



TP6 : Authentification multi-utilisateurs

Objectif

Créer une **page d'authentification complète** permettant à **plusieurs utilisateurs prédéfinis** de se connecter.

Une fois connecté, le **nom d'utilisateur s'affiche sur toutes les pages** de l'application.

Contexte

Tu disposes déjà d'une application React Native avec navigation entre :

- **HomeScreen**
- **ProfileScreen**
- **SettingsScreen**

Tu dois maintenant :

1. Ajouter une **page LoginScreen** pour gérer la connexion.
 2. Autoriser **3 utilisateurs** différents à se connecter.
 3. Passer le **nom de l'utilisateur connecté** à toutes les pages.
 4. Afficher ce nom sur chaque écran.
 5. Ajouter un **bouton Déconnexion** sur la page d'accueil.
-



Consignes techniques



Créer le fichier de connexion

Crée un nouveau fichier :

```
screens/LoginScreen.js
```



Ajouter la page dans la navigation

Dans `App.js`, ajoute l'écran de connexion et définis-le comme **écran initial** :

```
<Stack.Screen name="Login" component={LoginScreen} />
```

Et dans ton `Stack.Navigator` :

```
initialRouteName="Login"
```

3 Créer trois utilisateurs autorisés

Dans `LoginScreen.js`, définis une petite base de données locale :

```
const users = [
  { username: 'alex', password: '1234' },
  { username: 'marie', password: 'abcd' },
  { username: 'tomi', password: 'pass' },
];
```

Lors du clic sur **Se connecter**, vérifie si le couple (login / mot de passe) existe dans ce tableau.

☞ Si oui :

```
navigation.replace('Home', { username });
```

☞ Sinon :

Affiche un message d'erreur avec `Alert.alert('Erreur', 'Identifiants incorrects')`.

4 Transmettre le nom d'utilisateur connecté

- Passe `username` dans la navigation :
`navigation.replace('Home', { username });`
 - Dans **HomeScreen**, **ProfileScreen** et **SettingsScreen**, récupère le nom :
`const username = route.params?.username || 'Utilisateur';`
 - Puis affiche-le en haut de la page :
 - `<Text>Connecté en tant que : {username}</Text>`
-

5 Ajouter le bouton “Déconnexion”

Sur **HomeScreen**, ajoute un bouton :

```
<Button
  title="Déconnexion"
  color="red"
  onPress={() => navigation.replace('Login')}
/>
```

Cela ramène l'utilisateur à la page de connexion.