

SnapPic - Part 1 / Consignes

SnapPic est une application mobile permettant de prendre des photos et de les stocker dans une galerie d'images.

L'objectif du jour est de créer la structure de l'application à partir des maquettes ci-dessous. Vous allez devoir mettre en place la navigation et créer les différents screens ainsi que leur style.



Prépare ton environnement de travail

- 👉 Crées un dossier appelé **snap-pic**.
- 👉 Crées un setup react-native de base.
- 👉 Lances ton app avec yarn start et scan le QR code avec ton smartphone.



Mise en place de la navigation

Avant de commencer à développer le style et les fonctionnalités de l'application, il est primordial de mettre en place la mécanique de navigation.

- 👉 Installes tout les modules de React navigation.
- 👉 Dans un dossier nommé screens met en place les composants représentant les différents écrans. Au total, créés 3 écrans avec les noms suivants :
 - HomeScreen
 - GalleryScreen
 - SnapScreen
- 👉 Dans le fichier App.js, mets en place la navigation comme illustrée dans les maquettes.
- 👉 Customises le menu en mettant en place les icônes FontAwesome correspondantes aux différentes screens:
 - SnapScreen : camera
 - GalleryScreen : image
- 👉 Modifies les couleurs de icônes :
 - pour une icône active «#e8be4b
 - pour une colonne inactive #b2b2b2

👉 Fais en sorte que l'utilisateur arrive directement sur l'onglet 'Gallery' lorsqu'il valide son email dans HomeScreen.

💡 Pour spécifier l'écran sur lequel vous souhaitez arriver en naviguant du HomeScreen vers la partie de l'application avec le menu du bas, vous pouvez ajouter cette ligne dans le fichier HomeScreen.js :

```
navigation.navigate('TabNavigator', { screen: 'Gallery' });
```



Intégration des écrans

La navigation est maintenant en place, tu peux passer à l'intégration des screens !

HomeScreen

👉 Intègres l'écran HomeScreen en t'appuyant sur les maquettes. Cet écran d'accueil doit obligatoirement contenir un input avec comme placeholder "Email" et un bouton "Go to gallery" pour naviguer vers GalleryScreen.

👉 Mets en place une regex pour vérifier si l'email envoyé est bien valide. Si ce n'est pas le cas, affiches un message d'erreur « Invalid email address » en dessous de l'input.

💡 Les regex peuvent être fastidieuses à créer soi-même, heureusement internet regorge de regex prêtent à être utilisés.

👉 L'email, une fois validé, doit être enregistré dans le store via le reducer user.js (à créer dans le dossier reducers à créer lui aussi).

SnapScreen

L'écran SnapScreen serai à afficher la caméra, pour l'instant affiches juste le texte « SnapScreen »

GalleryScreen

👉 L'écran GalleryScreen affiche la liste de tes photos (fictives pour le moment) qui sera définie par le tableau photosData ci-dessous :

```
const photosData = [  
  'https://res.cloudinary.com/babahedz/image/upload/v1686507914/4_dh1u9w.png',  
  'https://res.cloudinary.com/babahedz/image/upload/v1686507914/1_f3guh8.png',  
  'https://res.cloudinary.com/babahedz/image/upload/v1686507914/2_lrjyge.png',  
  'https://res.cloudinary.com/babahedz/image/upload/v1686507913/3_hm866a.png',  
]
```

👉 Ajoutes une icône de suppression via FontAwesome (le nom de l'icône est **times**) pour chaque photo. L'effet n'enclenche aucune action pour le moment.

Bonne chance 🧑💻