Name: Simran Dung **D15B** **Roll No: 15**

EXPERIMENT 2

Aim : To design Flutter UI by including common widgets.
To include icons, images, fonts in Flutter app

Theory:

- Flutter is Google's UI toolkit for crafting beautiful, natively compiled iOS and Android apps from a single code base. To build any application we start with widgets – The building block of flutter applications.
- Widgets describe what their view should look like given their current configuration and state. It includes a text widget, row widget, column widget, container widget, and many more.
- Widgets: Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the apps. And the structure of the code of an app is a tree of widgets.

Category of Widgets:

There are mainly 14 categories in which the flutter widgets are divided. They are mainly segregated on the basis of the functionality they provide in a flutter application.

1. *Accessibility*: These are the set of widgets that make a flutter app more easily accessible.
2. *Animation and Motion:* These widgets add animation to other widgets.
3. *Assets, Images, and Icons*: These widgets take charge of assets such as display images and show icons.
4. *Async*: These provide async functionality in the flutter application.
5. *Basics:* These are the bundle of widgets that are absolutely necessary for the development of any flutter application.
6. *Cupertino*: These are the iOS designed widgets.
7. *Input*: This set of widgets provides input functionality in a flutter application.

8. *Interaction Models:* These widgets are here to manage touch events and route users to different views in the application.
9. *Layout:* This bundle of widgets helps in placing the other widgets on the screen as needed.
10. *Material Components:* This is a set of widgets that mainly follow material design by Google.
11. *Painting and effects:* This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
12. *Scrolling:* This provides scrollability of to a set of other widgets that arenot scrollable by default.
13. *Styling:* This deals with the theme, responsiveness, and sizing of the app.
14. *Text*: This displays text.

Types of Widgets:

There are broadly two types of widgets in the flutter:

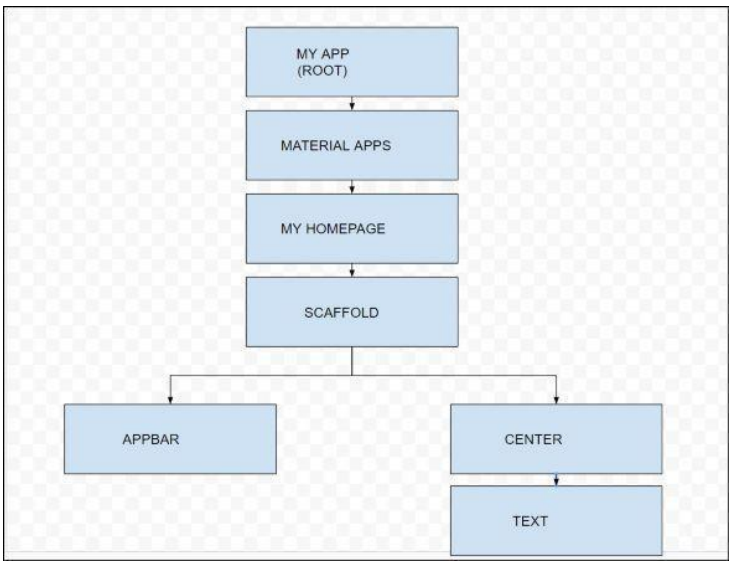
1. Stateless Widget - These are immutable widgets that don't change over time.

- The UI of a stateless widget is defined based on the configuration information passed to it during its creation.
- Example: Container, Text, Icon.

2. Stateful Widget - These are mutable widgets that can change dynamically.

- Stateful widgets maintain a mutable state that might change during the widget's lifetime.
- Example: TextField, ListView, Form.

Name: Simran Dung **D15B** **Roll No: 15**



WIDGET

Code:

1. **Login Page -**

Name: Simran Dung **D15B** **Roll No: 15**

login_screen.dart

news > lib > src > features > authentication > presentation > register > screens > login_screen.dart > _LoginScreen > build

```dart
import '../components/logIn_component.dart';

class LoginScreen extends StatelessWidget {
  const LoginScreen({super.key});

  static const routeName = '/login';

  @override
  Widget build(BuildContext context) {
    return BlocProvider<RegisterBloc>(
      create: (context) => RegisterBloc(context),
      child: const _LoginScreen(),
    ); // BlocProvider
  }
}

class _LoginScreen extends StatelessWidget {
  const _LoginScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final registerBloc = BlocProvider.of<RegisterBloc>(context);
    return BlocBuilder<AuthBloc, AuthState>(builder: (_, state) {
      if (state is LoadingAuthState) {
        return const Scaffold(
          body: Center(
            child: CircularProgressIndicator(),
          ), // Center
        ); // Scaffold
      } else {
        return Scaffold(
          body: Padding(
```

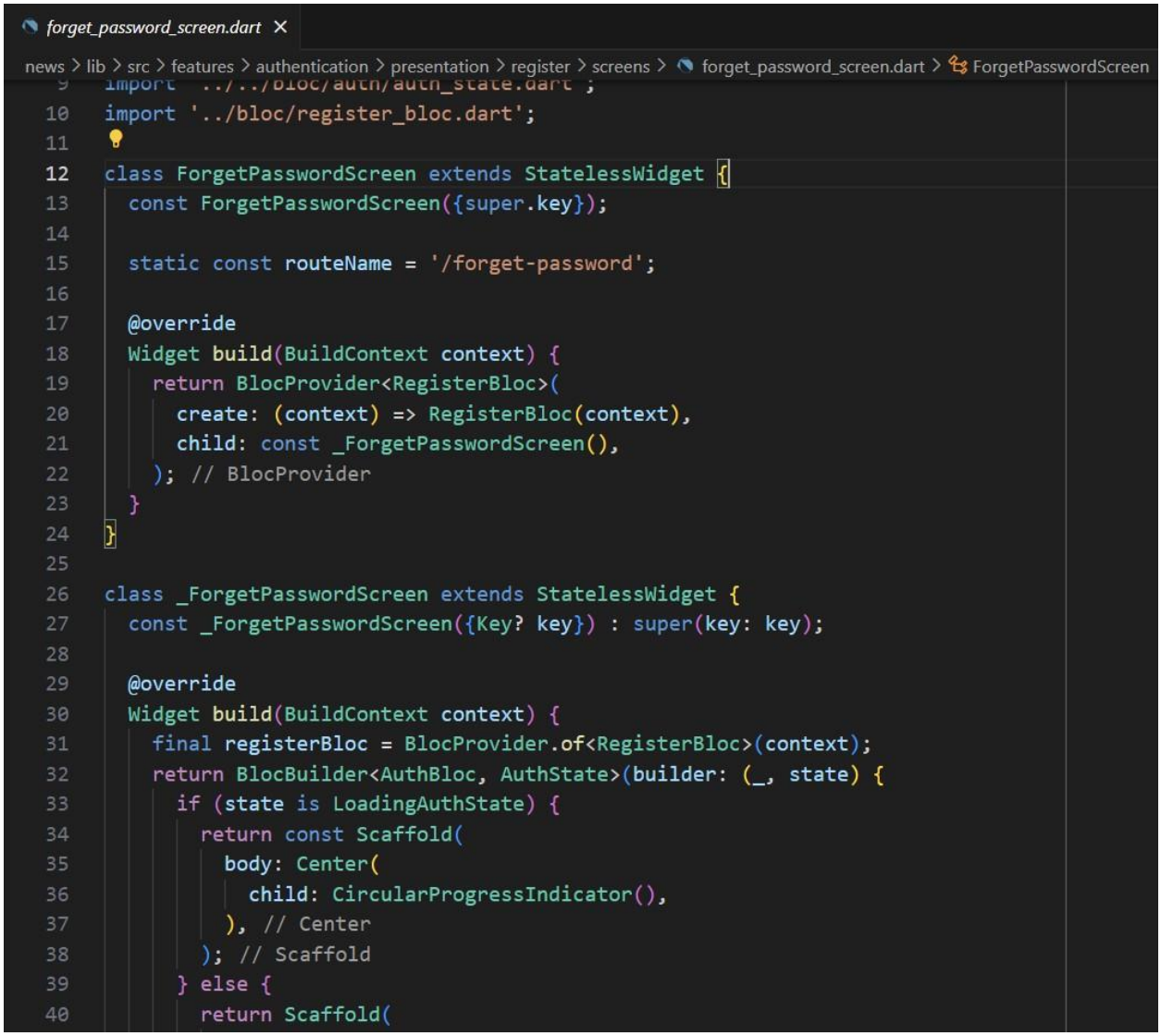
2. Sign Up Page:

signup_screen.dart

news > lib > src > features > authentication > presentation > register > screens > signup_screen.dart > SignUpScreen > SignUpScreen

```dart
import '../bloc/register_bloc.dart';
import '../components/logIn_component.dart';

class SignUpScreen extends StatelessWidget {
  const SignUpScreen({super.key});

  static const routeName = '/signup';

  @override
  Widget build(BuildContext context) {
    return BlocProvider<RegisterBloc>(
      create: (context) => RegisterBloc(context),
      child: const _SignUpScreen(),
    ); // BlocProvider
  }
}

class _SignUpScreen extends StatelessWidget {
  const _SignUpScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final registerBloc = BlocProvider.of<RegisterBloc>(context);
    return BlocBuilder<AuthBloc, AuthState>(builder: (_, state) {
      if (state is LoadingAuthState) {
        return const Scaffold(
          body: Center(
            child: CircularProgressIndicator(),
          ), // Center
        ); // Scaffold
      } else {
        return Scaffold(
```

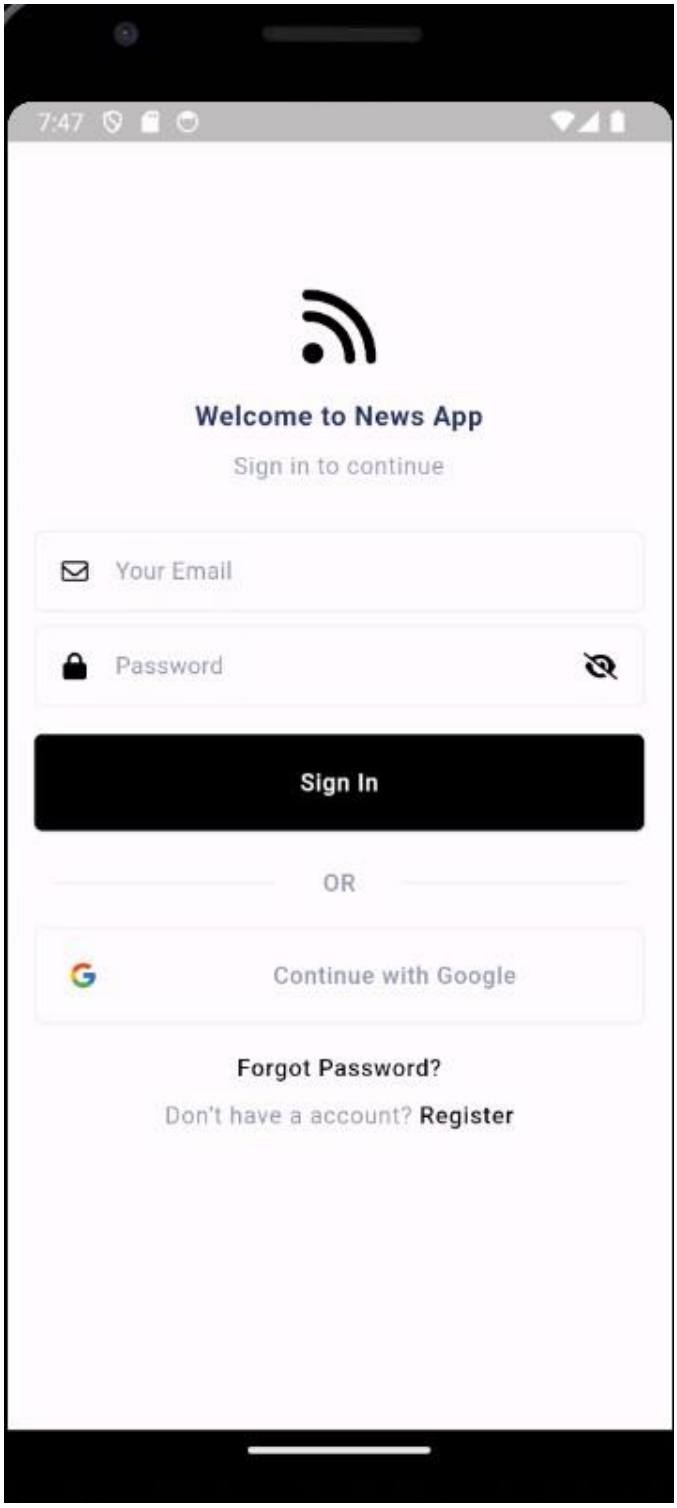
3. Forgot Password:



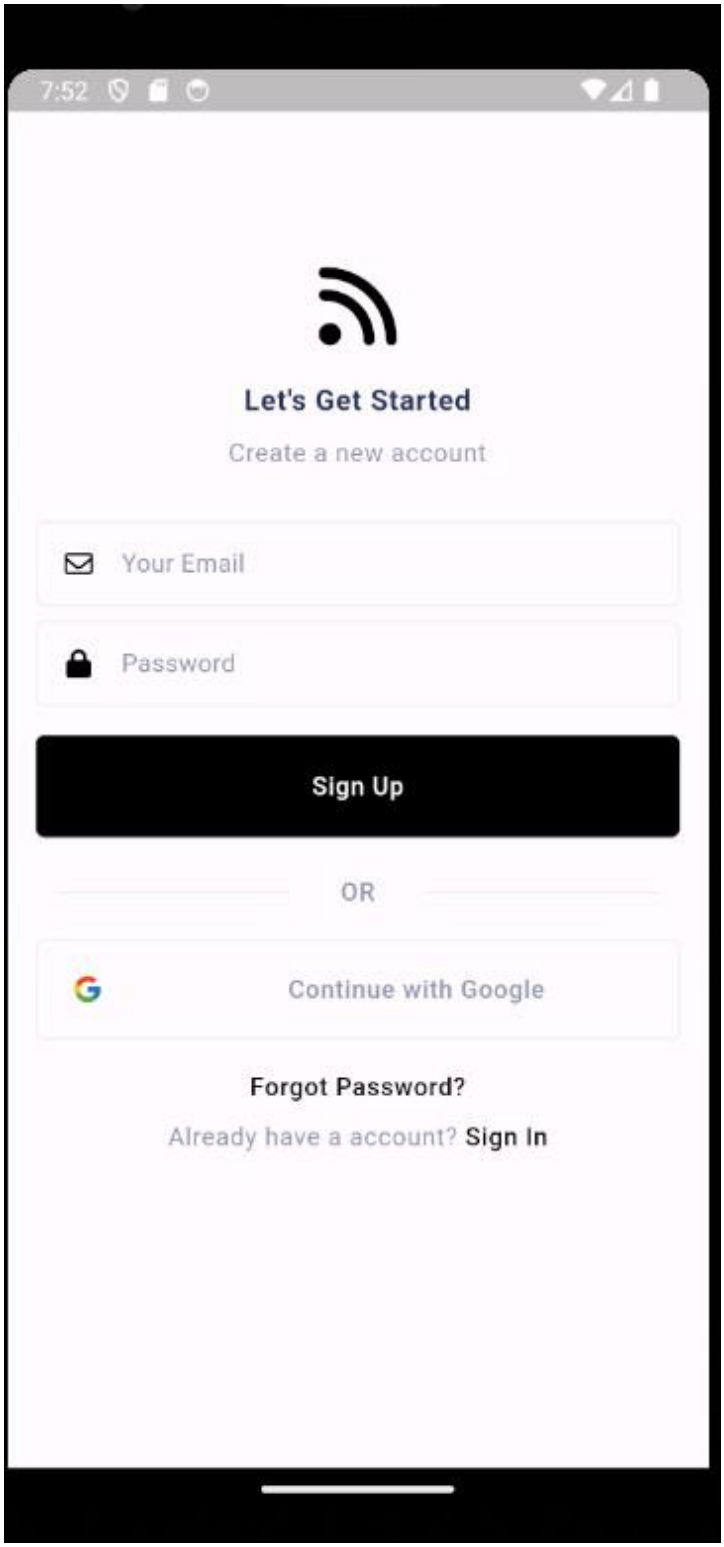
Name: Simran Dung **D15B** **Roll No: 15**

Output:

1. **Login Page:**



2. **Sign Up Page:**



Conclusion : Hence we have understood and studied about the basic widgets in flutter and made use of image, icons and fonts in flutter. With the help of this we have designed a simple login page.