

TP RECAP J1

1. Créer un composant d'icone de déconnexion (*faPowerOff*) et l'injecter dans le composant UI.
2. Afficher sur le bandeau header l'icone de déconnexion.

TP 1

1. Pour le client, créer l'enum de state + interface + modele (State='Active' /tva=20 / total = "this.ca * (1 + this.tva / 100);")
2. Créer le service des clients.
3. S'inscrire à l'observable depuis la page des clients et récupérer les données en base.

TP RECAP J2

1. Ettofer le modèle de données db.json en y ajoutant 3 users:
 - id: 1 / username: 'admin' / password: 'admin' / role: 'ADMIN' (*type enum*)
 - id: 2 / username: 'user1' / password: 'user1' / role: 'USER'
 - id: 3 / username: 'user2' / password: 'user2' / role: 'USER'
2. Créer le modèle associé dans Angular. (enum, interface, model)
3. Créer 2 services pour les users sous forme d'observable (*Le service peut être implémenté dans le module Core*), qui permettent :
 - de récupérer la liste des users.
 - de récupérer un user avec son username et password.

TP 2

1. Créer une page qui affiche les users (sans le mot de passe) : View + Lazyloading + Lien depuis la nav dans un tableau bicolor (*class table-striped*)
2. Ajouter 1 bouton qui redirige vers une route 'add User'

TP RECAP J3

1. Ajouter dans le db.json les 2 clients :
 - "id":5 / "state":"Active" / "tva": 20 / "ca": 5000 / "name": "Mobilo" / "comment": ""
 - "id":6 / "state": "Inactive" / "tva": 20 / "ca": 75000 / "name": "Tete" / "comment": "new !!"
2. Créer une méthode qui récupère l'ensemble des clients et qui modifie les données depuis l'observable pour ne renvoyer que les clients avec un CA en dessous de 100,000€.
3. Depuis la page des clients, ajouter un bouton nommé 'Filter' qui permet de switcher entre le tableau initial et le tableau filtré.

BONUS : Ajouter une icône 'filter' avant le nom du bouton

TP 3

1. Le besoin du client est :

...je veux un titre écrit en gras... et aussi le sous titre affiché en italique...
uniquement sur la page des clients.

Astuce : La propriété `IngClass` est une liste.

TP 4

1. Créer une page login, qui affiche à l'écran un formulaire avec deux champs de type input (Username – Password).
2. Ajouter un bouton de validation qui va effectuer une requête en DB pour authentifier (ou non) l'utilisateur.
3. Si l'utilisateur est authentifié, rediriger l'utilisateur vers la page Home.

BONUS: La page des users peut être accessible par tous les utilisateurs mais :

- *Si non connecté :* *Message 'Désolé, user non connecté'*
- *Si role user:* *Affichage uniquement de son user*
- *Si role admin:* *Affichage complet des users*

TP 5

1. Depuis la page des users, ajouter un bouton d'ajout de user, visible uniquement par le role Admin.
2. Le bouton doit rediriger vers un formulaire de création de user, qui permet pour le role 'Admin' de créer un nouvel user (Tous les champs sont obligatoires) et pour le role 'User' d'afficher un message d'erreur lui indiquant qu'il n'a pas les droits requis pour accéder à cette page.
3. Un bouton de retour à la page précédente devra être implémenté.

BONUS : Toutes les pages métiers 'users' doivent afficher un titre de page 'User Management', centré, en gras, couleur #4B0082 et background color #F8F8FF.

TP 6

1. Dans ma page des clients, je veux avoir un champs supplémentaire qui inclut 3 boutons (Icône View + Icône Edit + Icône Delete):
 1. **View** : Redirection vers une page de visualisation de mon client, qui affiche les informations suivantes : « Client », « Commentaire », « CA », « TVA », « TotalEuro », « TotalDollar » (Rappel : $1\text{€} = 1,18\text{\$}$) + 1 bouton de retour.
 2. **Edit** : Redirection vers la page d'édition de mon client, qui permet d' éditer tous me champs sauf le nom du client + 1 bouton de save + 1 bouton de retour.
 3. **Delete** : Suppression du client en base sans changer de page.