
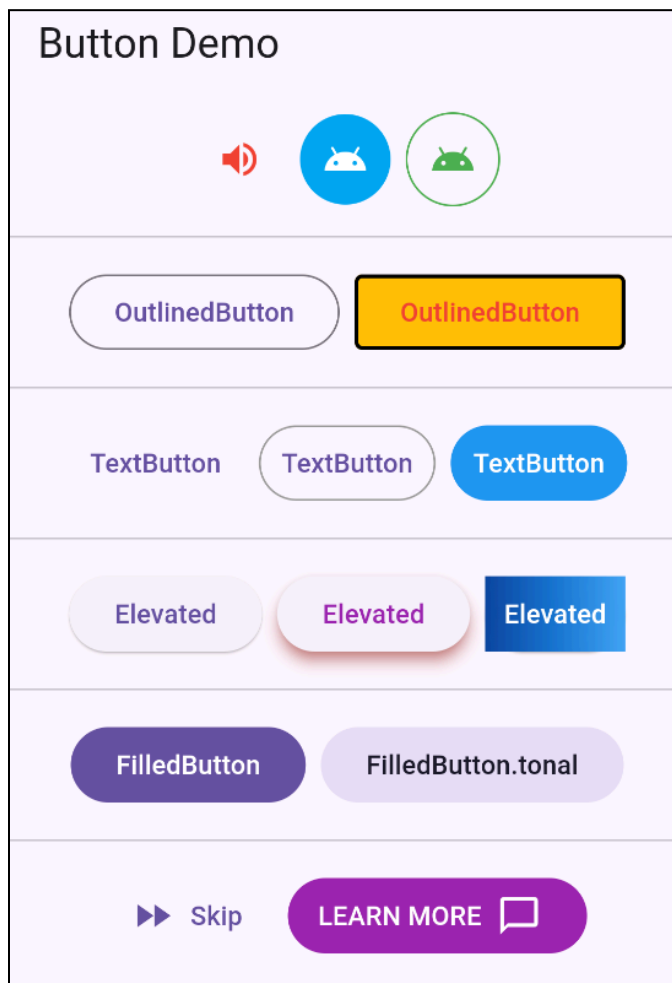


|   |                                     |   |  |
|---|-------------------------------------|---|--|
| <b>Mobile Programming Laboratory</b><br><b>Week 6: Widgets, UI design and Timer</b> |                                     |  |  |
| Name: Pongsapat Piniingam   | ID: 6631501079                      | Section: 02   |  |
| Date:   | Due: Midnight before the next class |   |  |

### Objective

- To experiment with UI widgets
- To practice UI design
- To apply Timer

### Exercise 1 Button demo



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

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

  Widget createIconButton() {
    return ButtonBar(
      alignment: MainAxisAlignment.center,
      children: [
        IconButton(
          icon: const Icon(Icons.volume_up),
          onPressed: () {},
          color: Colors.red,
        ),
        Ink(
          decoration: const ShapeDecoration(
            color: Colors.lightBlue,
            shape: CircleBorder(),
          ),
          child: IconButton(
            icon: const Icon(Icons.android),
            color: Colors.white,
            onPressed: () {},
          ),
        ),
        Ink(
          decoration: const ShapeDecoration(
            shape: CircleBorder(
              side: BorderSide(color: Colors.green),
            ),
          ),
          child: IconButton(
            icon: const Icon(Icons.android),
            color: Colors.green,
            onPressed: () {},
          ),
        ),
      ],
    );
  }
}
```

```
}
```

```
Widget createOutlinedButton() {  
  return ButtonBar(  
    alignment: MainAxisAlignment.center,  
    children: [  
      OutlinedButton(  
        onPressed: () {},  
        child: const Text('OutlinedButton'),  
      ),  
      OutlinedButton(  
        onPressed: () {},  
        style: OutlinedButton.styleFrom(  
          foregroundColor: Colors.red,  
          backgroundColor: Colors.amber,  
          shape: const RoundedRectangleBorder(  
            borderRadius: BorderRadius.all(Radius.circular(4))),  
            side: const BorderSide(color: Colors.black, width: 2),  
          ),  
          child: const Text('OutlinedButton'),  
        ),  
      ],  
    );  
}
```

```
Widget createTextButton() {  
  return ButtonBar(  
    alignment: MainAxisAlignment.center,  
    children: [  
      TextButton(  
        onPressed: () {},  
        child: const Text('TextButton'),  
      ),  
      TextButton(  
        onPressed: () {},  
        style: TextButton.styleFrom(  
          side: const BorderSide(color: Colors.grey, width: 1),  
        ),  
      ),  
    ],  
  );  
}
```

```

        child: const Text('TextButton'),
      ),
      TextButton(
        onPressed: () {},
        style: TextButton.styleFrom(
          foregroundColor: Colors.white,
          backgroundColor: Colors.blue,
        ),
        child: const Text('TextButton'),
      ),
    ],
  );
}

Widget createElevatedButton() {
  return ButtonBar(
    alignment: MainAxisAlignment.center,
    children: [
      ElevatedButton(
        onPressed: () {},
        child: const Text('Elevated'),
      ),
      ElevatedButton(
        onPressed: () {},
        style: ElevatedButton.styleFrom(
          foregroundColor: Colors.purple,
          elevation: 6,
          shadowColor: Colors.red,
        ),
        child: const Text('Elevated'),
      ),
      ElevatedButton(
        onPressed: () {},
        style: ElevatedButton.styleFrom(
          padding: const EdgeInsets.all(0.0),
        ),
        child: Container(
          decoration: const BoxDecoration(

```

```

        gradient: LinearGradient(
          colors: <Color>[
            Color(0xFF0D47A1),
            Color(0xFF1976D2),
            Color(0xFF42A5F5),
          ],
        ),
      ),
    ),
    padding: const EdgeInsets.all(10.0),
    child: const Text(
      'Elevated',
      style: TextStyle(color: Colors.white),
    ),
  ),
),
],
);
}

```

```

Widget createFilledButton() {
  return ButtonBar(
    alignment: MainAxisAlignment.center,
    children: [
      FilledButton(
        onPressed: () {},
        child: const Text('FilledButton'),
      ),
      FilledButton.tonal(
        onPressed: () {},
        child: const Text('FilledButton.tonal'),
      ),
    ],
  );
}

```

```

Widget createButtonWithIcon() {
  return ButtonBar(
    alignment: MainAxisAlignment.center,

```

```

        children: [
          TextButton.icon(
            onPressed: () {},
            icon: const Icon(Icons.fast_forward),
            label: const Text('Skip'),
          ),
          FilledButton.icon(
            onPressed: () {},
            icon: const Icon(Icons.chat_bubble_outline),
            iconAlignment: IconAlignment.end,
            label: const Text('LEARN MORE'),
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.purple,
            ),
          ),
        ],
      );
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Button Demo'),
    ),
    body: Center(
      child: Column(
        children: [
          createIconButton(),
          const Divider(),
          createOutlinedButton(),
          const Divider(),
          createTextButton(),
          const Divider(),
          createElevatedButton(),
          const Divider(),
          createFilledButton(),
          const Divider(),
        ],
      ),
    ),
  );
}

```

```
        createButtonWithIcon(),  
        ],  
    ),  
);  
}
```

## Images

Flutter supports images in different formats i.e. JPEG, WebP, GIF, animated WebP/GIF, PNG, BMP, and WBMP.

We can use both online and offline images. Let's try online first.

### Exercise 2 Online images

#### Image Demo



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

void main() {
  runApp(MaterialApp(home: ImageDemo()));
}
```



```

}

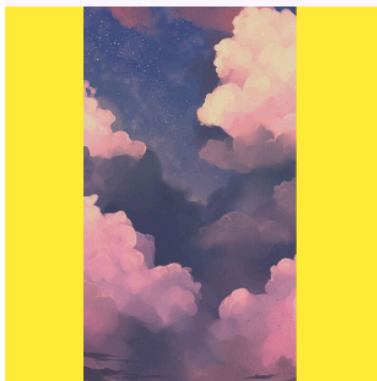
class ImageDemo extends StatelessWidget {
  const ImageDemo({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Image Demo')),
      body: Center(
        child: Image.network(
'https://i.pinimg.com/originals/2e/c6/b5/2ec6b5e14fe0cba0cb0aa5d2caeecccc6.jpg',
        ),
      ),
    );
  }
}

```

And you can specify either width or height to scale the image and keep the aspect ratio. Giving both width and height will try to fit the entire image by keeping aspect ratio. This is “contain” fit mode.

## Image Demo



```

import 'package:flutter/material.dart';

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

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Image Demo')),
    body: Container(
      width: 200,
      height: 200,
      color: Colors.yellow,
      child: Image.network(
        'https://i.pinimg.com/originals/2e/c6/b5/2ec6b5e14fe0cba0cb0aa5d2caecccc6.jpg',
      ),
    ),
  );
}

```

The above code is similar to:

```

body: Container(
  width: 200,
  height: 200,
  color: Colors.yellow,
  child: Image.network(
    'https://i.pinimg.com/originals/2e/c6/b5/2ec6b5e14fe0cba0cb0aa5d2caecccc6.jpg',
    fit: BoxFit.contain,
  ),
),

```

Note that for more efficient online image, please observe **Cached network image** at [https://pub.dev/packages/cached\\_network\\_image](https://pub.dev/packages/cached_network_image)

## Fit mode

See all fit modes at <https://api.flutter.dev/flutter/painting/BoxFit.html>

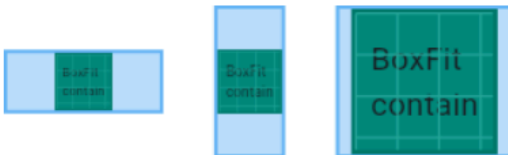
**fill** → const `BoxFit`

Fill the target box by distorting the source's aspect ratio.



**contain** → const `BoxFit`

As large as possible while still containing the source entirely within the target box.



**cover** → const `BoxFit`

As small as possible while still covering the entire target box.

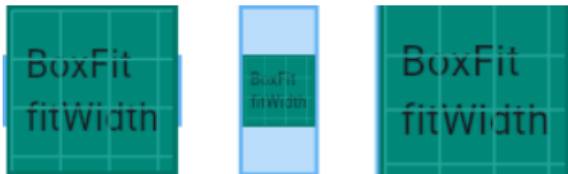
To actually clip the content, use `clipBehavior: Clip.hardEdge` alongside this in a `FittedBox`.



## **fitWidth** → const BoxFit

Make sure the full width of the source is shown, regardless of whether this means the source overflows the target box vertically.

To actually clip the content, use `clipBehavior: Clip.hardEdge` alongside this in a [FittedBox](#).



## **fitHeight** → const BoxFit

Make sure the full height of the source is shown, regardless of whether this means the source overflows the target box horizontally.

To actually clip the content, use `clipBehavior: Clip.hardEdge` alongside this in a [FittedBox](#).

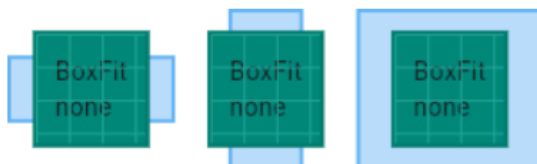


## **none** → const BoxFit

Align the source within the target box (by default, centering) and discard any portions of the source that lie outside the box.

The source image is not resized.

To actually clip the content, use `clipBehavior: Clip.hardEdge` alongside this in a [FittedBox](#).



## scaleDown → const BoxFit

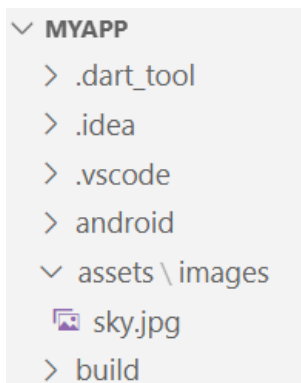
Align the source within the target box (by default, centering) and, if necessary, scale the source down to ensure that the source fits within the box.

This is the same as contain if that would shrink the image, otherwise it is the same as none.



## Offline images

Now to use an offline image, create an “assets/images” folder and put your images there. Assume that we have “sky.jpg” in that folder.



Next modify “pubspec.yaml”. Note that the file is pronounced like “yam-l” file.

```
# The following section is specific
flutter:

  # The following line ensures that
  # included with your application,
  # the material Icons class.
  uses-material-design: true

  # To add assets to your application
  assets:
    # - assets/icons/
    - assets/images/sky.jpg
```

If you have multiple images, you can provide only the folder name.

```
# To add assets to your
assets:
  # - assets/icons/
  - assets/images/
```

Then use the code below to load and display offline (asset) images.

```
body: Center(
  child: Image(image: AssetImage("assets/images/sky.jpg"),)
),
```

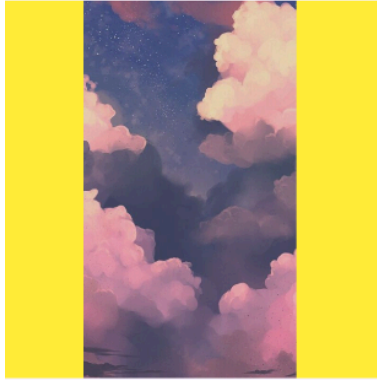
And you can shorten the code:

```
body: Center(
  child: Image.asset("assets/images/sky.jpg"),
),
```

And also we can set width, height and fit mode the same as Network image.

### Exercise 3 Offline image

#### Image Demo

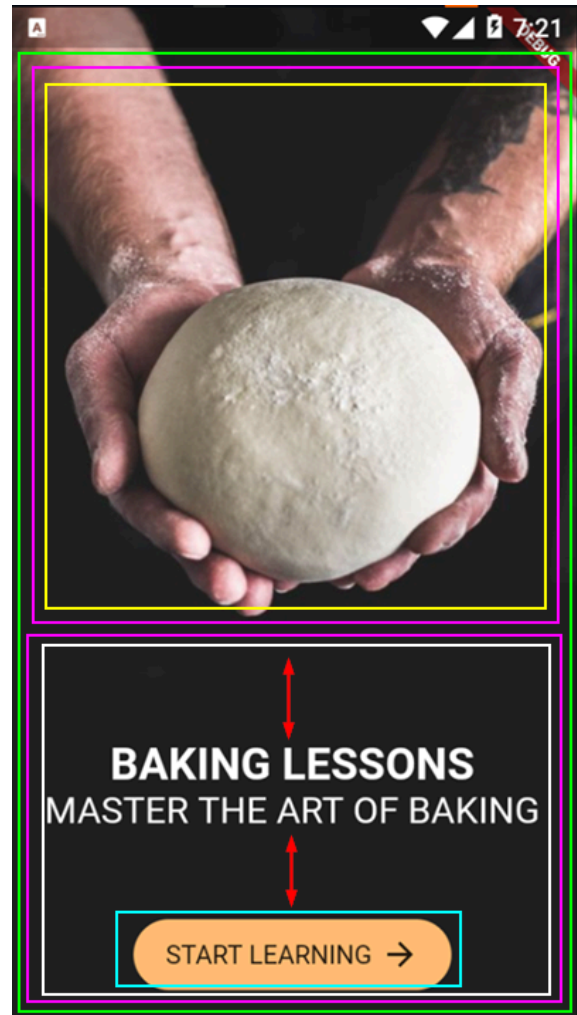


```
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Image Demo'),
      ),
      body: Container(
        color: Colors.yellow,
        child: Image.asset(
          'assets/images/sky.jpg',
          width: 200,
          height: 200,
          fit: BoxFit.contain,
        ),
      ),
    );
  }
}
```

**Exercise 4** Use the given assets to complete the design below.



Note:

- Green is Column
- Magenta is Expanded
- Yellow is Image
- White is Column
- Cyan is Button
- Red is Spacer

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

void main() {
  runApp(MaterialApp(home: WelcomeScreen()));
}
```



```

}

const Color bgColor = Color(0xFF202020);
const Color primaryColor = Color(0xFFFFBD73);

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: bgColor,
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: [
          Expanded(
            flex: 2,
            child: Image.asset('assets/images/baking.jpg', fit: BoxFit.cover),
          ),
          Expanded(
            child: Column(
              children: [
                const Spacer(flex: 1),
                Text(
                  'Baking lessons'.toUpperCase(),
                  style: TextStyle(
                    color: Colors.white,
                    fontWeight: FontWeight.bold,
                    fontSize: 28,
                  ),
                ),
                Text(
                  'Master the art of baking'.toUpperCase(),
                  style: TextStyle(color: Colors.white, fontSize: 22),
                ),
                const Spacer(flex: 2),
                FilledButton.icon(
                  onPressed: () {},

```


```
        icon: const Icon(Icons.arrow_forward),
        iconAlignment: IconAlignment.end,
        style: FilledButton.styleFrom(
          backgroundColor: primaryColor,
          foregroundColor: bgColor,
        ),
        label: Text(
          'Start learning'.toUpperCase(),
          style: TextStyle(fontSize: 18),
        ),
      ),
      const SizedBox(height: 16),
    ],
  ),
],
);
}
```

**Exercise 5** Use the given assets to complete the design below.

## Cooking Recipes


### Papaya Salad


Lorem ipsum dolor sit amet consectetur adipisicing elit. Soluta ducimus in modi illo ad ipsa non officiis. Ea placeat necessitatibus in aliquid ullam quasi porro vel dolores, dignissimos quisquam aspernatur.




★ ★ ★ ★ ★

3128 reviews

PREP:  
5 mins


COOK:  
10 mins

FEEDS:  
1-3

**Paste your code here.**


**Exercise 6** Use the given assets to complete the design below.

## Tourist Place




### Chiang Rai Clock Tower


Chiang Rai, Thailand



CALL



ROUTE



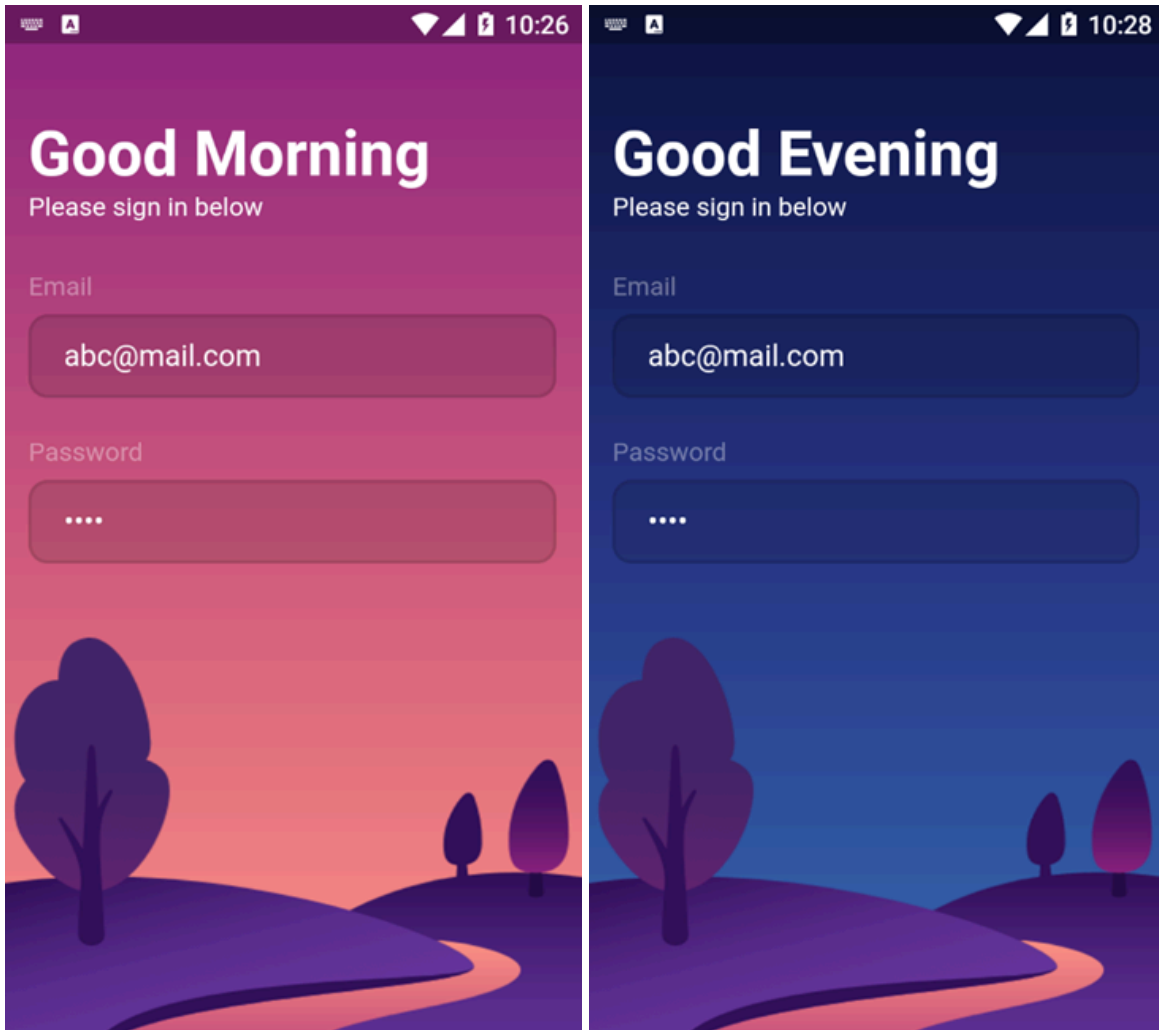
SHARE

★ 559

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur

**Paste your code here.**

## Exercise 7 Dynamic login page



```
// ref: https://github.com/abuanwar072/Flutter-Day-and-Night-Mood-Animation
import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(home: LoginDN()));
}

class LoginDN extends StatefulWidget {
  const LoginDN({super.key});
```

```

@override
State<LoginDN> createState() => _LoginDNState();
}

class _LoginDNState extends State<LoginDN> {
  List<Color> darkBgColors = [
    const Color(0xFF0D1441),
    const Color(0xFF283584),
    const Color(0xFF376AB2),
  ];

  List<Color> lightBgColors = [
    const Color(0xFF8C2480),
    const Color(0xFFCE587D),
    const Color(0xFFFF9485),
  ];

  @override
  Widget build(BuildContext context) {
    Size size = MediaQuery.of(context).size;
    DateTime now = DateTime.now();

    return Scaffold(
      body: Container(
        width: size.width,
        height: size.height,
        // or use infinity
        // width: double.infinity,
        // height: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: now.hour < 18 ? lightBgColors : darkBgColors,
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
          ),
        ),
      child: SafeArea(
        child: Column(

```

```

children: [
  Expanded(
    flex: 3,
    child: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Spacer(),
          Text(
            now.hour < 18 ? 'Good Morning' : 'Good Evening',
            style: Theme.of(context).textTheme.headlineLarge!
              .copyWith(
                fontWeight: FontWeight.bold,
                color: Colors.white,
              ),
          ),
          const Text(
            'Please sign in below',
            style: TextStyle(color: Colors.white),
          ),
          const Spacer(),
          const Text(
            'Email',
            style: TextStyle(color: Colors.white38),
          ),
          const SizedBox(height: 8),
          Container(
            padding: const EdgeInsets.symmetric(horizontal: 20),
            decoration: BoxDecoration(
              color: Colors.black12,
              borderRadius: BorderRadius.circular(12),
              border: Border.all(
                width: 2,
                color: Colors.black.withValues(alpha: 0.1),
              ),
            ),
            child: TextField(

```

```

        controller: TextEditingController(
          text: 'abc@mail.com',
        ),
        style: const TextStyle(color: Colors.white),
        decoration: const InputDecoration(
          enabledBorder: InputBorder.none,
          focusedBorder: InputBorder.none,
        ),
      ),
    ),
    const SizedBox(height: 24),
    const Text(
      'Password',
      style: TextStyle(color: Colors.white38),
    ),
    const SizedBox(height: 8),
    Container(
      padding: const EdgeInsets.symmetric(horizontal: 20),
      decoration: BoxDecoration(
        color: Colors.black12,
        borderRadius: BorderRadius.circular(12),
        border: Border.all(
          width: 2,
          color: Colors.black.withValues(alpha: 0.1),
        ),
      ),
      child: TextField(
        controller: TextEditingController(text: '1234'),
        obscureText: true,
        style: const TextStyle(color: Colors.white),
        decoration: const InputDecoration(
          enabledBorder: InputBorder.none,
          focusedBorder: InputBorder.none,
        ),
      ),
    ),
    const Spacer(),
  ],

```

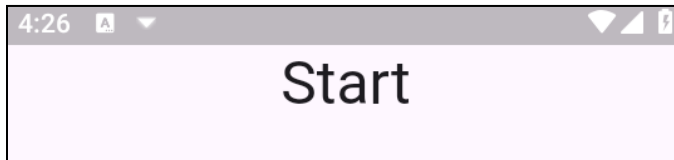


```
        ),  
      ),  
    ),  
    Expanded(  
      flex: 2,  
      child: SizedBox(  
        width: size.width,  
        child: Image.asset(  
          'assets/images/landscape.png',  
          fit: BoxFit.fill,  
        ),  
      ),  
    ),  
  ),  
  1,  
),  
,  
,  
);  
}  
}
```

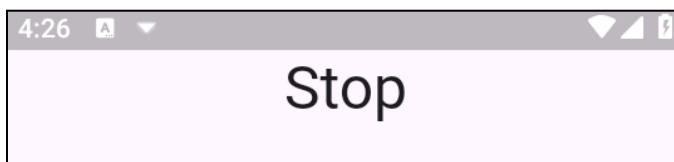
**Exercise 8** Using a Timer to run a process after a period.

We use a class `Timer(duration, callback)`.

This app waits for 3 seconds before calling a function.



After 3 seconds,



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

class TimerDemo extends StatefulWidget {
  const TimerDemo({super.key});

  @override
  State<TimerDemo> createState() => _TimerDemoState();
}

class _TimerDemoState extends State<TimerDemo> {
  String status = 'Start';

  void stop() {
    setState(() {
      status = 'Stop';
    });
  }

  @override
  void initState() {
```

```

    super.initState();
    // start timer
    Timer(const Duration(seconds: 3), stop);
    // Future.delayed(const Duration(seconds: 3), stop);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: Align(
          alignment: Alignment.topCenter,
          child: Text(
            status,
            style: Theme.of(context).textTheme.headlineLarge,
          ),
        ),
      ),
    );
  }
}

```

Observe that this example uses `initState()` as a class constructor which runs once at app starts.

Note that you can also use `Future.delayed(Duration(seconds: 3), stop);` to get the same result. However, Future class cannot be stopped before timeout while Timer class can.

To stop the Timer class immediately, use `timer.cancel()`.

```

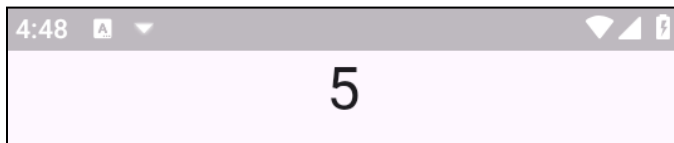
Timer timer = Timer(const Duration(seconds: 3), stop);
timer.cancel();

```

**Exercise 9** Update the timer every defined interval. This example demonstrates a countdown timer at every second.

We use a class `Timer.periodic(duration, (timer) {})` to repeat a process at an interval and `timer.cancel()` to stop the timer.

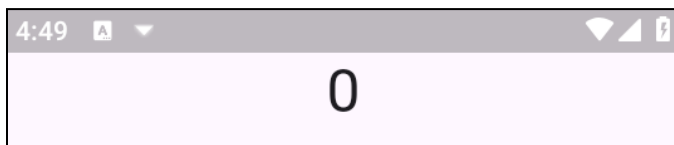
First the number starts at 5.



The number decreases every second.



The timer stops when the number reaches 0.



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

class Countdown extends StatefulWidget {
  const Countdown({super.key});

  @override
  State<CountDown> createState() => _CountDownState();
}

class _CountDownState extends State<CountDown> {
  int count = 5;
```

```

void countdown(Timer timer) {
  setState(() {
    count--;
    if (count == 0) {
      timer.cancel();
    }
  });
}

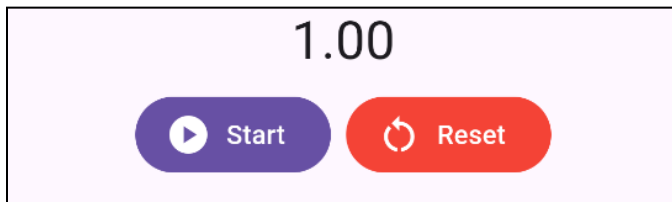
@override
void initState() {
  super.initState();
  Timer.periodic(const Duration(seconds: 1), (timer) => countdown(timer));
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SafeArea(
      child: Align(
        alignment: Alignment.topCenter,
        child: Text(
          count.toString(),
          style: Theme.of(context).textTheme.headlineLarge,
        ),
      ),
    ),
  );
}
}

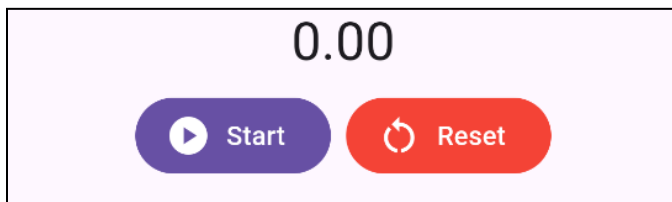
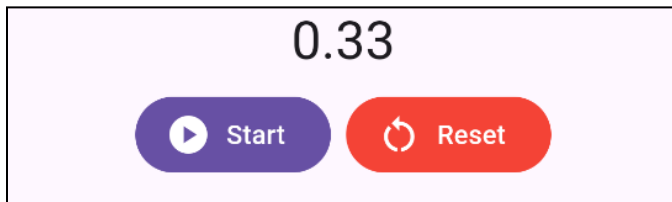
```

Note that do NOT forget to cancel the timer. Otherwise the timer will run forever!

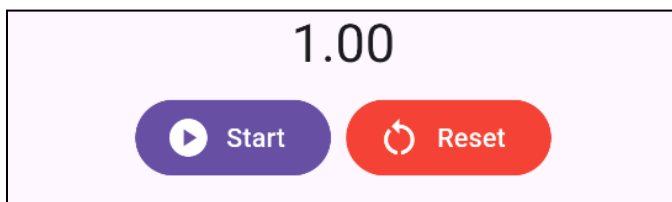
**Exercise 10** Create a countdown timer with a start button and a reset button. Also set a timer to decrease every 0.01 second from 1.00 to 0.00.



When clicking “Start”, the counter decreases by 0.01 second till 0.00.



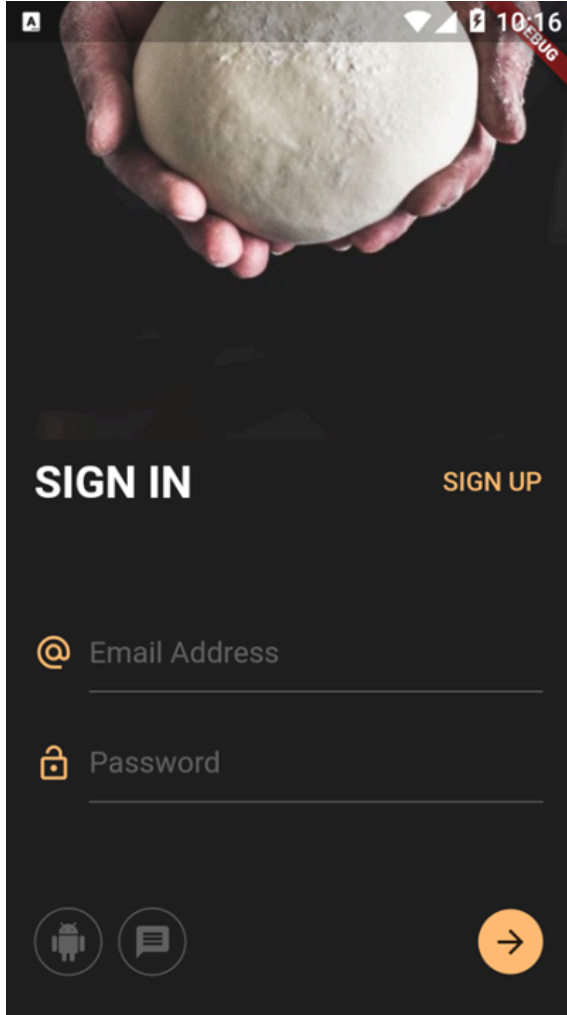
Clicking “Reset” will reset the counter to 1.00.



Note that you can also click “Reset” while the counter is running.

**Paste your code here.**

**Assignment 1** Create a page as shown below.



Paste your codes here.

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

const Color bgColor = Color(0xFF202020);
const Color primaryColor = Color(0xFFFFBD73);

class Asm1 extends StatelessWidget {
  const Asm1({super.key});

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

```
return Scaffold(  
  backgroundColor: bgColor,  
  body: Column(  
    crossAxisAlignment: CrossAxisAlignment.stretch,  
    children: [  
      Expanded(  
        flex: 4,  
        child: Image.asset(  
          "assets/images/baking.jpg",  
          fit: BoxFit.cover,  
          alignment: AlignmentGeometry.xy(0, 1.1),  
        ),  
      ),  
      Expanded(  
        flex: 4,  
        child: Padding(  
          padding: const EdgeInsets.only(left: 15, right: 15),  
          child: Column(  
            children: [  
              Row(  
                mainAxisAlignment: MainAxisAlignment.spaceBetween,  
                children: [  
                  Text(  
                    "sign in".toUpperCase(),  
                    style: TextStyle(  
                      color: Colors.white,  
                      fontWeight: FontWeight.bold,  
                      fontSize: 30,  
                    ),  
                  ),  
                  Text(  
                    "SIGN UP",  
                    style: TextStyle(  
                      color: primaryColor,  
                      fontWeight: FontWeight.bold,  
                    ),  
                  ),  
                ],  
              ),  
            ],  
          ),  
        ),  
      ),  
    ],  
  ),  
);
```



```
    ),  
    Spacer(),  
    Row(  
      children: [  
        Padding(  
          padding: const EdgeInsets.only(right: 10),  
          child: Icon(Icons.alternate_email, color:  
primaryColor),  
        ),  
        SizedBox(  
          width: 250,  
          child: TextField(  
            decoration: InputDecoration(  
              hintText: "Email Address",  
            ),  
          ),  
        ),  
      ],  
    ),  
    Row(  
      children: [  
        Padding(  
          padding: const EdgeInsets.only(right: 10),  
          child: Icon(  
            Icons.lock_open_outlined,  
            color: primaryColor,  
          ),  
        ),  
        SizedBox(  
          width: 250,  
          child: TextField(  
            decoration: InputDecoration(hintText: "Password"),  
          ),  
        ),  
      ],  
    ),  
    Spacer(),  
    Row(  

```

```

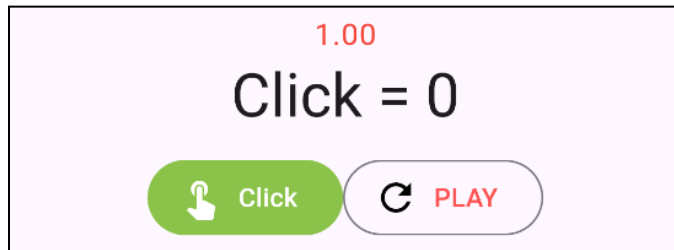
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          OverflowBar(
            children: [
              IconButton.outlined(
                onPressed: () {},
                icon: Icon(Icons.android),
              ),
              IconButton.outlined(
                onPressed: () {},
                icon: Icon(Icons.comment),
              ),
            ],
          ),
          OverflowBar(
            children: [
              IconButton.filled(
                onPressed: () {},
                icon: Icon(Icons.arrow_forward),
                color: bgColor,
                style: IconButton.styleFrom(
                  backgroundColor: primaryColor,
                ),
              ),
            ],
          ),
          SizedBox(height: 20),
        ],
      ),
    ),
  ],
),
);
}
}

```



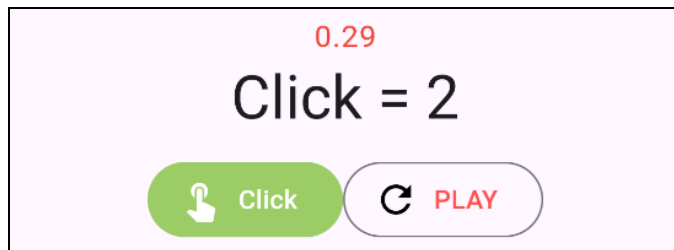
**Assignment 2** A “click as fast as you can” game.

When an app starts,

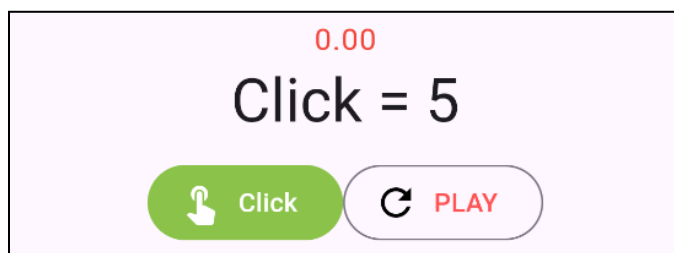


You click ‘PLAY’ to start a game. The time starts with 1.00 and auto decreases every 0.01 second.

While time is running, you can click a button “Click” to add a click.



When the time is up, the “CLICK” button does nothing.



If you click “PLAY” again, everything is reset and the countdown restarts.

Note that while time is running down, you can click ‘PLAY’ to restart the game.

**Paste your code here.**

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

```

class Asm2 extends StatefulWidget {
  const Asm2({super.key});

  @override
  State<Asm2> createState() => _Asm2State();
}

class _Asm2State extends State<Asm2> {
  double countdown = 1;
  int click = 0;
  Timer? timer;
  bool isButtonEnabled = true;

  void startCountdown() {
    timer = Timer.periodic(Duration(milliseconds: 100), (t) {
      setState(() {
        countdown -= 0.01;
        if (countdown <= 0) {
          countdown = 0;
          isButtonEnabled = false;
          t.cancel();
        }
      });
    });
  }

  void clickCount() {
    if (countdown == 1) {
      startCountdown();
    }
    setState(() {
      click++;
    });
  }

  void resetGame() {
    setState(() {
      countdown = 1;
    });
  }
}

```

```

        click = 0;
        isEnabled = true;
    });
    timer?.cancel();
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        body: SafeArea(
            child: Center(
                child: Column(
                    children: [
                        SizedBox(height: 20),
                        Text(countdown.toString(), style: TextStyle(color: Colors.red)),
                        Text("Click = $click", style: TextStyle(fontSize: 30)),
                        OverflowBar(
                            children: [
                                ElevatedButton.icon(
                                    onPressed: isEnabled ? clickCount : null,
                                    icon: Icon(Icons.ads_click),
                                    style: IconButton.styleFrom(
                                        backgroundColor: Colors.lightGreen,
                                        foregroundColor: Colors.white,
                                    ),
                                    label: Text("Click"),
                                ),
                                OutlinedButton.icon(
                                    onPressed: resetGame,
                                    icon: Icon(Icons.replay),
                                    style: IconButton.styleFrom(foregroundColor: Colors.black),
                                    label: Text("Click", style: TextStyle(color: Colors.red)),
                                ),
                            ],
                        ),
                    ],
                ),
            ),
        ),
    );
}

```

```
    ),  
    );  
}  
}
```