Indiquez le nom et le prénom de la personne authentifiée sur toutes les pages du site




Karine PUECHMOREL – Z134
Karine.puechmorel@enac.fr

e. **Modification du menu de navigation :**

Modifier le menu de navigation de manière à ce que :

- Les menus "se déconnecter " et ceux propres aux gestionnaires n'apparaissent que si l'utilisateur est authentifié - *cf slide 137*
- Le menu de "Administration" et "Création de compte" n'apparaissent que si l'utilisateur est authentifié et qu'il a un statut administrateur

N.B : Pour le menu « se déconnecter », penser à créer une route logout qui déconnecte l'utilisateur en supprimant les sessions existantes. - *cf slide 134*

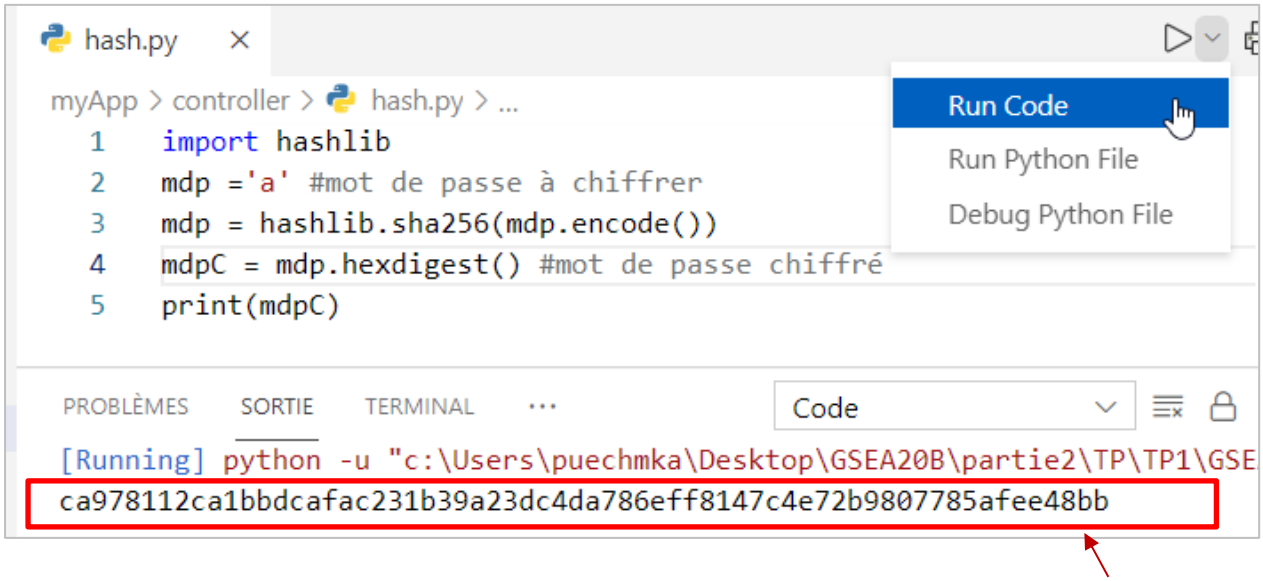
**f. Chiffrement du mot de passe :**

Afin d'améliorer la sécurité des mots de passe enregistrés en base de données. Nous allons chiffrer les mots de passe lors de la création des comptes et lors de l'authentification.

Pour cela, nous allons utiliser la library python hashlib: https://www.geeksforgeeks.org/sha-in-python/

**1) C**hiffrement des mots de passe.

- Dans vscode, créer un fichier hash.py contenant le code suivant




- Exécuter le code dans la console, vous obtenez le chiffrement pour le mot de passe de Louis Blériot.
- Copier le mot de passe chiffré obtenu, puis collez-le dans la table identification.
- Procéder de même pour les mots de passe d'Hélène Boucher et de admin. Vous devez obtenir ceci :

| idUser | nom | prenom | mail | login | motPasse | statut | newMdp |
|---|---|---|---|---|---|---|---|
| 1 | Blériot | Louis | louis.bleriot@enac.fr | a | ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9... | 0 | 0 |
| 2 | Boucher | Hélène | helene.boucher@enac.fr | b | 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb... | 1 | 0 |
| 3 | admin | admin | admin@enac.fr | admin | 8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a8... | 0 | 0 |



Karine PUECHMOREL – Z134
Karine.puechmorel@enac.fr

**2)** Authentification avec mot de passe chiffré.

- Tester à nouveau l'authentification. Vous devez systématiquement obtenir un refus.

En effet, votre requête SQL actuelle compare le mot de passe en clair avec un mot de passe chiffré. Elle renverra donc toujours une erreur. Pour régler ce problème, dans verifAuthData, procéder au chiffrement du mot de passe reçu avant que la requête sql ne soit exécutée.

```
import hashlib
def verifAuth(login, mdp):
    mdp = hashlib.sha256(mdp.encode())
    mdpC = mdp.hexdigest() #mot de passe chiffré
```



Karine PUECHMOREL – Z134
Karine.puechmorel@enac.fr

# Equipe 2 :
## *Création des comptes utilisateurs*

***Principe :***

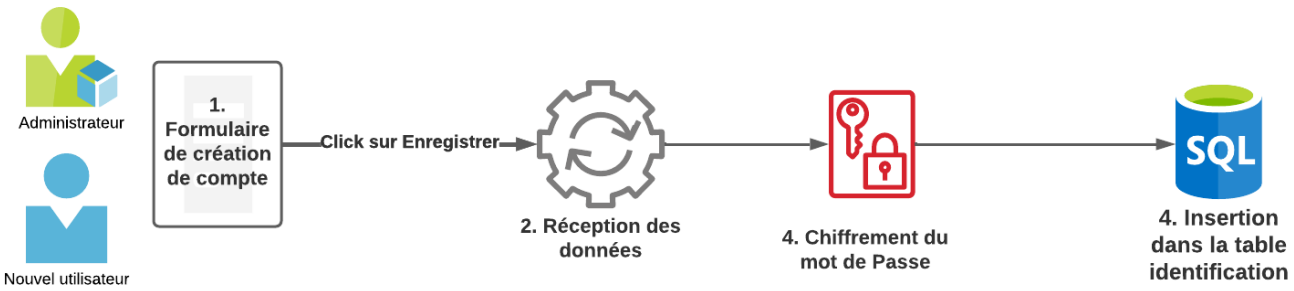
Selon les sujets, le compte utilisateur peut être crée soit par une personne non connectée soit par un administrateur.

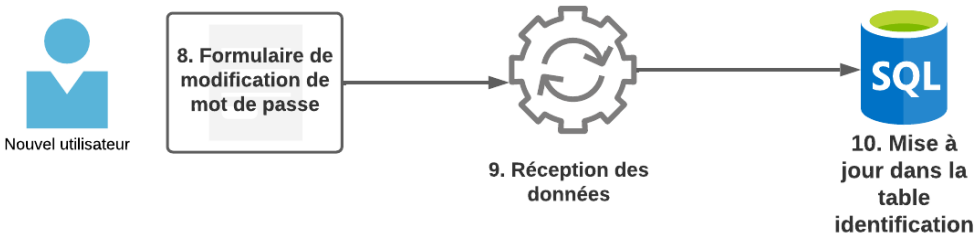
***Création du compte utilisateur***

Tout nouveau compte est créé à partir d'un formulaire. Ces données sont transmises au serveur web pour insertion dans la base de données. Avant insertion en BDD, le mot de passe est chiffré.

***Modification du mot de passe***

L'utilisateur se connecte à l'application et peut accéder à un formulaire pour modifier son mot de passe. Les modifications sont enregistrées en BDD.






Karine PUECHMOREL – Z134
Karine.puechmorel@enac.fr

***Travail à réaliser :***

**a. Envoi et réception des données du formulaire**

**Dans compte.html**

- Dans un fichier html, rajouter un formulaire de création de compte si cela n'a pas été réalisé lors de la séance 1 du BE

***Dans views.py et function.py :***

- Réceptionner les données du formulaire de création
- Chiffrer le mot de passe

```
import hashlib
mdp = hashlib.sha256(mdp.encode())
mdpC = mdp.hexdigest() #mot de passe chiffré
```

- Appeler la fonction add_userData(email , nom, prenom, statut, login, motPasse, avatar) qui sera créée juste après dans bdd.py. Note : mdpC = mot de passe Chiffré

**b. Insertion et mise à jour des données dans la table utilisateur**

***Dans bdd.py :***

- Créer la fonction **add_userData** (email , nom, prenom, statut, login, motPasse, avatar) qui insère les paramètres du nouveau compte dans la table identification - *cf cours slides 121-122*

- Créer la fonction **update_userData** (champ, newValue, idUser) qui modifie un champ de la table identification avec la valeur newValue pour l'utilisateur ayant pour id IdUser (*cf slide 126*)

**c. Création du formulaire de mise à jour du mot de passe.**

- Demander aux membres de l'équipe 1 de vous fournir le code permettant l'authentification des utilisateurs.
- Dans un nouveau fichier nommé profil.html, proposer un formulaire pour changer de mot de passe avec les champs « Ancien mot de passe », « Nouveau mot de passe », « Confirmer votre mot de passe ». Le menu « Mon profil » ne doit être accessible qu'une fois l'utilisateur authentifié.

**d. Réception et enregistrement des données du formulaire de mise à jour**

***Dans views.py et functions.py :***

- Réceptionner les données du formulaire de mise à jour.
- Vérifier que le mot de passe et celui de confirmation sont bien identiques.
- Appeler la fonction **update_userData** (champ , newvalue, idUser) où la valeur de idUser est enregistrée en session afin de mettre à jour le nouveau mot de passe (champ="motPasse")

- Afficher un message d'information indiquant que la modification s'est bien déroulée sur la page HTML de votre choix.



Karine PUECHMOREL – Z134
Karine.puechmorel@enac.fr