

Client Serveur avec Flask - TD2 Suite

15 Avril 2024

8 Page de connexion

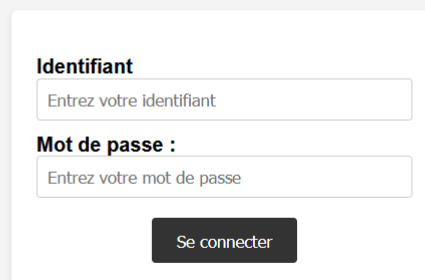
Durant les dernières parties, nous avons vu comment utiliser une bibliothèque Python (datetime) et afficher des informations de l'heure en cours. Nous avons également vu comment créer un formulaire ainsi que la récupération des informations envoyées par l'utilisateur.

Dans la suite du TD, vous devrez réaliser un système de connexion, permettant de sécuriser votre site.

Pour cela vous allez devoir commencer par créer un nouveau fichier "connexion.html" ainsi qu'un nouveau décorateur (Voir partie précédente).

Cette page devra contenir un formulaire de connexion comportant deux champs, le premier correspondant à l'identifiant, le second correspondant au mot de passe. Le formulaire devra également avoir un bouton permettant d'envoyer le formulaire.

Ceci est la page de connexion, veuillez vous connecter pour accéder au site !



The image shows a login form with a light gray background. At the top, there is a bold heading: "Ceci est la page de connexion, veuillez vous connecter pour accéder au site !". Below this, there is a white rectangular box containing the form fields. The first field is labeled "Identifiant" and has a placeholder text "Entrez votre identifiant". The second field is labeled "Mot de passe :" and has a placeholder text "Entrez votre mot de passe". Below these fields is a dark gray button with the text "Se connecter" in white.

Quelles méthode le formulaire doit-il avoir ?
N'oubliez pas d'ajouter l'URL à laquelle le formulaire devra envoyer les informations.

9 Vérification connexion

Nous allons maintenant modifier notre fichier "views.py". Selon l'URL que vous avez choisi dans la partie précédente, créez un nouveau décorateur. Quelle méthode devrat-il avoir ?

Ce décorateur va nous permettre de vérifier si le mot de passe et l'identifiant sont bons. Avant cela, nous allons devoir créer deux variables, une pour votre identifiant, l'autre pour votre mot de passe. Ces variables seront ensuite comparées aux données du formulaire pour vérifier si elles sont identiques.

Une fois ces deux variables créées, récupérez les données du formulaire et vérifiez qu'elles sont identiques.

10 Variable de session

Notre but va être de faire en sorte de limiter l'accès de toutes les pages de notre site aux utilisateurs connectés. Ajoutez au début de votre programme la ligne suivante :

```
1 app.secret_key = 'c1esecrete'
```

Ensuite, nous allons initialiser notre variable de session "is_connected" à True lorsque les données sont identiques.

```
1 session['is_connected'] = True
```

Une variable de session permet d'utiliser une variable dans tous les décorateurs de notre programme, permettant ainsi de faciliter les vérifications. Une fois la variable initialisée, nous allons renvoyer l'utilisateur sur notre page d'accueil. Pour cela, ajoutez la ligne de code suivante :

```
1 return index()
```

Nous allons maintenant vérifier que l'utilisateur est connecté dans tous les décorateurs. Pourquoi devons nous effectuer cette vérification ?

Pour savoir si l'utilisateur est connecté, nous allons utiliser la fonction suivante :

```
1 if session.get('is_connected', False):
```

Recherchez comment fonctionne cette ligne de code. Puis faites les vérifications dans tous les décorateurs. Si l'utilisateur n'est pas connecté, alors redirigez le vers la page de connexion.

11 Déconnexion

Pour se déconnecter, nous allons ajouter un bouton à notre barre de navigation. Ouvrez le fichier "navbar.html" puis ajouter un bouton "Déconnexion". Ajoutez à ce bouton l'attribut "href" avec la valeur "/deconnexion".

Revenez maintenant au fichier python et ajoutez un nouveau décorateur déconnexion. Dans ce décorateur, nous allons supprimer notre variable de session "is_connected", puis nous allons envoyer l'utilisateur sur la page de connexion.

```
1 session.pop('is_connected', None)
2 return connexion()
```

12 Pour aller plus loin

Ajoutez du CSS à votre formulaire de connexion afin de le rendre plus agréable visuellement. Attention, sur Flask, le fichier CSS doit se trouver dans un dossier à part entière.

<https://pythonhow.com/python-tutorial/flask/Adding-CSS-styling-to-your-website/>