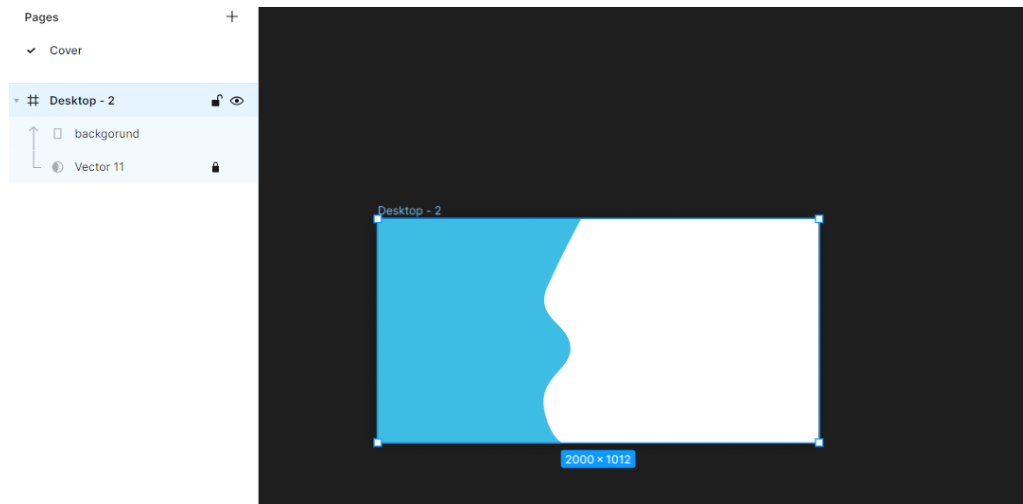


# Documentation de Travail

## Réaliser

## Réalisation de Background pour version Desktop :

La première chose que j'ai faite c'est de réaliser le design en figma, j'ai utilisé un Vector pour désigner le zigzag ;



## Réalisé le Projet flutter :

Premièrement j'ai réalisé un widget responsive "Responsive" qui est responsable de la gestion de la réactivité c'est-à-dire est la capacité de s'adapter dynamiquement à différentes tailles d'écran et à fournir une expérience utilisatrice optimale, quel que soit l'appareil utilisé, la classe "Responsive" contient plusieurs méthodes statiques pour déterminer la taille de l'écran par exemple :

`getScreenReolution(context)`: Cette méthode renvoie un dictionnaire contenant la largeur et la hauteur de l'écran

```
class Responsive extends StatelessWidget {
  const Responsive({
    super.key,
    required Widget desktop,
    required Widget mobile,
    Widget? tablet});

  static Map<String, double> getScreenReolution(context) {
    return {
      'width': MediaQuery.of(context).size.width,
      'height': MediaQuery.of(context).size.height
    };
  }

  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    throw UnimplementedError();
  }
}
```

## Réaliser le widget Web :

La première chose que j'ai faite c'est d'ajouter le background que j'ai réalisé on figma a mon code : pour ça j'ai utilisé un container pour créer une boîte rectangulaire pour remplir toute la hauteur et la largeur de l'écran (grâce aux propriétés height: double.infinity et width: double.infinity).

Avec une décoration avec une couleur bleue (définie par color: Colors.blueAccent) et une image de fond mon Background (définie par image: Decoration Image(...)).

```
return Scaffold(
  body: Container(
    height: double.infinity,
    width: double.infinity,
    decoration: const BoxDecoration(
      color: Colors.blueAccent,
      image: DecorationImage(
        image: AssetImage('assets/images/desktop.png'),
        fit: BoxFit.cover,
      ),
    ),
  ), return Scaffold(
  body: Container(
    height: double.infinity,
    width: double.infinity,
    decoration: const BoxDecoration(
      color: Colors.blueAccent,
      image: DecorationImage(
        image: AssetImage('assets/images/desktop.png'),
        fit: BoxFit.cover,
      ),
    ),
  ),
),
```

Après j'ai réalisé l'header de mon page en utilisant un row : pour disposer mes deux widgets (logo et icône de liste) côte à côte. :

```
child: Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Image.asset(
      'assets/images/Image 2.png',
      width: Responsive.getScreenReolution(context)['width']! / 5,
      height: Responsive.getScreenReolution(context)['height']! / 5,
    ),
    GestureDetector(
      onTap: () {},
      child: Icon(
        Icons.list_sharp,
        color: Colors.black,
        size: 40,
      ),
    ),
  ],
),
```

Ensuite j'ai utilisé un autre Row pour disposer les deux widgets de container de la partie bleu et le container de partie blanche :

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    Container(...), // Container  
    Container(...), // Container  
  ],  
) // Row
```

### Le premier container Contient :

Les deux textes (Bonjour et le paragraphe) eurent aussi les deux texte Field (Nom et l'intention).

```
Container(  
  constraints: BoxConstraints(  
    maxWidth: Responsive.getScreenReolution(context)['width']! / 4,  
  ), // BoxConstraints  
  child: Text(  
    'Pour commencer...',  
    style: TextStyle(...), // TextStyle  
  ), // Text  
) // Container  
SizeBox(height: Responsive.getScreenReolution(context)['height']! / 40),  
Container(  
  constraints: BoxConstraints(  
    maxWidth: Responsive.getScreenReolution(context)['width']! / 4,  
  ), // BoxConstraints  
  child: Text(  
    'faire votre Shynlei, jouer avec, c'est l'occasion d'écouter vos rêves, de les partager et de prendre confian',  
    style: TextStyle(...), // TextStyle  
  ), // Text  
) // Container
```

L'autre container contient les 7 étapes de syhnlei qui est composé d'un Svg et d'un texte pour ça j'ai utilisé un widget personnalisé qui combine l'image SVG avec le texte dans une rangée horizontale (Row):

```
class SvgTextWidget extends StatelessWidget {  
  final String svgPath;  
  final String text;  
  SvgTextWidget({required this.svgPath, required this.text});  
  @override  
  Widget build(BuildContext context) {  
    double taille = Responsive.getScreenReolution(context)['width']! +  
      Responsive.getScreenReolution(context)['height']! ;  
    return Row(  
      children: [  
        SvgPicture.asset(  
          svgPath,  
          width: taille / 160,  
          height: Responsive.getScreenReolution(context)['height']! /  
            14,  
        ),  
        SizeBox(width: Responsive.getScreenReolution(context)['width']! /  
          60),  
        Text(  
          text,  
          style: TextStyle(fontSize: taille / 160),  
        ),  
      ],  
    ),  
  ),  
}
```

## Réaliser le widget phone :

La première chose que j'ai faite s'est de réaliser un autre widget un widget personnalisé qui combine l'image SVG avec le texte avec les démonstrations possible pour l'interface mobile

```
class SvgTextWidgetphone extends StatelessWidget {
  final String svgPath;
  final String text;

  SvgTextWidgetphone({required this.svgPath, required this.text});

  @override
  Widget build(BuildContext context) {
    double taille = Responsive.getScreenReolution(context)['width']! +
      Responsive.getScreenReolution(context)['height']! ;

    return Row(
      children: [
        SvgPicture.asset(
          svgPath,
          width: taille / 160,
          height: Responsive.getScreenReolution(context)['height']! /
            14,
        ),
        SizedBox(width: Responsive.getScreenReolution(context)['width']! /
          60),
        Text(
          text,
          style: TextStyle(fontSize: taille / 100),
        ),
      ],
    );
  }
}
```

Pour l'interface mobile j'ai centré (Center) le contenu et rempli avec un espace de 20 points de chaque côté (Padding). Et j'ai l'organisé dans une colonne (Column) pour empiler les éléments verticalement Le contenu de page contiennent deux textes et les widgets personnalisés avec un bouton (Texte Button) enveloppé dans un conteneur (Container)

Le bouton est stylisé avec une couleur de fond cyan, une bordure, une forme de coin arrondi et un texte centré.

```

Text(
  'Votre Shynlei ',
  style: TextStyle(
    fontFamily: 'Rubik',
    fontSize: Responsive.getScreenReolution(context)['height']! / 20,
    fontWeight: FontWeight.w800,
    height: 1.185,
  ), // TextStyle
), // Text
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 60), // Espacement entre les deux Text
Container(
  constraints: BoxConstraints(
    maxWidth: Responsive.getScreenReolution(context)['width']! / 1.3,
  ), // BoxConstraints
  child: Text(
    '7 étapes, 2 fiches pour prendre note des rencontres, 1 page pour éclairer le sens, 3 interprétations pour vous aider',
    style: TextStyle(
      fontFamily: 'Rubik',
      fontSize: Responsive.getScreenReolution(context)['height']! / 51,
      fontWeight: FontWeight.w600,
      height: 1.5,
    ), // TextStyle
  ), // Text
), // Container
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 40),
SvgTextWidgetphone(
  svgPath: 'assets/images/Page-1.svg', // Chemin vers votre fichier SVG
  text: "1. Le rêve libre l'expression",
), // SvgTextWidgetphone
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 40),
SvgTextWidgetphone(...), // SvgTextWidgetphone
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 40),
SvgTextWidgetphone(...), // SvgTextWidgetphone
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 40),
SvgTextWidgetphone(...), // SvgTextWidgetphone
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 40),
SvgTextWidgetphone(...), // SvgTextWidgetphone
), // SizedBox(height: Responsive.getScreenReolution(context)['height']! / 40),
SvgTextWidgetphone(...), // SvgTextWidgetphone
), // child: TextButton(
  onPressed: _onNextPressed,
  child: Text(
    "EXPRIMER MES REVES >",
    style: TextStyle(
      fontFamily: 'Rubik',
      fontSize: Responsive.getScreenReolution(context)['height']! / 51,
      fontWeight: FontWeight.w500,
      height: 1.185,
      color: Colors.white,
    ), // TextStyle
  ), // Text
), // TextButton
), //

```

## Home page :

Pour combiner les deux widgets (page web et page mobile) j'ai utilisé un widget personnalisé pour rendre différents widgets en fonction de la taille de l'écran. Elle prend trois widgets en entrée : desktop, tablet, et mobile, et en fonction de la largeur de l'écran, elle renvoie le widget approprié.

```

import 'package:flutter/material.dart';

class ScreenHelper extends StatelessWidget {
  final Widget mobile;
  final Widget tablet;
  final Widget desktop;

  ScreenHelper({Key? key, required this.desktop, required this.mobile, required this.tablet})
    : super(key: key);

  static bool isMobile(BuildContext context) =>
    MediaQuery.of(context).size.width < 800.0;

  static bool isTablet(BuildContext context) =>
    MediaQuery.of(context).size.width >= 800.0 &&
    MediaQuery.of(context).size.width < 1200.0;

  static bool isDesktop(BuildContext context) =>
    MediaQuery.of(context).size.width >= 1200.0;

  @override
  Widget build(BuildContext context) {
    return LayoutBuilder(
      builder: (BuildContext context, BoxConstraints constraints) {
        if (constraints.maxWidth >= 1200.0) {
          return desktop;
        } else if (constraints.maxWidth >= 800.0 &&
          constraints.maxWidth < 1200.0) {
          return tablet;
        } else {
          return mobile;
        }
      },
    );
  }
}

```

Dans page home j'ai utilisé ce widget.

Lorsque l'écran est de type bureau (desktop), il affichera le widget Web pages. Lorsque l'écran est de type tablette (tablet), il affichera également le widget Web pages. Et enfin, lorsqu'il s'agit d'un écran de téléphone mobile (mobile), il affichera le widget phone page.

```

class home extends StatelessWidget {
  @override
  Widget build(BuildContext context)
  {
    return Container(
      child: ScreenHelper(
        desktop: WebPages(),
        mobile: const phonepage(),
        tablet: WebPages(),
      )
    );
  }
}

```