1b) Write a simple Dart program to understand the language basics.

```
void main() {
  var firstName = "John";
  var lastName = "Doe";
  print("Full name is $firstName $lastName");
}
```

Output : Full name is John Doe

```
void main() {
  int a = 5;
  int b = 3;
  print("The sum is ${a + b}");
}
```

Output : The sum is 8

```
import 'dart:io';

void main() {
  print("Enter number:");
  int? number = int.parse(stdin.readLineSync()!); // Reading input and converting it to int
  print("The entered number is ${number}");
}
```

Output: Enter number:
50
The entered number is 50

2a) Explore various Flutter widgets (Text, Image, Container, etc.).

```
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("Welcome")),
        body: Center(child: Text("Hello World!!")),
      ),
    );
  }}
```

Images:

```
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("Image Example")),
        body: Center(
          child: Image.asset('assets/images/sample.jpg'),
        ),
      ),
    );
  }
}
```

Container:

```
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("Container Example")),
        body: Center(
          child: Container(
            padding: EdgeInsets.all(20),
            decoration: BoxDecoration(
              border: Border.all(color: Colors.blue, width: 2),
            ),
            child: Text("I am inside a container!"),
          ),
        ),
      ),
    );
  }
}
```

2b) Implement different layout structures using Row, Column, and Stack widgets

Row Widget

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

void main() => runApp(MaterialApp(home: Scaffold(body: RowWidget())));

class RowWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        Container(color: Colors.green, child: Text("React.js", style: TextStyle(color: Colors.white))),
        Container(color: Colors.blue, child: Text("Flutter", style: TextStyle(color: Colors.white))),
        Container(color: Colors.red, child: Text("MySQL", style: TextStyle(color: Colors.white))),
      ],
    );
  }
}
```

Column Widget

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

void main() => runApp(MaterialApp(home: Scaffold(body: ColumnWidget())));

class ColumnWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      children: [
        Container(color: Colors.orange, child: Text("React.js", style: TextStyle(color: Colors.white))),
        Container(color: Colors.purple, child: Text("Flutter", style: TextStyle(color: Colors.white))),
        Container(color: Colors.teal, child: Text("MySQL", style: TextStyle(color: Colors.white))),
      ],
    );
  }
}
```

Stack Widget

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

void main() => runApp(MaterialApp(home: Scaffold(body: StackWidget())));

class StackWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Stack(
      alignment: Alignment.center,
      children: [
        Container(width: 200, height: 200, color: Colors.yellow),
        Container(width: 150, height: 150, color: Colors.red),
        Text("Stack", style: TextStyle(color: Colors.white)),
      ],
    );
  }
}
```

3. a) Design a responsive UI that adapts to different screen sizes.

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Responsive UI")),
        body: LayoutBuilder(
          builder: (context, constraints) {
            if (constraints.maxWidth < 600) {
              return const Center(child: Text("Mobile Screen", style: TextStyle(fontSize: 20)));
            } else if (constraints.maxWidth < 1200) {
              return const Center(child: Text("Tablet Screen", style: TextStyle(fontSize: 24)));
            } else {
              return const Center(child: Text("Desktop Screen", style: TextStyle(fontSize: 28)));
            }
          },
        ),
      ),
    );
  }
}
```

3 b) Implement media queries and breakpoints for responsiveness.

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("Media Query Example")),
        body: Builder(
          builder: (context) {
            double width = MediaQuery.of(context).size.width;
            String text = width < 600
                ? "Mobile Screen"
                : width < 1200
                    ? "Tablet Screen"
                    : "Desktop Screen";
            return Center(child: Text(text, style: const TextStyle(fontSize: 24)));
          },
        ),
      ),
    );
  }
}
```

4. a) Set up navigation between different screens using Navigator.

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

void main() => runApp(MaterialApp(home: Screen1()));

class Screen1 extends StatelessWidget {
  @override
  Widget build(BuildContext context) => Scaffold(
        appBar: AppBar(title: const Text("Screen 1")),
        body: Center(
          child: ElevatedButton(
            onPressed: () => Navigator.push(
              context,
              MaterialPageRoute(builder: (_) => Screen2()),
            ),
            child: const Text("Go to Screen 2"),
          ),
        ),
      );
}

class Screen2 extends StatelessWidget {
  @override
  Widget build(BuildContext context) => Scaffold(
        appBar: AppBar(title: const Text("Screen 2")),
        body: Center(
          child: ElevatedButton(
            onPressed: () => Navigator.pop(context),
            child: const Text("Back to Screen 1"),
          ),
        ),
      );
}
```

4b) Implement navigation with named routes.

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

void main() {
  runApp(MaterialApp(
    initialRoute: '/',
    routes: {
      '/': (context) => Scaffold(
        appBar: AppBar(title: Text('Home')),
        body: Center(
          child: ElevatedButton(
            onPressed: () => Navigator.pushNamed(context, '/about'),
            child: Text('Go to About'),
          ),
        ),
      ),
      '/about': (context) => Scaffold(
        appBar: AppBar(title: Text('About')),
        body: Center(
          child: ElevatedButton(
            onPressed: () => Navigator.pop(context),
            child: Text('Go back'),
          ),
        ),
      ),
    },
  ));
}
```

5. a) Learn about stateful and stateless widgets.

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

void main() => runApp(MaterialApp(home: Scaffold(
  appBar: AppBar(title: Text('Stateful Example')),
  body: StatefulCounter(),
)));

class StatefulCounter extends StatefulWidget {
  @override
  _StatefulCounterState createState() => _StatefulCounterState();
}

class _StatefulCounterState extends State<StatefulCounter> {
  int counter = 0;

  @override
  Widget build(BuildContext context) {
    return Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text('Counter: $counter'),
          ElevatedButton(onPressed: () => setState(() => counter++), child: Text('Increment')),
        ],
      ),
    );
  }
}
```

5 b) Implement state management using set State and Provider.

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

void main() => runApp(ChangeNotifierProvider(
  create: (_) => CounterModel(),
  child: MaterialApp(home: CounterPage()),
));

class CounterModel extends ChangeNotifier {
  int _counter = 0;
  int get counter => _counter;
  void increment() {
    _counter++;
    notifyListeners();
  }
}

class CounterPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Counter')),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text('Counter: ${Provider.of<CounterModel>(context).counter}'),
            ElevatedButton(
              onPressed: () => Provider.of<CounterModel>(context, listen: false).increment(),
              child: Text('Increment'),
            ),
          ],
        ),
      ),
    );
  }
}
```

6. a) Create custom widgets for specific UI elements.

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

void main() => runApp(MaterialApp(
  home: Scaffold(
    appBar: AppBar(title: Text('Custom Button Example')),
    body: Center(
      child: ElevatedButton(
        onPressed: () => print('Button Pressed'),
        child: Text('Click Me'),
      ),
    ),
  ),
));
```

6b) Apply styling using themes and custom styles.

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

void main() => runApp(MaterialApp(
  title: 'Custom Themes',
  theme: ThemeData(
    useMaterial3: true,
    colorScheme: ColorScheme.fromSeed(seedColor: Colors.purple, brightness: Brightness.dark),
    textTheme: TextTheme(
      titleLarge: GoogleFonts.oswald(fontSize: 30, fontStyle: FontStyle.italic),
      bodyMedium: GoogleFonts.merriweather(),
    ),
  ),
  home: Scaffold(
    appBar: AppBar(
      title: Text('Custom Themes', style: GoogleFonts.oswald(fontSize: 30, fontStyle: FontStyle.italic)),
      backgroundColor: Colors.purple,
    ),
    body: Center(
      child: Container(
        color: Colors.purple[700],
        padding: EdgeInsets.all(12),
        child: Text(
          'Text with a background color',
          style: GoogleFonts.merriweather(color: Colors.white),
        ),
      ),
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () {},
      child: Icon(Icons.add),
      backgroundColor: Colors.purple[700],
    ),
  ),
));
```