7. Laboratory Exercise

A. Program

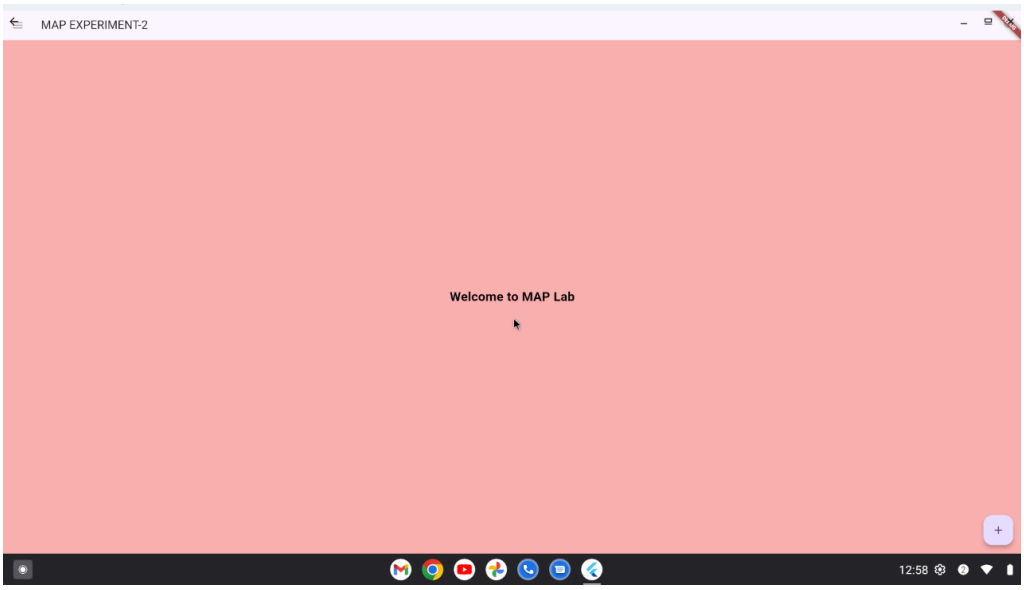
i. Design UI for mobile app using the following widgets.

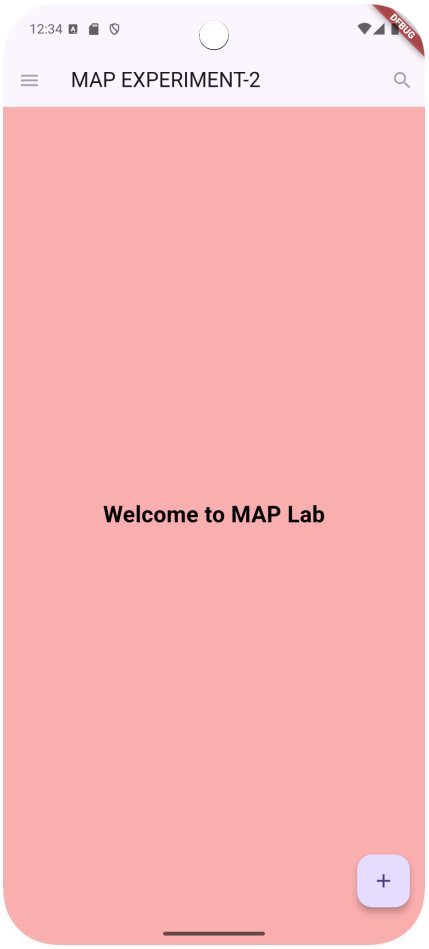
MaterialApp, Scaffold, AppBar, Text, Center, FloatingActionButton, TextStyle.

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

void main() {
 runApp(
   const MaterialApp(
     title: 'Flutter Tutorial',
     home: TutorialHome(),
   ),
 );
}

class TutorialHome extends StatelessWidget {
 const TutorialHome({super.key});
A
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     // Set the background color to baby pink
     backgroundColor: Color(0xFFFBB0B0), // Light pink color
     appBar: AppBar(
       leading: const IconButton(
         icon: Icon(Icons.menu),
         tooltip: 'Navigation menu',
         onPressed: null,
       ),
       title: const Text('MAP EXPERIMENT-2'),
       actions: const [
         IconButton(
           icon: Icon(Icons.search),
           tooltip: 'Search',
           onPressed: null,
         ),
       ],
     ),
     body: const Center(
       child: Text(
         'Welcome to MAP Lab', // Add body text
         style: TextStyle(
           fontSize: 24, // Set font size
           fontWeight: FontWeight.bold, // Make it bold
           color: Colors.black, // Text color
         ),
       ),
```

```
      ),
      floatingActionButton: const FloatingActionButton(
        tooltip: 'Add',
        onPressed: null,
        child: Icon(Icons.add),
      ),
    );
  }
}
```

Output:





Name: Shruti Kadam Roll No.:46

8. Post-Experiments Exercise

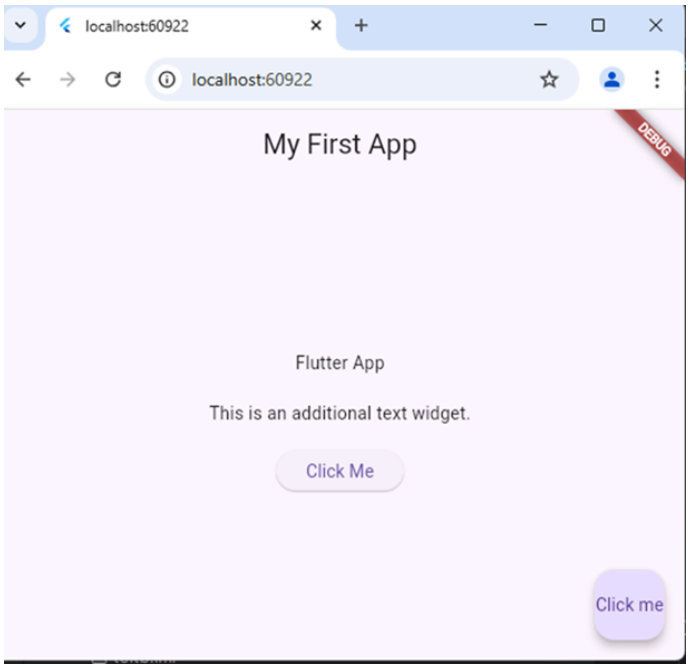
A. Questions:

1. Modify the app UI to include two more widgets and show the output.

CODE:

```
import 'package:flutter/material.dart';
void main() => runApp(MaterialApp(
 home: Scaffold(
  appBar: AppBar(
   title: Text('My First App'),
   centerTitle: true,
  ),
  body: Center(
   child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
     Text('Flutter App'),
     SizedBox(height: 20), // Adds some space between the
widgets
     Text('This is an additional text widget.'),
     SizedBox(height: 20), // Adds some space between the
widgets
     ElevatedButton(
      onPressed: () {
      print('Button Pressed!');
      },
      child: Text('Click Me'),
     ),
    ],
   ),
  ),
  floatingActionButton: FloatingActionButton(
   onPressed: () {
    // Action to perform when the FAB is pressed
    print('Floating Action Button Pressed!');
   },
   child: Text("Click me"),
  ),
 ),
));
```

OUTPUT:

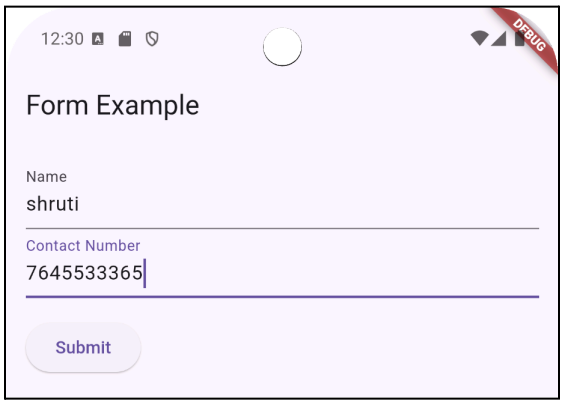


A. Program

i. Create a form for the mobile app having two text fields: person's name and contact number. The form should be able to accept the input when the user submit the form. Also validate the form for null values and display appropriate messages.

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


void main() {
 runApp(MyApp());
}

class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: Scaffold(
       appBar: AppBar(title: Text('Form Example')),
       body: Padding(
         padding: const EdgeInsets.all(16.0),
         child: MyForm(),
       ),
     ),
   );
 }
}

class MyForm extends StatefulWidget {
 @override
 _MyFormState createState() => _MyFormState();
}

class _MyFormState extends State<MyForm> {
 final _formKey = GlobalKey<FormState>();
 final _nameController = TextEditingController();
 final _contactController = TextEditingController();

 // Function to handle form submission
 void _submitForm() {
   if (_formKey.currentState?.validate() ?? false) {
     ScaffoldMessenger.of(context).showSnackBar(
       SnackBar(content: Text('Form Submitted Successfully')),
     );
   }
 }

 @override
 Widget build(BuildContext context) {
   return Form(
```

```
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          TextFormField(
            controller: _nameController,
            decoration: InputDecoration(labelText: 'Name'),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter a name';
              }
              return null;
            },
          ),
          TextFormField(
            controller: _contactController,
            decoration: InputDecoration(labelText: 'Contact Number'),
            keyboardType: TextInputType.phone,
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter a contact number';
              } else if (!RegExp(r'^[0-9]+$').hasMatch(value)) {
                return 'Please enter a valid contact number';
              }
              return null;
            },
          ),
          Padding(
            padding: const EdgeInsets.symmetric(vertical: 16.0),
            child: ElevatedButton(
              onPressed: _submitForm,
              child: Text('Submit'),
            ),
          ),
        ],
      ),
    );
  }
}
```



8. Post-Experiment Exercise

```
import 'package:flutter/material.dart';
void main() {
 runApp(MyApp());
}
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: Scaffold(
       backgroundColor: Color(0xFFFBB0B0), // Baby pink background color
       appBar: AppBar(title: Text('Form Example')),
       body: Padding(
         padding: const EdgeInsets.all(16.0),
         child: MyForm(),
       ),
     ),
   );
 }
}
class MyForm extends StatefulWidget {
 @override
 _MyFormState createState() => _MyFormState();
}
class _MyFormState extends State<MyForm> {
 final _formKey = GlobalKey<FormState>();
 final _nameController = TextEditingController();
 final _contactController = TextEditingController();
 final _emailController = TextEditingController();
 String? _gender; // To store selected gender
 // Function to handle form submission
 void _submitForm() {
   if (_formKey.currentState?.validate() ?? false) {
     if (_gender == null) {
       ScaffoldMessenger.of(context).showSnackBar(
         SnackBar(content: Text('Please select a gender')),
       );
     } else {
       ScaffoldMessenger.of(context).showSnackBar(
         SnackBar(content: Text('Form Submitted Successfully')),
       );
     }
   }
```

```
}
@override
Widget build(BuildContext context) {
  return Form(
    key: _formKey,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        // Name field
        TextFormField(
          controller: _nameController,
          decoration: InputDecoration(labelText: 'Name'),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter a name';
            }
            return null;
          },
        ),
        // Contact number field
        TextFormField(
          controller: _contactController,
          decoration: InputDecoration(labelText: 'Contact Number'),
          keyboardType: TextInputType.phone,
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter a contact number';
            } else if (!RegExp(r'^[0-9]+$').hasMatch(value)) {
              return 'Please enter a valid contact number';
            }
            return null;
          },
        ),
        // Email field
        TextFormField(
          controller: _emailController,
          decoration: InputDecoration(labelText: 'Email'),
          keyboardType: TextInputType.emailAddress,
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter an email';
            } else if
(!RegExp(r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$').hasMatch(value)) {
              return 'Please enter a valid email address';
```

```
    }
    return null;
  },
),
// Gender field - Radio buttons
Row(
  children: <Widget>[
    Text('Gender: '),
    Row(
      children: <Widget>[
        Radio<String>(
          value: 'Male',
          groupValue: _gender,
          onChanged: (String? value) {
            setState(() {
              _gender = value;
            });
          },
        ),
        Text('Male'),
        Radio<String>(
          value: 'Female',
          groupValue: _gender,
          onChanged: (String? value) {
            setState(() {
              _gender = value;
            });
          },
        ),
        Text('Female'),
        Radio<String>(
          value: 'Other',
          groupValue: _gender,
          onChanged: (String? value) {
            setState(() {
              _gender = value;
            });
          },
        ),
        Text('Other'),
      ],
    ),
  ],
),
```

```
        Padding(
          padding: const EdgeInsets.symmetric(vertical: 16.0),
          child: ElevatedButton(
            onPressed: _submitForm,
            child: Text('Submit'),
          ),
        ),
      ],
    ),
  );
 }
}
```

OUTPUT: