EXERCISE 1: An application that uses GUI components, Fonts

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

void main() {
  runApp(MaterialApp(
    home: Home(),
  ));
}

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
          title: Text("Hello World"),
          centerTitle: true,
          backgroundColor: Color.fromARGB(255, 34, 126, 255)),
      body: Center(
        child: Text(
          "Hello World",
          style: TextStyle(
            fontSize: 45.0,
            fontWeight: FontWeight.bold,
            letterSpacing: 2.0,
            color: Colors.blueGrey[600],
            fontFamily: 'Arial',
          ),
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () {},
        child: Text("+"),
        backgroundColor: Color.fromARGB(255, 34, 126, 255),
      ),
    );
  }
}
```

EXERCISE 2: An application that uses Layout Managers and Event

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

void main() {
  runApp(const MaterialApp(
    home: Home(),
  ));
}

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

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  int projects = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color.fromARGB(255, 223, 223, 225),
      appBar: AppBar(
        title: Text("Sample layout"),
        backgroundColor: Colors.black12,
        centerTitle: true,
        elevation: 0.0,
      ),
      body: Padding(
        padding: EdgeInsets.fromLTRB(30.0, 40.0, 30.0, 0.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Center(
              child: CircleAvatar(
                backgroundImage: AssetImage('assets/flutter.png'),
                radius: 50.0,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            Text(
              "Sample layout",
              style: TextStyle(
                color: Colors.black,
                letterSpacing: 2.0,
              ),
            ),
            SizedBox(
              height: 10.0,
            ),
            Text(
              "$projects",
              style: TextStyle(
                color: Colors.blue,
                letterSpacing: 2.0,
                fontSize: 28.0,
                fontWeight: FontWeight.bold,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            ElevatedButton(
              onPressed: () {
                setState(() {
                  projects++;
                });
              },
              onLongPress: () {
                setState(() {
                  projects *= 2;
                });
              },
              child: Icon(
                Icons.add,
              ),
              style: ElevatedButton.styleFrom(
                primary: Colors.green,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            ElevatedButton(
              onPressed: () {
                setState(() {
                  if (projects > 0) projects--;
                });
              },
              onLongPress: () {
                setState(() {
                  if (projects > 0) projects ~/= 2;
                });
```

```
              },
              child: Icon(
                Icons.remove,
              ),
              style: ElevatedButton.styleFrom(
                primary: Colors.deepOrange,
              ),
            ),
            SizedBox(
              height: 20.0,
            ),
            Row(
              children: [
                SizedBox(
                  width: 20.0,
                ),
              ],
            )
          ],
        ),
      ),
    );
  }
}
```

EXERCISE 3: Creation of Calculator Application

```
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(
      title: 'Calculator',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      debugShowCheckedModeBanner: false,
      home: const MyHomePage(),
    );
  }
}
```

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

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  String output = "0";

  String _output = "0";
  double num1 = 0.0;
  double num2 = 0.0;
  String operand = "";

  buttonPressed(String buttonText) {
    if (buttonText == "CLEAR") {
      _output = "0";
      num1 = 0.0;
      num2 = 0.0;
      operand = "";
    } else if (buttonText == "+" ||
        buttonText == "-" ||
        buttonText == "/" ||
        buttonText == "X") {
      num1 = double.parse(output);

      operand = buttonText;

      _output = "0";
    } else if (buttonText == ".") {
      if (_output.contains(".")) {
        return;
      } else {
        _output = _output + buttonText;
      }
    } else if (buttonText == "=") {
      num2 = double.parse(output);

      if (operand == "+") {
        _output = (num1 + num2).toString();
      }
      if (operand == "-") {
        _output = (num1 - num2).toString();
      }
      if (operand == "X") {
        _output = (num1 * num2).toString();
      }
      if (operand == "/") {
        _output = (num1 / num2).toString();
      }

      num1 = 0.0;
      num2 = 0.0;
      operand = "";
    } else {
      _output = _output + buttonText;
    }

    setState(() {
      output = double.parse(_output).toStringAsFixed(2);
    });
  }

  Widget buildButton(String buttonText) {
    return Expanded(
        child: OutlinedButton(
      style: OutlinedButton.styleFrom(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(0.0),
        ),

        side: const BorderSide(width: 1, color: Colors.grey),
        minimumSize: const Size.fromHeight(
            50.0), // Set this padding: EdgeInsets.zero, // and this
      ),
      child: Text(
        buttonText,
        style: const TextStyle(fontSize: 20.0, fontWeight: FontWeight.bold),
      ),
      onPressed: () => buttonPressed(buttonText),
    ));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: const Text("Calculator"),
        ),
        body: Column(
          children: <Widget>[
            const Expanded(
              child: Divider(
                color: Colors.white,
              ),
            ),
            Column(children: [
              Container(
                  alignment: Alignment.centerRight,
                  padding: const EdgeInsets.symmetric(
                      vertical: 24.0, horizontal: 12.0),
                  child: Text(output,
                      style: const TextStyle(
                        fontSize: 48.0,
                        fontWeight: FontWeight.bold,
                      ))),
              Row(children: [
                buildButton("7"),
                buildButton("8"),
                buildButton("9"),
                buildButton("/")
              ]),
              Row(children: [
                buildButton("4"),
                buildButton("5"),
                buildButton("6"),
                buildButton("X")
              ]),
              Row(children: [
                buildButton("1"),
                buildButton("2"),
                buildButton("3"),
                buildButton("-")
              ]),
              Row(children: [
                buildButton("."),
                buildButton("0"),
                buildButton("00"),
                buildButton("+")
              ]),
              Row(children: [
                buildButton("CLEAR"),
                buildButton("="),
              ])
            ])
          ],
        ));
  }
}
```

EXERCISE 4: An application that draws basic graphical primitives

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

final Color darkBlue = Color.fromARGB(255, 18, 32, 47);
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark().copyWith(scaffoldBackgroundColor: darkBlue),
      debugShowCheckedModeBanner: false,
      home: Scaffold(
// Outer white container with padding
        body: Container(
          color: Colors.black,
          padding: EdgeInsets.symmetric(horizontal: 40, vertical: 80),
// Inner yellow container
          child: Container(
// pass double.infinity to prevent shrinking of the painter area to 0.
            width: double.infinity,
            height: double.infinity,
            color: Color.fromARGB(255, 126, 125, 125),
            child: CustomPaint(painter: FaceOutlinePainter()),
          ),
        ),
      ),
    );
  }
}

class FaceOutlinePainter extends CustomPainter {
  @override
  void paint(Canvas canvas, Size size) {
    final paint = Paint();
    paint.style = PaintingStyle.stroke;
    paint.strokeWidth = 4.0;
    paint.color = Color.fromARGB(255, 244, 67, 54);

    canvas.drawOval(
      Rect.fromLTWH(size.width - 120, 40, 100, 100),
      paint,
    );

    canvas.drawRect(
      Rect.fromLTWH(20, 40, 100, 100),
      paint,
    );

    final mouth = Path();
    mouth.moveTo(size.width * 0.8, size.height * 0.6);
    mouth.arcToPoint(
      Offset(size.width * 0.2, size.height * 0.6),
      radius: Radius.circular(150),
    );
    canvas.drawPath(mouth, paint);
  }

  bool shouldRepaint(FaceOutlinePainter oldDelegate) => false;
}
```

EXERCISE 5: An application that uses database for persistent

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart'; void main() => runApp(
MaterialApp(
theme: ThemeData( brightness: Brightness.light, primaryColor: Colors.blue,
accentColor: Colors.orange),
home: MyApp(),
),
);



class MyApp extends StatefulWidget { @override
_MyAppState createState() => _MyAppState();
}



class _MyAppState extends State<MyApp> { List todos = List();
String input = ''; createTodos() {
DocumentReference documentReference =
Firestore.instance.collection('MyTodos').document(input);
Map<String, String> todos = {'todoTitle': input};
documentReference.setData(todos).whenComplete(() { print('$input created');
});

}
```

```
deleteTodos() {} @override
Widget build(BuildContext context) { return Scaffold(
appBar: AppBar(
title: Text('To-Do List'),
),
floatingActionButton: FloatingActionButton( child: Icon(Icons.add),
onPressed: () { showDialog( context: context,
builder: (BuildContext context) { return AlertDialog(
title: Text('Add To-Do'), content: TextField( onChanged: (String value) {
input = value;
},
),
actions: <Widget>[ FlatButton( onPressed: () {
createTodos(); Navigator.of(context).pop();
},
child: Text('Add'),
),
],
);

},
);
},
),
body: StreamBuilder(
stream: Firestore.instance.collection('MyTodos').snapshots(), builder: (context,
snapshots) {
return ListView.builder( shrinkWrap: true,
itemCount: snapshots.data.documents.length, itemBuilder: (BuildContext context,
int index) { DocumentSnapshot documentSnapshot =
snapshots.data.documents[index]; return Dismissible(
key: Key(index.toString()), child: Card(
elevation: 4.0,
margin: EdgeInsets.all(8.0), shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(8),
),
child: ListTile(
title: Text(documentSnapshot['todoTitle']), trailing: IconButton(
icon: Icon( Icons.delete, color: Colors.red,
),
onPressed: () {
setState(() { todos.removeAt(index);
});
},
);
}));
}
}



),
));

      }),
```

EXERCISE 6: An application that makes use of RSS feed

```
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:webfeed/webfeed.dart';
import 'package:http/http.dart' as http;
import 'package:url_launcher/url_launcher.dart';

void main() {
  runApp(const RSSDemo());
}

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(title: "RSS Feed", home: RSSMainPicture());
  }
}

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

  @override
  State<RSSMainPicture> createState() => _RSSMainPictureState();
```

```
}

class _RSSMainPictureState extends State<RSSMainPicture> {
  late Future<RssFeed> result;
  Future<RssFeed> giver() async {
    var response = await http.get(Uri.parse(
        "https://www.espncricinfo.com/rss/content/story/feeds/0.xml"));
    var channel = RssFeed.parse(response.body);
    return channel;
  }

  @override
  void initState() {
    super.initState();
    result = giver();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("News"),
        actions: [
          IconButton(
              onPressed: () => result = giver(),
              icon: const Icon(Icons.refresh_rounded)),
        ],
      ),
      body: FutureBuilder<RssFeed?>(
        future: result,
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            if (kDebugMode) {
              print("Error");
            }
            return Container();
          } else if (snapshot.connectionState == ConnectionState.waiting) {
            return const Center(
              child: CircularProgressIndicator(),
            );
          } else if (snapshot.hasData) {
            var feed = snapshot.data!;
            var items = feed.items;
            return ListView.builder(
              itemCount: items?.length,
              itemBuilder: (context, index) {
                var item = items![index];
                return GestureDetector(
                  onTap: () async {
                    if (!await launchUrl(Uri.parse(item.link!))) {
                      throw 'Could not launch ${item.link}';
                    }
                  },
                  child: ListTile(
                    // leading: CachedNetworkImage(
                    //   imageUrl: mediaImage!,
                    //   progressIndicatorBuilder: (context, url, downloadProgress) =>
                    //       CircularProgressIndicator(value: downloadProgress.progress),
                    //   errorWidget: (context, url, error) => const Icon(Icons.error),
                    // ),
                    title: Text(item.title!),
                    subtitle: Text("${item.pubDate!}"),
                  ),
                );
              },
            );
          }
          return Container();
        },
      ),
    );
  }
}
```

EXERCISE 7: An application that implements multithreading

main.dart

```
import 'package:expt7/pages/home.dart';
import 'package:flutter/material.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        brightness: Brightness.dark,
      ),
      home: const Home(),
    );
  }
}
```

<u>home.dart</u>

```
import 'dart:async';
import 'dart:math';

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

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

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  int randint=99;
  static FutureOr<int> randGen(int cal){
    var rng = Random();
    return rng.nextInt(100);
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Multithreading App",
        ),
        centerTitle: true,
      ),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
              Text(
                "Random Number: ",
                style: TextStyle(
                  fontSize: 20.0,
                ),
              ),
              Text(
                "${randint}",
                style: TextStyle(
                  fontSize: 20.0,
                ),
              ),
            ],
          ),
          SizedBox(
            height: 20.0,
          ),
          TextButton(
            onPressed: () async{
              int result = await compute(randGen,randint);
              setState(() {
                randint = result;
              });
            },
            child: Text(
              "Press Me!",
              style: TextStyle(
                fontSize: 20.0,
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

EXERCISE 8: An application that uses GPS location information

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

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.pink,
      ),
      home: const Home(),
    );
  }
}
class Home extends StatelessWidget {
  const Home({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text(
          "My Location"
        ),
        centerTitle: true,
      ),
      body: const LocationInfo(

      ),
      floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
    );
  }
}

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

  @override
  State<LocationInfo> createState() => _LocationInfoState();
}

class _LocationInfoState extends State<LocationInfo> {
  String _myLoc ="My Location";
  Location location=new Location();
  late bool _serviceEnabled;
  late PermissionStatus _permissionGranted;
  late LocationData _locationData;
  bool _isListenLocation =false, _isGetLocation = false;

  @override
  Widget build(BuildContext context) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: <Widget>[
        const SizedBox(
          height: 20.0,
        ),
        const Icon(
          Icons.location_pin,
        ),
        const SizedBox(
          height: 20.0,
        ),
        Center(
          child: Text(
            "$_myLoc",
            style: TextStyle(
              fontSize: 20.0,
            ),
          ),
        ),
        const SizedBox(
          height: 20.0,
        ),
        FloatingActionButton(
            child: Icon(
              Icons.location_on_sharp,
            ),
            onPressed: updateLoc,
        ),
      ],
    );
  }
  void updateLoc() async{
    _serviceEnabled = await location.serviceEnabled();
    if(!_serviceEnabled){
      _serviceEnabled = await location.requestService();
      if(_serviceEnabled)
        return;
    }
    _permissionGranted = await location.hasPermission();
    if(_permissionGranted == PermissionStatus.denied){
      _permissionGranted = await location.requestPermission();
```

```
      if(_permissionGranted != PermissionStatus.granted)
        return;
    }
    _locationData = await location.getLocation();
    setState(() {
      _isGetLocation = true;
    });
    if(_isGetLocation){
      _myLoc="${_locationData.latitude} / ${_locationData.longitude}";
    }
  }
}
```

EXERCISE 9: An application that takes advantage of rich gesture-based UI handling

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

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Gyroscope and ui'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  double _dx = 0,
      _dy = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: StreamBuilder<GyroscopeEvent>(
        stream: SensorsPlatform.instance.gyroscopeEvents,
        builder: (context, snapshot) {
          if (snapshot.hasData) {
            _dy = _dy + snapshot.data!.y * 10;
            _dx = _dx + snapshot.data!.x * 10;
          }
          return Stack(
            children: [
              Positioned(
                top: _dy,
                left: _dx,
                child: GestureDetector(
                  onPanUpdate: (details) {
                    setState(() {
                      _dy = max(0, _dy + details.delta.dy);
                      _dx = max(0, _dx + details.delta.dx);
                    });
                  },
                  child: const CircleAvatar(),
                ),
              )
            ],
          );
        },
      ),
    );
  }
}
```

EXERCISE 10: An application that creates an alert upon user action

```
import 'package:expt10/pages/home.dart';
import 'package:flutter/material.dart';

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Experiment 10',
      theme: ThemeData.dark(),
      home: const Home(),
    );
  }
}
```

home.dart

```
import 'package:expt10/services/local_notification_service.dart';
import 'package:flutter/material.dart';

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

  @override
```

```
 State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
 late final LocalNotificationService service;
 @override
 void initState(){
  service = LocalNotificationService();
  service.initialize();
  super.initState();
 }
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: const Text(
     "Local Notifications Expt"
    ),
    backgroundColor: const Color(0xff006473),
    centerTitle: true,
   ),
   body: Padding(
    padding: EdgeInsets.all(MediaQuery.of(context).size.width*0.25),
    child: Column(
     children: <Widget>[
      TextButton(
        onPressed: () async {
         await service.showNotification(
           id: 0,
           title: "Sample Notification",
           body: "Sample Body"
         );
        },
        child: const Text(
         "Get an instant Notification"
        ),
      ),
      TextButton(
       onPressed: () async {
        await service.showScheduledNotification(
          id: 0,
          title: "Sample Notification",
          body: "Sample Body",
          seconds: 4,
        );
       },
       child: const Text(
                  "Get a delayed Notification"
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

local_notification_service.dart

```
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:timezone/timezone.dart' as tz;
import 'package:timezone/data/latest.dart' as tz;

class LocalNotificationService {
  LocalNotificationService();

  final _localNotificationService = FlutterLocalNotificationsPlugin();

  Future<void> initialize() async{
    tz.initializeTimeZones();
    const AndroidInitializationSettings androidInitializationSettings =
    AndroidInitializationSettings('ic_stat_assistant_navigation');

    const DarwinInitializationSettings iosInitializationSettings =
        DarwinInitializationSettings(
          requestAlertPermission: true,
          requestBadgePermission: true,
          requestSoundPermission: true,
        );
    const InitializationSettings settings = InitializationSettings(
        android: androidInitializationSettings,
        iOS: iosInitializationSettings
    );

    await _localNotificationService.initialize(settings);
  }
  Future<NotificationDetails> _notificationDetails() async{
    const AndroidNotificationDetails androidNotificationDetails =
AndroidNotificationDetails(
        "channel_id", "channel_name",
      channelDescription: "Description",
      importance: Importance.max,
      priority: Priority.max,
      playSound: true,
    );
    const DarwinNotificationDetails darwinNotificationDetails =
DarwinNotificationDetails();
    return const NotificationDetails(android: androidNotificationDetails,iOS:
darwinNotificationDetails);
  }
  Future<void> showNotification({
    required int id,
    required String title,
    required String body}) async{
      final details = await _notificationDetails();
      await _localNotificationService.show(id, title, body, details);
  }
  Future<void> showScheduledNotification({
    required int id,
    required String title,
    required String body,
    required int seconds
  }) async{
    final details = await _notificationDetails();
    await _localNotificationService.zonedSchedule(
      id,
      title,
      body,
      tz.TZDateTime.from(DateTime.now().add(Duration(seconds: seconds)), tz.local,),
      details,
      androidAllowWhileIdle: true,
      uiLocalNotificationDateInterpretation:
UILocalNotificationDateInterpretation.absoluteTime
    );
  }
}
```

EXERCISE 11: An application that creates an alarm clock

```
import 'package:flutter/material.dart';
import 'pages/home.dart';
void main() {
  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.cyan,
        brightness: Brightness.dark,
      ),
      home: const Home(),
    );
  }
}
```

home.dart

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

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

  @override
  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {
  TimeOfDay time= TimeOfDay(hour: 23, minute: 59);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Alarm Clock",
        ),
        centerTitle: true,
        elevation: 0.0,
        backgroundColor: Colors.cyan,
      ),
      body: Padding(
        padding: EdgeInsets.all(20),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  Text(
                    "Time set: ",
                    style: TextStyle(
                      fontSize: 30.0,
                    ),
                  ),
                  Text(
                    "${time.hour.toString().padLeft(2,'0')}:$
{time.minute.toString().padLeft(2,'0')}",
                    style: TextStyle(
                      fontSize: 30.0,
                      color: Colors.cyan,
                    ),
                  )
                ],
              ),
              SizedBox(
                height: 30.0,
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [
                  TextButton(
                    onPressed: () async{
                      TimeOfDay? newTime = await showTimePicker(
                          context: context,
                          initialTime: time,
                      );
                      if(newTime == null) return;
                      setState(() {
                        time = newTime;
                      });
                    },
                    child: Text(
                      "Edit Time",
                      style: TextStyle(
                        fontSize: 17.0,
                      ),
                    ),
                  ),
                  TextButton(
                    onPressed: () {
                      FlutterAlarmClock.createAlarm(time.hour,time.minute);
                    },
                    child: Text(
                      "Set Alarm",
                      style: TextStyle(
                        fontSize: 17.0,
                      ),
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

EXERCISE 12: An application that performs REST-based API calls

```
import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

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

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Api Calls',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(title: 'Codeforces Problem Set'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  late Future<Map<String,dynamic>> info;
  @override
  void initState(){
    info=giver();
    super.initState();
  }

  Future<Map<String,dynamic>> giver() async{
    var response = await http.get(Uri.parse("https://www.boredapi.com/api/activity"));
    Map<String,dynamic> result=json.decode(response.body);
    //print(result);
    return result;
  }
  @override
  Widget build(BuildContext context){
    return Scaffold(
      appBar: AppBar(
        title: const Text("Bored API"),
        actions: [
          IconButton(onPressed: ()=>setState(() {
            info=giver();
          }), icon: const Icon(Icons.refresh_rounded))
        ],
      ),
      body: FutureBuilder<Map<String,dynamic>>(
        future: info,
        builder: (context,snapshot){
          if(snapshot.connectionState==ConnectionState.waiting){
            return const Center(child: CircularProgressIndicator());
          }
          Map<String,dynamic> data={};
          if(snapshot.hasData){
            data=snapshot.data!;
            return Center(
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text("Activity: ${data["activity"]}"),
                  Text("Type: ${data["type"]}"),
                  Text("Participants: ${data["participants"]}"),
                  Text("Price: \$${data["price"]}"),
                ],
              ),
            );
          }
          return Container();
        },
      ),
    );
  }
}
```