ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

Atelier Navigation et Routage('routing')

Partie 1 Navigation Simple

On va avoir besoin de trois classes, on mettre chacune dans un fichier différent:

main.dart,

firstpage.dart

secondpage.dart

Rappel : *Utilisation de Navigator*

Le widget ***Navigator*** affiche les écrans comme étant une pile utilisant la transition correcte en utilisant les animations de transition correctes pour la plate-forme cible.

Pour naviguer à un nouvel écran, accéder à travers le Navigator à travers le *BuildContext* du route et appeler des méthodes impératives comme **push()** et **pop()**.

main.dart

```
class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation And Routing Demo',
      theme: ThemeData(
        primarySwatch: Colors.lightGreen,
      ),
      home:FirstPage(),
    );
  }

}
```

firstpage.dart

```
class FirstPage extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Routing App'),),
        body: Center(
          child: Column(
```

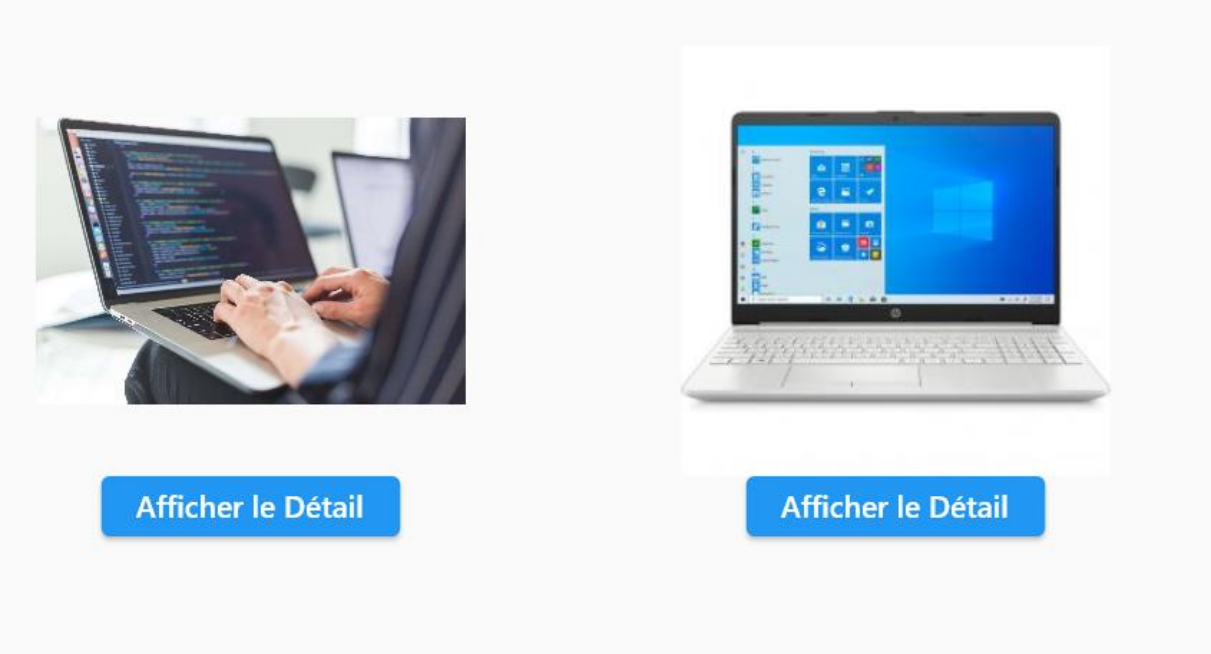
```
                .....
                Text('First Page', style: TextStyle(fontSize: 50), ),
                ElevatedButton(onPressed: (){
                  Navigator.of(context).push(
                    MaterialPageRoute(builder: (context)=>
                              SecondPage(),
                          )
                  );
                }, child: Text('Naviguer vers la seconde page')),
              ],
.....

}
```

secondpage.dart

```
class SecondPage extends StatelessWidget{



   @override
  Widget build(BuildContext context) {
    return Scaffold(
      ……
          children: [
            Text('Second Page', style: TextStyle(fontSize:  50),),
            Text('Vous y êtes', style: TextStyle(fontSize:  25),)
            ElevatedButton(onPressed: (){
              Navigator.of(context).pop();
            }
                , child: Text('Retour au menu')),

          ],
        ),
```

Application

Appliquez le concept de navigation, en construisant une interface similaire à la suivante :



A noter : Pour l'ajout d'une image sur Flutter

1. Créer un dossier nommé *images* ou *assets* dans votre projet, et mettez-y les images que vous voulez ajouter à votre programme
2. Pour l'ajout des images à partir de ce dossier, on utilise l'instruction :

```
Image.asset("images/coding2.jpg") ;
```

3. Il est aussi obligatoire d'ajouter dans la section flutter de votre fichier **pubsec.yaml** , le chemin relatif de vos images

```
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true

  # To add assets to your application, add an assets section, like this:
  # assets:
  #   - images/a_dot_burr.jpeg
  #   - images/a_dot_ham.jpeg
  assets:
    - images/coding.jpg
```

Deuxième Partie : Les routes/itinéraires nommés

Dans la navigation vers un nouvel écran et retour, vous avez appris à naviguer vers un nouvel écran en créant un nouveau 'route' et en le poussant vers le Navigateur.

Toutefois, si vous devez accéder au même écran dans de nombreuses parties de votre application, cette approche peut entraîner une duplication de code. La solution consiste à définir un «route nommé » et à utiliser le « route nommé » pour la navigation.

main.dart

```
Widget build(BuildContext context) {
  return MaterialApp(
    initialRoute: '/',
    routes:<String, WidgetBuilder> {
      '/' : (context) => FirstPage(),
      '/second': (context) => SecondPage()
    },
  );
```

FirstPage.dart

```
class FirstPage extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Routing App'),),
        body: Center(
             …

              ElevatedButton(onPressed: (){
                Navigator.pushNamed(context, '/second');

              }, child: Text('Naviguer vers la seconde page')),
            ],
          )
```

La dernier page (secondpage.dart) qui contient Navigator.pop(context) reste la même.

Troisième partie : Cherchez le widget ***BottomNavigationBar*** et essayez de l'adapter à votre code.