Experiment 2: To Design Flutter UI by including common widgets

Code:

```
import 'package:flutter/material.dart';

class MyAppBar extends StatelessWidget {
 const MyAppBar({required this.title, Key? key}) : super(key: key);

 // Fields in a Widget subclass are always marked "final".

 final Widget title;

 @override
 Widget build(BuildContext context) {
  return Container(
   height: 56.0, // in logical pixels
   padding: const EdgeInsets.symmetric(horizontal: 8.0),
   decoration: BoxDecoration(color: Colors.blue[500]),
   // Row is a horizontal, linear layout.
   child: Row(
    // <Widget> is the type of items in the list.
    children: [
     const IconButton(
      icon: Icon(Icons.menu),
      tooltip: 'Navigation menu',
      onPressed: null, // null disables the button
     ),
     // Expanded expands its child
     // to fill the available space.
     Expanded(
      child: title,
     ),
     const IconButton(
      icon: Icon(Icons.search),
      tooltip: 'Search',
      onPressed: null,
     ),
    ],
   ),
  );
 }
}

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

 @override
 Widget build(BuildContext context) {
  // Material is a conceptual piece
  // of paper on which the UI appears.
  return Material(
   // Column is a vertical, linear layout.
   child: Column(
    children: [
     MyAppBar(
          title: Text(
            'MAD & PWA practicle',
            style: Theme.of(context) //
                .primaryTextTheme
                .headline6,
          ),

        ),
        const Expanded(
          child: Center(
            child: Text('Basic Text Wideget'),


          ),

        ),
        const Expanded(
          child: Center(
            child: Text(' NAME:Chandrakant Anil Jadhav \n Roll NO:32 \n Department: IT'),


          ),
        ),
      ],
    ),
  );
 }
}

void main() {
 runApp(
   const MaterialApp(
     title: 'My app', // used by the OS task switcher
     home: SafeArea(
       child: MyScaffold(),
     ),
   ),
 );
}
```

Exp 3: To create an interactive form using form widgets in flutter

```
import

'package:flutter/material.dart';

void main() => runApp(MyApp());
```

```
class MyApp extends StatelessWidget
 {@override
 Widget build(BuildContext context)
   { final appTitle = 'Flutter Form
   Demo';return MaterialApp(
     title: appTitle,
     home: Scaffold(
     appBar: AppBar(
         title: Text(appTitle),
       ),
       body: MyCustomForm(),
     ),
   );
 }
}
// Create a Form widget.
class MyCustomForm extends StatefulWidget {
 @override
 MyCustomFormState
   createState() {return
   MyCustomFormState();
 }
}
// Create a corresponding State class, which holds data related to the
form. class MyCustomFormState extends State<MyCustomForm> {
 // Create a global key that uniquely identifies the Form widget
 // and allows validation of the form.
 final _formKey = GlobalKey<FormState>();

 @override
 Widget build(BuildContext context) {
   // Build a Form widget using the _formKey created
   above.return Form(
     key: _formKey,
     child: Column(
       crossAxisAlignment:
       CrossAxisAlignment.start,children: <Widget>[
         TextFormField(
           decoration: const
             InputDecoration(icon: const
             Icon(Icons.person), hintText:
             'Enter your full name',
             labelText: 'Name',
           ),
           validator: (String? value)
             {if (value!.isEmpty) {
               return 'Please enter some text';
             }
             return null;
           },
          ),
          TextFormField(
            decoration: const
              InputDecoration(icon: const
              Icon(Icons.phone), hintText:
              'Enter a phone number',
              labelText: 'Phone',
            ),
            validator: (String? value)
              {if (value!.isEmpty) {
                return 'Please enter valid phone number';
              }
              return null;
            },
          ),
          TextFormField(
            decoration: const InputDecoration(
              icon: const
              Icon(Icons.calendar_today),hintText:
              'Enter your date of birth', labelText:
              'Dob',
            ),
            validator: (String? value)
              {if (value!.isEmpty) {
                return 'Please enter valid date';
              }
              return null;
            },
          ),
          new Container(
              padding: const EdgeInsets.only(left: 150.0, top:
              40.0),child: new ElevatedButton(
                child: const
                Text('Submit'),
                onPressed: () {
                  // It returns true if the form is valid, otherwise returns
                  false if (_formKey.currentState!.validate()) {
                    // If the form is valid, display a Snackbar.
                    Scaffold.of(context)
                        .showSnackBar(SnackBar(content: Text('Data is in processing.')));
                  }
                },
              )),
        ],
      ),
    );
  }
}
```

Exp 4: To design a layout of Flutter App using layout widgets

```
import 'dart:ui';

import 'package:flutter/material.dart';

void main() => runApp(const

MyApp());class MyApp extends

StatelessWidget {
 const MyApp({Key? key}) : super(key: key);

 static const String _title = 'Flutter Tutorial';

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: _title,
     home:
     Scaffold(
       appBar: AppBar(title: const Text(_title)),
       body: const MyStatefulWidget(),
     ),
   );
 }
}

class MyStatefulWidget extends StatefulWidget {
 const MyStatefulWidget({Key? key}) : super(key: key);

 @override
 State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends
 State<MyStatefulWidget> {@override
 Widget build(BuildContext context) {
   return Center(
       child: GridView(
         gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
           crossAxisCount: 3,
         ),
         primary: false,
         padding: const
         EdgeInsets.all(20),children:
         <Widget>[
           Container(
             padding: const
             EdgeInsets.all(8),child: const
             Text("Rutuja"), color:
             Colors.orange[200],
           ),
           Container(
             padding: const
             EdgeInsets.all(8),child: const
             Text("Navghane"),
                    color: Colors.green[200],
                   ),
                   Container(
                    padding: const
                    EdgeInsets.all(8),child: const
                    Text("Roll No 48"), color:
                    Colors.red[200],
                   ),
                   Container(
                    padding: const EdgeInsets.all(8),
                    child: const Text("Bharati
                    Vidyapeeth"),color:
                    Colors.purple[200],
                   ),
                   Container(
                    padding: const EdgeInsets.all(8),
                    child: const Text("College Of
                    Engineering"),color:
                    Colors.blueGrey[200],
                   ),
                   Container(
                    padding: const EdgeInsets.all(12),
                    child: const Text("Information
                    Technology"),color: Colors.yellow[200],
                   ),
                  ],
                 )
               );
              }
             }
```

Exp 5: To include icons, images, charts in Flutter App

```
import

'package:flutter/material.dart'; void

main() => runApp(MyApp()); class

MyApp extends StatelessWidget {

// This widget is the root
// of your application

 @override
 Widget build(BuildContext context) {

  return MaterialApp(
home: Scaffold(
  appBar:
  AppBar(
     title: Text('Rutuja Navghane'),
    ),

    body: Center(
     child:
     Column(
      children: <Widget>[
       Image.asset('assets/images/photo.jp
       g'),
      ],
     ),
    ),
   ),
  );
 }
}
```

Pubspec.yaml

```
name: exp1
description: A new Flutter project.

# The following line prevents the package from being accidentally
published to # pub.dev using `flutter pub publish`. This is preferred for
private packages. publish_to: 'none' # Remove this line if you wish to
publish to pub.dev

# The following defines the version and build number for your
application. # A version number is three numbers separated by dots,
like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in
flutter # build by specifying --build-name and --build-number,
respectively.
# In Android, build-name is used as versionName while build-number used as versionCode.
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number used as
CFBundleVersion. # Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Arti
cles/Core FoundationKeys.html
version: 1.0.0+1

environment:
 sdk: ">=2.16.2 <3.0.0"

# Dependencies specify other packages that your package needs in order to
work. # To automatically upgrade your package dependencies to the latest
versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers
```

```
below to # the latest version available on pub.dev. To see which dependencies
have newer
# versions available, run `flutter pub
outdated`.dependencies:
 flutter:
   sdk: flutter



 # The following adds the Cupertino Icons font to your
 application. # Use with the CupertinoIcons class for iOS style
 icons. cupertino_icons: ^1.0.2

dev_dependencies:
 flutter_test:
   sdk: flutter

 # The "flutter_lints" package below contains a set of recommended
 lints to # encourage good coding practices. The lint set provided by
 the package is# activated in the `analysis_options.yaml` file located at
 the root of your
 # package. See that file for information about deactivating specific
 lint# rules and activating additional ones.
 flutter_lints: ^1.0.0

# For information on the generic Dart part of this file, see
the# following page:
https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter.
flutter:
 assets:
   - assets/images/photo.jpg

 # 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

 # An image asset can refer to one or more resolution-specific "variants",
 see # https://flutter.dev/assets-and-images/#resolution-aware.

 # For details regarding adding assets from package
 dependencies, see # https://flutter.dev/assets-and-
 images/#from-packages

 # To add custom fonts to your application, add a fonts section
 here, # in this "flutter" section. Each entry in this list should
 have a
 # "family" key with the font family name, and a "fonts" key
```

with a # list giving the asset and other descriptors for the font.
For
# example:

```
# fonts:
#  - family: Schyler
#     fonts:
#       - asset: fonts/Schyler-
Regular.ttf#      - asset:
fonts/Schyler-Italic.ttf
#         style: italic
#  - family: Trajan
Pro#   fonts:
#       - asset: fonts/TrajanPro.ttf
#       - asset:
fonts/TrajanPro_Bold.ttf#
        weight: 700
#
# For details regarding fonts from package
dependencies,# see https://flutter.dev/custom-
fonts/#from-packages
```

Exp 6: To Apply navigation, routing and gesture in Flutter App

```
import
'package:flutter/material.dart'; void
main() => runApp(MyApp()); class
MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Flutter Demo',
home:
MyHomePage(),
);
}
}
class MyHomePage extends StatefulWidget {
@override
_MyHomePageState createState() => _MyHomePageState();
}
class MySecondPage extends StatefulWidget
{@override
_MySecondPageState createState() => _MySecondPageState();
}
class _MyHomePageState extends State<MyHomePage> {
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('Hello,My name is Rutuja . This is home-page'),
),
body: Center(
child: RaisedButton(
child: Text('Go to Second Screen'),
onPressed: () {
Navigator.push
(context,
MaterialPageRoute(builder: (context) => MySecondPage()),
);
},
),
),
);
}
}
class _MySecondPageState extends State<MySecondPage>
{@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('This is Second Screen after navigation...'),
),
body: Center(
child: RaisedButton(
child: Text('Go back to Home Screen !!'),
onPressed: () {
Navigator.pop(context);
},
),
),
);
}
}
```