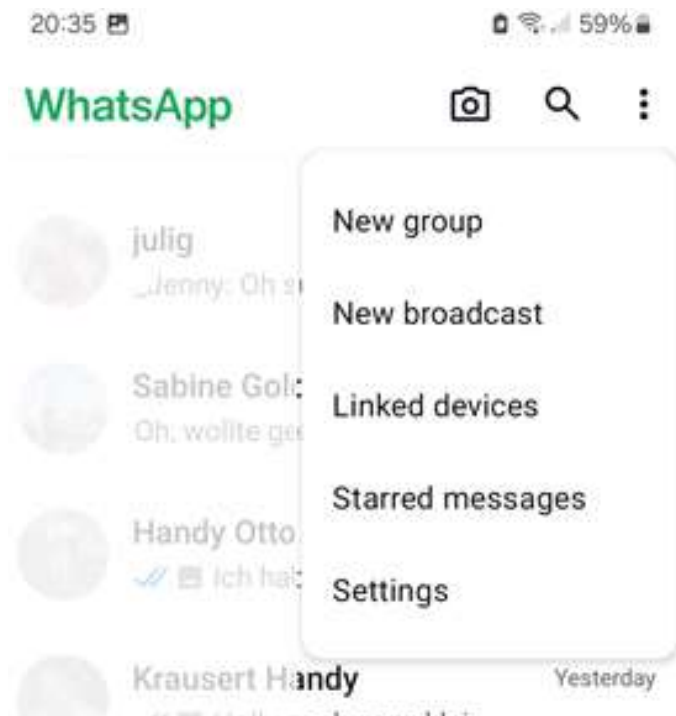
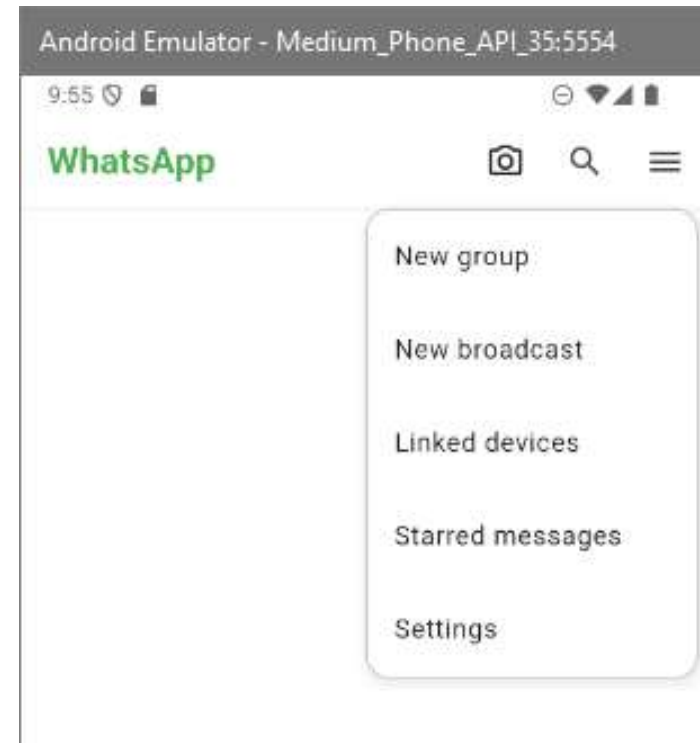


Goal 1: Create an app with an AppBar like WhatsApp



The original



Our goal

AppBar with action buttons



```
appBar: AppBar(  
  backgroundColor: Colors.white,  
  title: const Text(  
    "WhatsApp",  
    style: TextStyle(color: Colors.green, fontWeight: FontWeight.bold),  
  ), // Text  
  actions: [  
    IconButton(  
      onPressed: () {},  
      icon: const Icon(  
        Icons.camera_alt_outlined,  
      ), // Icon // IconButton  
    ),  
    IconButton(  
      onPressed: () {},  
      icon: const Icon(  
        Icons.search,  
      ), // Icon // IconButton  
    ),  
  ],  
), // AppBar
```

WhatsApp



Android Emulator - Medium_Phone_API_35:5554

4:27

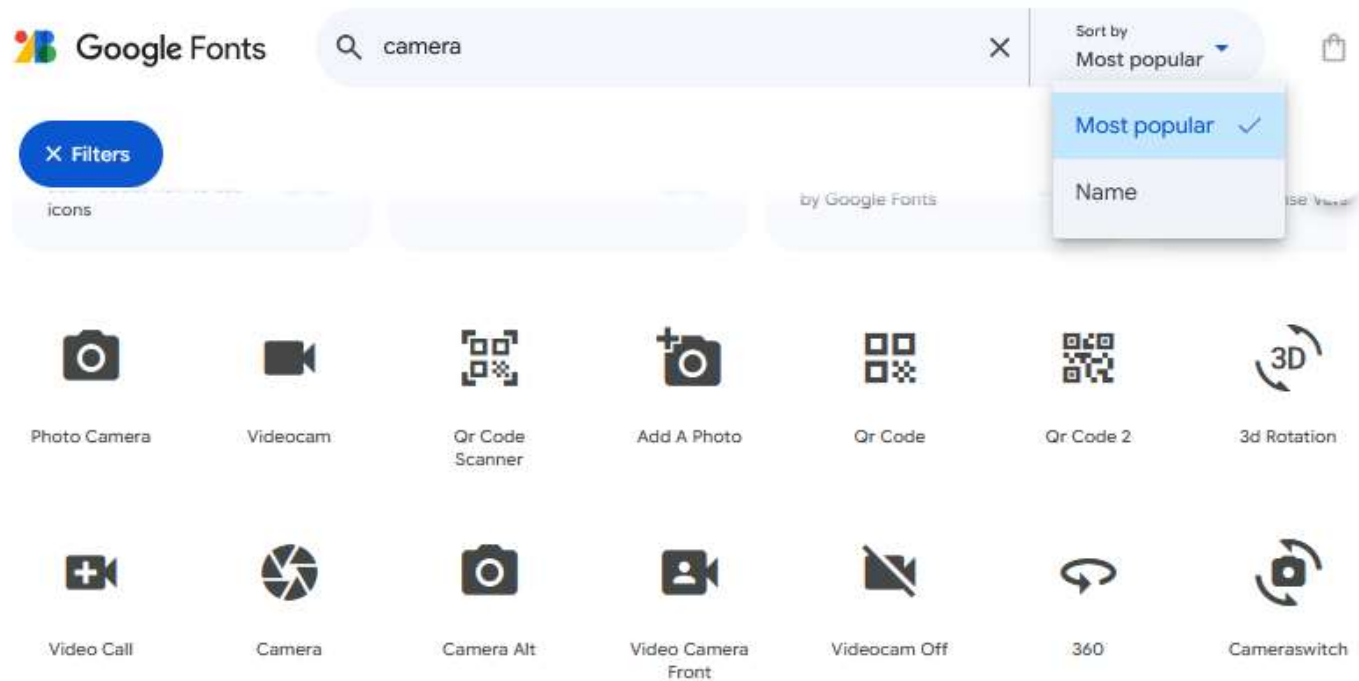
WhatsApp





Search for Flutter icons

See <https://fonts.google.com/icons>



Separator in AppBar

WhatsApp



ChatGPT ▾

↑ Share



Flutter create a separator under the AppBar

In Flutter, you can create a separator (divider) under the `AppBar` using the `Divider` widget or a `Container` with a bottom border. Here are two common ways to achieve this:

```
appBar: AppBar(  
  title: Text('My App'),  
  bottom: PreferredSize(  
    preferredSize: Size.fromHeight(1.0),  
    child: Divider(  
      height: 1,  
      thickness: 1,  
      color: Colors.grey, // Adjust color as needed  
    ),  
  ),  
)
```

AppBar with separator, still without “three dots”



```
appBar: AppBar(  
  backgroundColor: Colors.white,  
  title: const Text(  
    "WhatsApp",  
    style: TextStyle(color: Colors.green, fontWeight: FontWeight.bold),  
  ), // Text  
  bottom: PreferredSize(  
    preferredSize: const Size.fromHeight(1),  
    child: Divider(  
      height: 1,  
      color: Colors.grey.shade200,  
    ), // Divider // PreferredSize  
  actions: [  
    IconButton(  
      onPressed: () {},  
      icon: const Icon(  
        Icons.camera_alt_outlined,  
      ), // Icon // IconButton  
    IconButton(  
      onPressed: () {},  
      icon: const Icon(  
        Icons.search,  
      ), // Icon // IconButton  
    ],  
  ), // AppBar
```

WhatsApp






Hamburger menu (Drawer)

how to implement hamburger menu in flutter

In Flutter, you can implement a **hamburger menu (drawer menu)** using the `Drawer` widget inside a `Scaffold`. Here's a step-by-step guide:

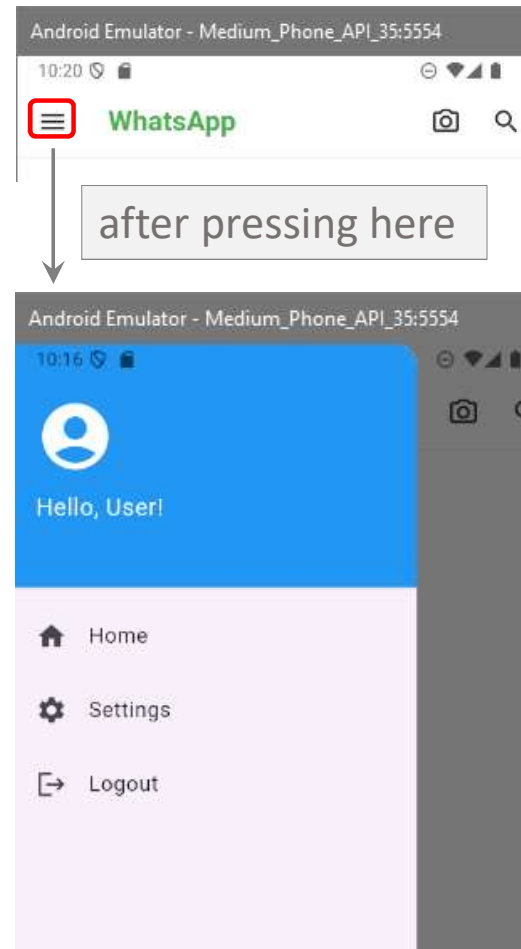
Steps to Implement a Hamburger Menu

1. Use `Scaffold` and define an `AppBar` with a menu button.
2. Add a `Drawer` to the `Scaffold`.
3. Populate the `Drawer` with menu items. 

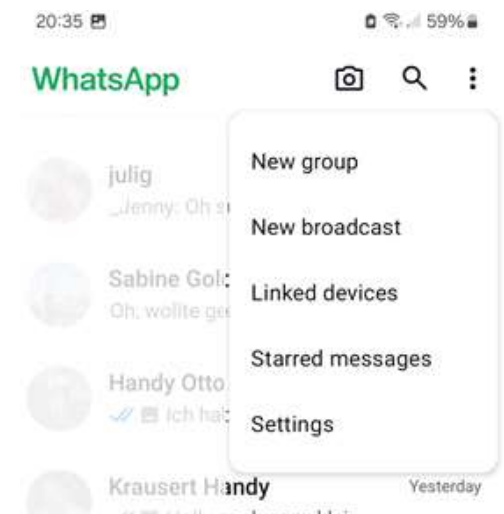


Hamburger menu (Drawer)

```
appBar: AppBar(  
  title: Text('Hamburger Menu'),  
)  
,  
drawer: Drawer( // 🍷 This is the hamburger menu  
  child: ListView(  
    padding: EdgeInsets.zero,  
    children: [  
      DrawerHeader(  
        decoration: BoxDecoration(color: Colors.blue),  
        child: Column(  
          crossAxisAlignment: CrossAxisAlignment.start,  
          children: [  
            Icon(Icons.account_circle, size: 60, color: Colors.white),  
            SizedBox(height: 10),  
            Text("Hello, User!",  
              style: TextStyle(color: Colors.white, fontSize: 18)),  
          ],  
        ),  
      ),  
    ],  
  ),  
  ListTile(  
    leading: Icon(Icons.home),  
    title: Text('Home'),  
    onTap: () {  
      Navigator.pop(context); // Close the drawer  
    },  
  ),  
),
```



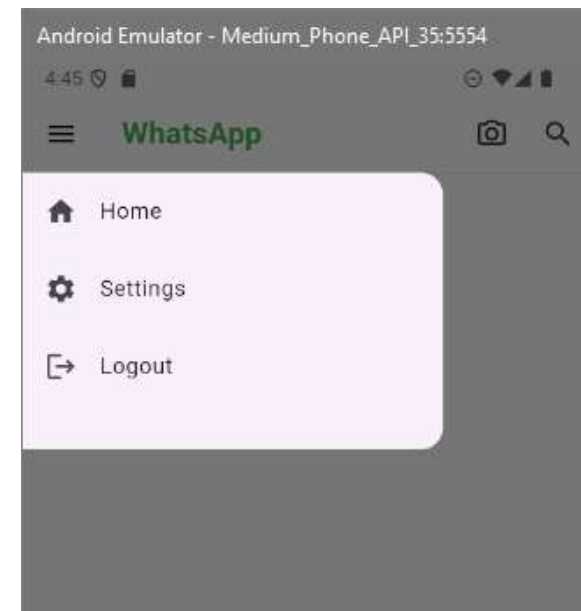
What we see in WhatsApp:





Drawer without Header, restricted in size and moved

```
drawer: Align(  
  alignment: Alignment.topLeft,  
  child: Container(  
    height: 200,  
    margin: const EdgeInsets.only(top: 80),  
    child: Drawer(  
      // ➡ This is the hamburger menu  
      child: ListView(  
        padding: EdgeInsets.zero,  
        children: [  
          ListTile(  
            leading: const Icon(Icons.home),  
            title: const Text('Home'),  
            onTap: () {  
              Navigator.pop(context); // Close the drawer  
            },  
          ),  
        ],  
      ),  
    ),  
  ),  
), // ListTile
```





Container with rounded border and shadow

ChatGPT ▾

🔗 Share

flutter container with rounded grey border and shadow

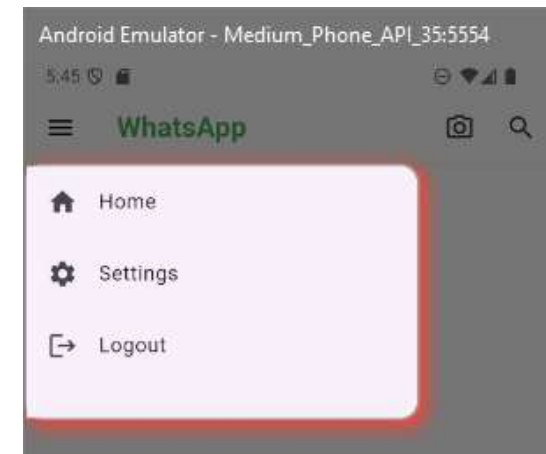
In Flutter, you can create a `Container` with a rounded grey border and shadow using `BoxDecoration`. Here's an example:

```
child: Container(  
  width: 200,  
  height: 100,  
  decoration: BoxDecoration(  
    color: Colors.white,  
    border: Border.all(color: Colors.grey, width: 2), // Grey border  
    borderRadius: BorderRadius.circular(15), // Rounded corners  
    boxShadow: [  
      BoxShadow(  
        color: Colors.black26, // Shadow color  
        blurRadius: 10, // Blur intensity  
        spreadRadius: 2, // Spread of the shadow  
        offset: Offset(4, 4), // Shadow position  
      ),  
    ],  
  ),  
),
```



Drawer does not respect the surrounding rectangle

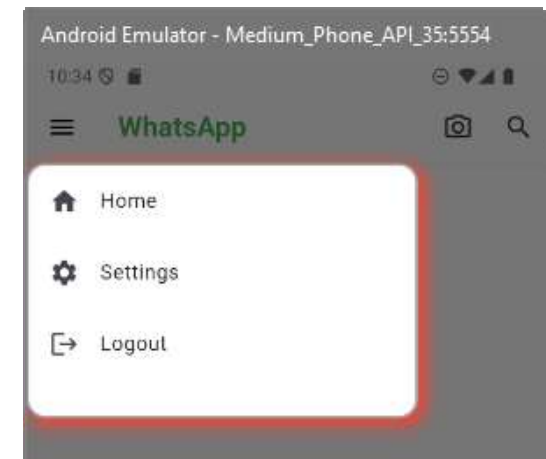
```
drawer: Align(  
  alignment: Alignment.topLeft,  
  child: Container(  
    height: 200,  
    margin: const EdgeInsets.only(top: 80),  
    decoration: BoxDecoration(  
      color: Colors.white,  
      border: Border.all(color: Colors.grey, width: 2), // Grey border  
      borderRadius: BorderRadius.circular(15), // Rounded corners  
      boxShadow: const [  
        BoxShadow(  
          color: Colors.red, // Shadow color  
          blurRadius: 10, // Blur intensity  
          spreadRadius: 2, // Spread of the shadow  
          offset: Offset(4, 4), // Shadow position  
        ), // BoxShadow  
      ],  
    ), // BoxDecoration  
    child: Drawer(  
      // 🍔 This is the hamburger menu  
      child: ListView(  
        padding: EdgeInsets.zero,  
        children: [  
          ListTile(  
            leading: const Icon(Icons.home),
```





Set Drawer background to transparent

```
drawer: Align(  
  alignment: Alignment.topLeft,  
  child: Container(  
    height: 200,  
    margin: const EdgeInsets.only(top: 80),  
    decoration: BoxDecoration(  
      color: Colors.white,  
      border: Border.all(color: Colors.grey, width: 2), // Grey border  
      borderRadius: BorderRadius.circular(15), // Rounded corners  
      boxShadow: const [  
        BoxShadow(  
          color: Colors.red, // Shadow color  
          blurRadius: 10, // Blur intensity  
          spreadRadius: 2, // Spread of the shadow  
          offset: Offset(4, 4), // Shadow position  
        ), // BoxShadow  
      ],  
    ), // BoxDecoration  
    child: Drawer(  
      backgroundColor: Colors.transparent,  
      // 🍷 This is the hamburger menu  
      child: ListView(  
        padding: EdgeInsets.zero,  
        children: [  
          ListTile(  
            leading: const Icon(Icons.home),
```





Alternative: give Drawer a shape

ChatGPT ▾

flutter what is the default shape of a Drawer

In Flutter, the default shape of a `Drawer` is a **rectangular** shape with square corners. It does not have any built-in rounded edges or custom shapes by default. However, you can customize its shape using the `shape` property of the `Drawer` widget by providing a `RoundedRectangleBorder` or any other `ShapeBorder`.

Example:

To give the `Drawer` rounded corners:

```
dart                                                                    Copy Edit

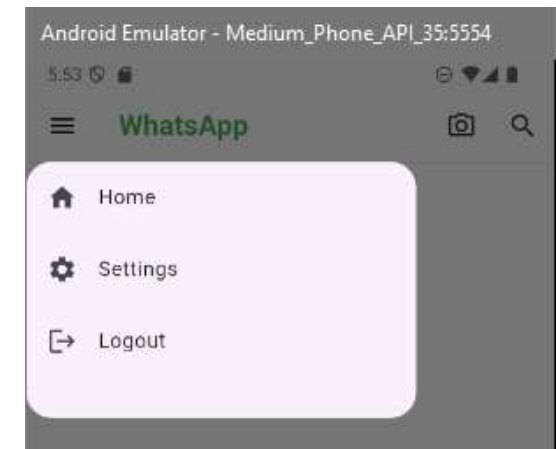
Drawer(
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.horizontal(right: Radius.circular(20)),
  ),
)
```



Alternative: give Drawer a shape

```
drawer: Align(  
  alignment: Alignment.topLeft,  
  child: Container(  
    height: 200,  
    margin: const EdgeInsets.only(top: 80),  
    // decoration: BoxDecoration(  
    //   color: Colors.white,  
    //   border: Border.all(color: Colors.grey, width: 2), // Grey border  
    //   borderRadius: BorderRadius.circular(15), // Rounded corners  
    //   boxShadow: const [  
    //     BoxShadow(  
    //       color: Colors.red, // Shadow color  
    //       blurRadius: 10, // Blur intensity  
    //       spreadRadius: 2, // Spread of the shadow  
    //       offset: Offset(4, 4), // Shadow position  
    //     ),  
    //   ],  
    // ),  
    child: Drawer(  
      shape: const RoundedRectangleBorder(  
        borderRadius: BorderRadius.all(Radius.circular(20)),  
      ), // RoundedRectangleBorder  
      // 🍔 This is the hamburger menu  
      child: ListView(  
        // ...  
      ),  
    ),  
  ),  
),
```

no longer needed here



As far as I know you cannot define a shadow in this solution.



Do not fade the app when Drawer is opened

ChatGPT ▾

🔗 Share

Flutter avoid that app is faded out when Drawer is opened

By default, when you open a `Drawer` in Flutter, the main screen gets a faded overlay to indicate that the drawer is active. If you want to prevent the main screen from fading when the drawer is opened, you can modify the `Drawer` behavior in the following ways:

Solution 1: Remove the Scrim Color

Set the `scrimColor` property of the `Drawer` to `Colors.transparent`:

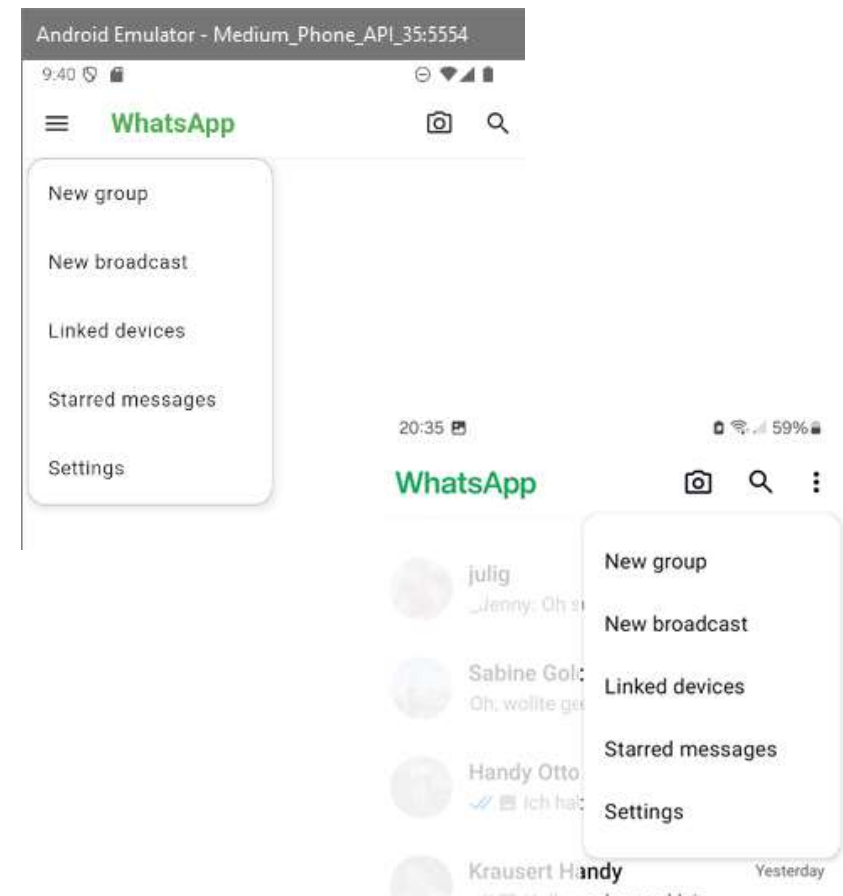
```
dart                                                                    Copy Edit
Scaffold(
  drawerScrimColor: Colors.transparent, // Prevents fading
  drawer: Drawer(
    child: ListView(
```

Adapt border and shadow and use entries like in WhatsApp



```
drawer: Align(  
  alignment: Alignment.topLeft,  
  child: Container(  
    height: 283,  
    margin: const EdgeInsets.only(top: 80, left: 4),  
    decoration: BoxDecoration(  
      color: Colors.white,  
      border: Border.all(color: Colors.grey.shade400, width: 1),  
      borderRadius: BorderRadius.circular(15), // Rounded corners  
      boxShadow: [  
        BoxShadow(  
          color: Colors.grey.shade200, // Shadow color  
          blurRadius: 4, // Blur intensity  
          spreadRadius: 2, // Spread of the shadow  
          offset: const Offset(1, 2), // Shadow position  
        ),  
      ],  
    ),  
    child: Drawer(  
      backgroundColor: Colors.transparent,  
      // This is the hamburger menu  
      child: ListView(  
        padding: EdgeInsets.zero,  
        children: [  
          ListTile(  
            title: const Text('New group'),  

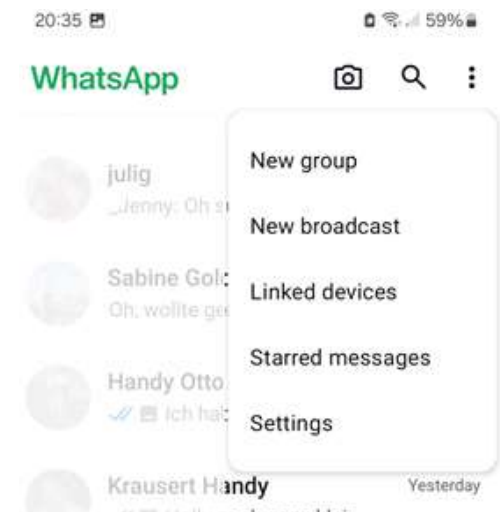
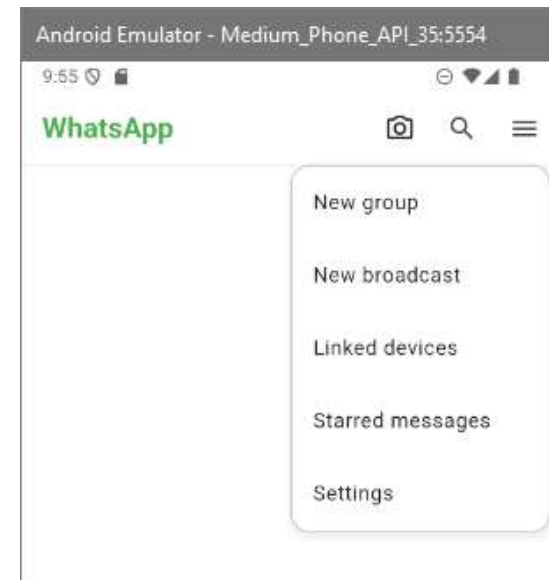
```



Drawer on the right

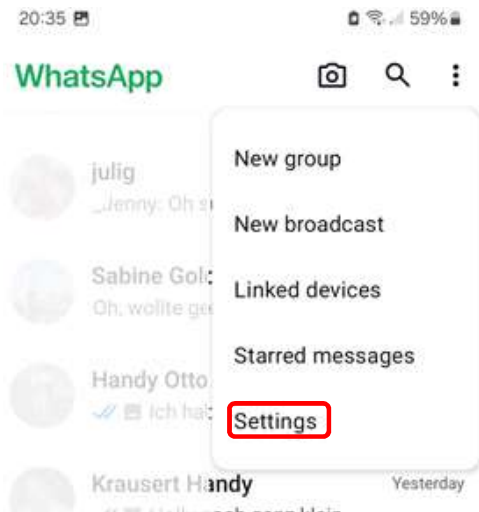


```
IconButton(  
  onPressed: () {},  
  icon: const Icon(  
    Icons.search,  
    color: Colors.black87,  
  )), // Icon // IconButton  
const EndDrawerButton()  
],  
), // AppBar  
endDrawer: Align(  
  alignment: Alignment.topRight  
  child: Container(  
    height: 283,  
    margin: const EdgeInsets.only(top: 80, right: 4),  
    decoration: BoxDecoration(  
      color: Colors.white,  
      border: Border.all(color: Colors.grey.shade400, width: 1),  
      borderRadius: BorderRadius.circular(15), // Rounded corners  
      boxShadow: [  
        BoxShadow(  
          color: Colors.grey.shade200, // Shadow color  
          blurRadius: 4, // Blur intensity  
          spreadRadius: 2, // Spread of the shadow  
          offset: const Offset(1, 2), // Shadow position  
        ), // BoxShadow  
      ],  
    ), // BoxDecoration  
    child: Drawer(  
      backgroundColor: Colors.transparent,
```

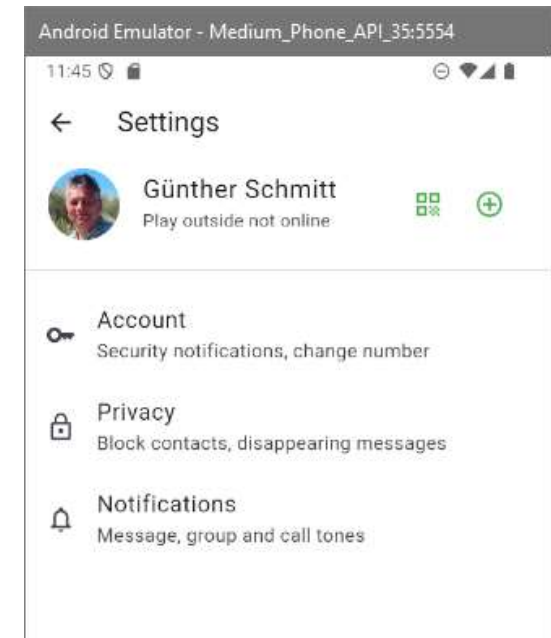
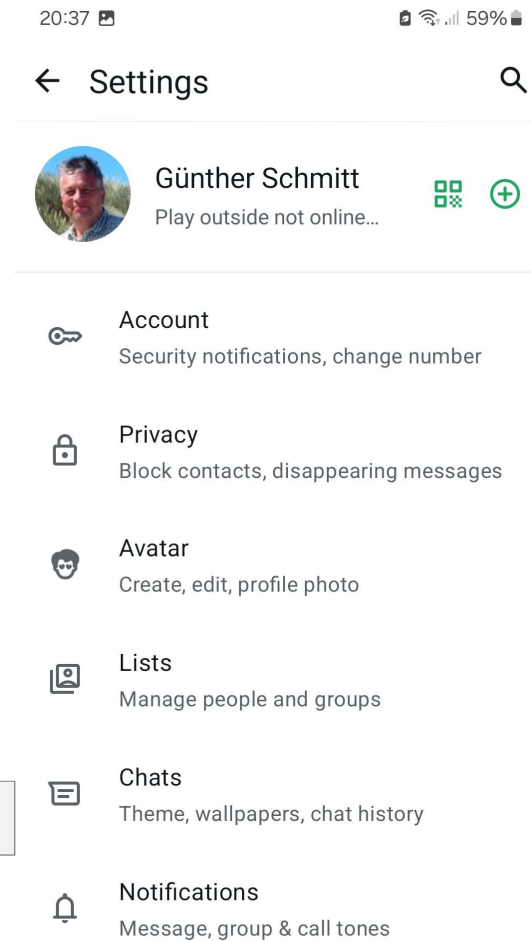




Goal 2: Open Settings Page from Drawer



The original one



Our simplification



Open Settings Page from Drawer

ChatGPT ▾

Share

flutter open a settings page from drawer

To open a settings page from the drawer in Flutter, follow these steps:

1. Create the Settings Page

First, create a separate Dart file for the settings page (`settings_page.dart`).

```
dart Copy Edit

import 'package:flutter/material.dart';

class SettingsPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Settings")),
      body: Center(child: Text("Settings Page")),
    );
  }
}
```

2. Modify the Main Page with a Drawer

Now, update your main page to include a drawer with a "Settings" navigation item.

```
ListTile(
  leading: Icon(Icons.settings),
  title: Text("Settings"),
  onTap: () {
    Navigator.pop(context); // Close the drawer
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => SettingsPage()),
    );
  },
),
```



Paste ChatGPT code and fix some warnings

```
ListTile(  
  //leading: const Icon(Icons.settings),  
  title: const Text('Settings'),  
  onTap: () {  
    // close the drawer  
    Navigator.of(context).pop();  
    Navigator.of(context).push(MaterialPageRoute(  
      builder: (context) => const SettingsPage()),); // MaterialPageRoute  
  },  
), // ListTile
```

```
class SettingsPage extends StatelessWidget {  
  const SettingsPage({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: const Text("Settings")),  
      body: const Center(child: Text("Settings Page")),  
    ); // Scaffold  
  }  
}
```

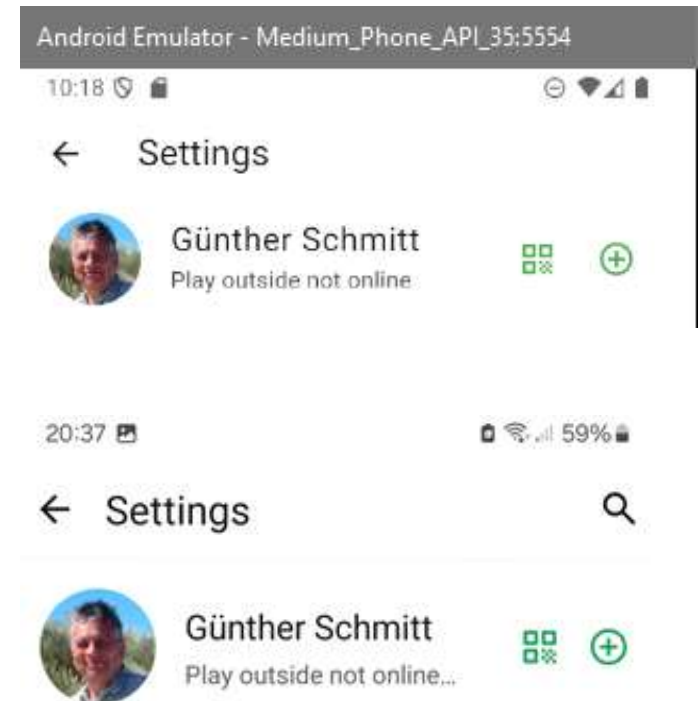
❗ Constructors for public widgets should have a named 'key' parameter. [dart\(use_key_in_widget_constructors\)](#)
Try adding a named parameter to the constructor.





Simulate upper part of WhatsApp's Settings page

```
ListTile(  
  leading: const CircleAvatar(  
    radius: 30,  
    foregroundImage: AssetImage(  
      "assets/images/gs.jpg",  
    )), // AssetImage // CircleAvatar  
  title: const Text("Günther Schmitt", style: TextStyle(fontSize: 20)),  
  subtitle: const Text("Play outside not online"),  
  trailing: Row(  
    mainAxisAlignment: MainAxisAlignment.min,  
    children: [  
      IconButton(  
        onPressed: () {},  
        icon: const Icon(Icons.qr_code, color: Colors.green)), // I  
      IconButton(  
        onPressed: () {},  
        icon: const Icon(Icons.add_circle_outline,  
          color: Colors.green)), // Icon // IconButton  
    ],  
  ), // Row // ListTile
```



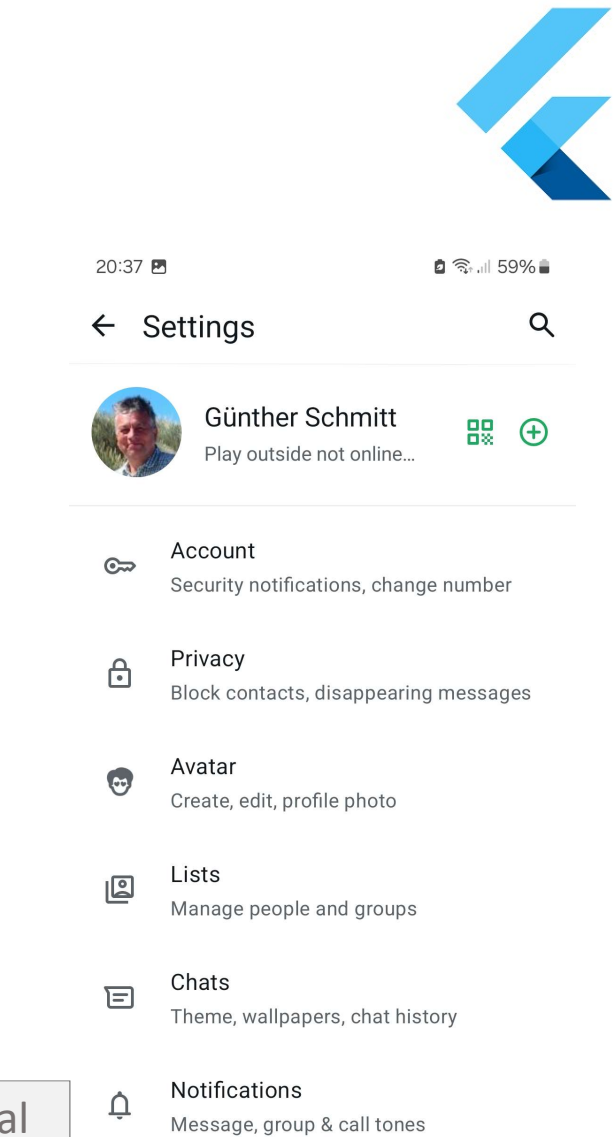
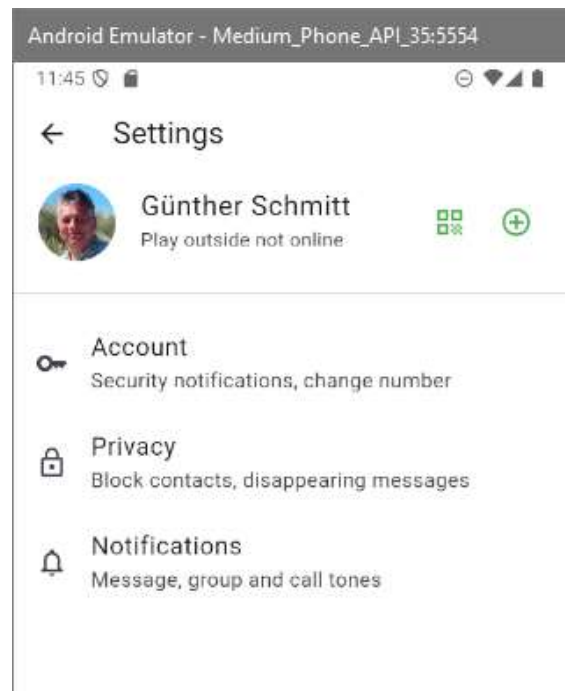
Exercise

Enhance the Settings Page so that it displays

- Account
- Privacy
- Notifications

like in the WhatsApp Settings.

Also show a small line above these items.

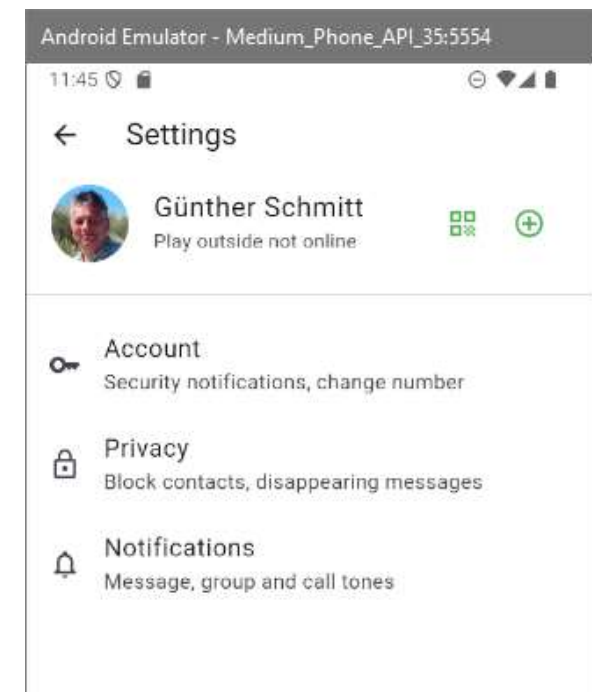


The original

Possible solution



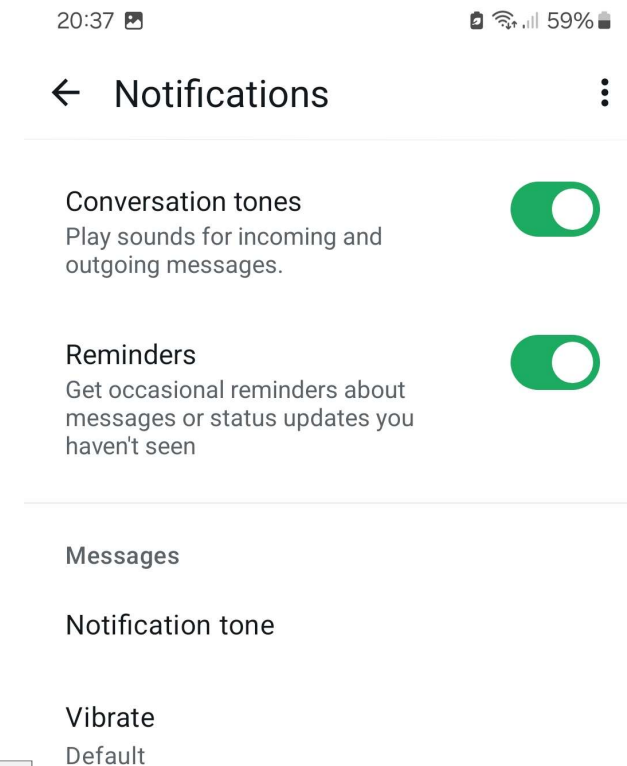
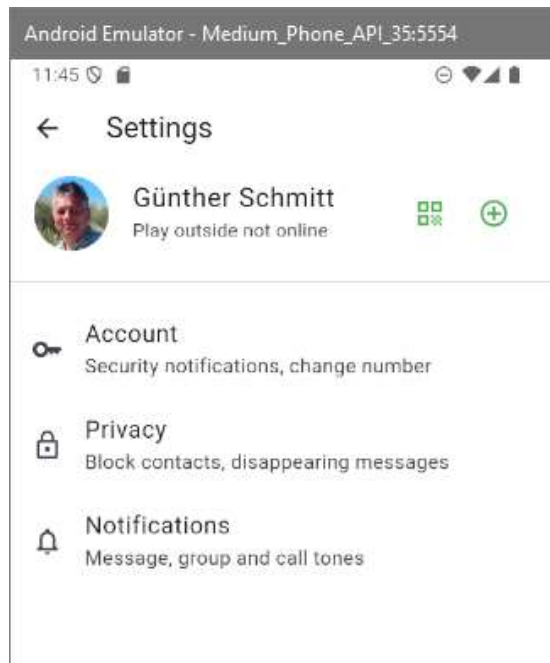
```
const Divider(height: 30, thickness: 1),
const ListTile(
  title: Text("Account", style: TextStyle(fontSize: 18)),
  subtitle: Text("Security notifications, change number",
    style: TextStyle(fontSize: 15)), // Text
  // for an overview of icons see https://fonts.google.com/icons
  leading: Icon(Icons.key_outlined),
), // ListTile
const ListTile(
  title: Text("Privacy", style: TextStyle(fontSize: 18)),
  subtitle: Text("Block contacts, disappearing messages",
    style: TextStyle(fontSize: 15)), // Text
  leading: Icon(Icons.lock_outline),
), // ListTile
const ListTile(
  title: Text("Notifications", style: TextStyle(fontSize: 18)),
  subtitle: Text("Message, group and call tones",
    style: TextStyle(fontSize: 15)), // Text
  leading: Icon(Icons.notifications_outlined),
) // ListTile
```





Exercise

Implement that pressing on „Notifications“ ListTile opens a new page:



The original

Possible solution:

Memorizing such options for next start (also called making such settings „persistent“) could be a topic for the end of our Flutter training.

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

  @override
  State<NotificationSettingsPage> createState() =>
    _NotificationSettingsPageState();
}

class _NotificationSettingsPageState extends State<NotificationSettingsPage> {
  bool switchValue = true;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Notifications"),
      ), // AppBar
      body: Column(
        children: [
          SwitchListTile(
            activeTrackColor: Colors.green.shade700,
            value: switchValue,
            onChanged: (value) {
              setState(() {
                switchValue = value;
              });
            },
            title: const Text("Conversation tones",
              style: TextStyle(fontSize: 20)), // Text
            subtitle: const Text(
              "Play sounds for incoming and outgoing messages.",
              style: TextStyle(fontSize: 15)), // Text
          ) // SwitchListTile
        ],
      )); // Column // Scaffold
  }
}
```