

7	endDrawer	Như drawer nhưng phía ngược lại
8	endDrawerEnableOpenGesture	Điều khiển việc drag của drawer
9	extendBody	Mặc định là false, nếu giá trị này là true thì phần BottomNavigationBar hoặc persistentFooterButton sẽ được thêm vào giao diện body (chiếm diện tích) và đưa vào phần bên dưới cùng của body
10	extendBodyBehindAppBar	Tương tự extendBody nhưng là appBar được đưa vào phần trên của body
11	floatingActionButtonLocation	Điều khiển vị trí floatingActionButton
12	persistentFooterButtons	Nút hiển thị ở cuối màn hình
13	resizeToAvoidBottomInsets	Thuộc tính true cho phép các thành phần float sẽ không ảnh hưởng nền bàn phím ảo khi hiển thị lên

2.6. MaterialApp Widget

Đây là một trong những Widget được đưa ra đầu tiên trong Flutter nó chứa các thành phần Material Widget khác như Text widget, DropdownButton widget, AppBar widget, Scaffold widget, ListView widget, StatelessWidget, StatefulWidget, IconButton widget, TextField widget, Padding widget, ThemeData widget,...

Phương thức khởi tạo:

```
MaterialApp({Key? key, GlobalKey<NavigatorState>? navigatorKey,
GlobalKey<ScaffoldMessengerState>? scaffoldMessengerKey,
Widget? home,
Map<String, WidgetBuilder> routes = const <String, WidgetBuilder>{},
String? initialRoute, RouteFactory? onGenerateRoute,
InitialRouteListFactory? onGenerateInitialRoutes,
RouteFactory? onUnknownRoute,
List<NavigatorObserver> navigatorObservers = const <NavigatorObserver>[],
TransitionBuilder? builder,
String title = '',
```

```

GenerateAppTitle? onGenerateTitle,
Color? color, ThemeData? theme, ThemeData? darkTheme,
ThemeData? highContrastTheme, ThemeData? highContrastDarkTheme,
ThemeMode? themeMode = ThemeMode.system,
Duration themeAnimationDuration = kThemeAnimationDuration,
Curve themeAnimationCurve = Curves.linear, Locale? locale,
Iterable<LocalizationsDelegate>? localizationsDelegates,
LocaleListResolutionCallback? localeListResolutionCallback,
LocaleResolutionCallback? localeResolutionCallback,
Iterable<Locale> supportedLocales = const <Locale>[Locale('en', 'US')],
bool debugShowMaterialGrid = false, bool showPerformanceOverlay = false,
bool checkerboardRasterCacheImages = false,
bool checkerboardOffscreenLayers = false,
bool showSemanticsDebugger = false,
bool debugShowCheckedModeBanner = true,
Map<ShortcutActivator, Intent>? shortcuts,
Map<Type, Action<Intent>>? actions, String? restorationScopeId,
ScrollBehavior? scrollBehavior, @Deprecated('Remove this parameter as it
is now ignored. ' 'MaterialApp never introduces its own MediaQuery; the
View widget takes care of that. ' 'This feature was deprecated after
v3.7.0-29.0.pre.') bool useInheritedMediaQuery = false))

```

Các thuộc tính:

STT	Thuộc tính	Ý nghĩa
1	action	Chứa danh sách các intent điều khiển các intent (phần thông tin chuyển đổi giữa các màn hình)
2	shortcuts	Xử lý phím tắt
3	backButtonDispatcher	Xử lý nhấn nút back
4	checkerboardRasterCacheImage	Bật/tắt cơ chế lưu tạm hình ảnh
5	color	Màu ứng dụng (nền icon ứng dụng)
6	theme	Giao diện ứng dụng
7	highContrastTheme	Giao diện ứng dụng khi ở chế độ tương phản cao

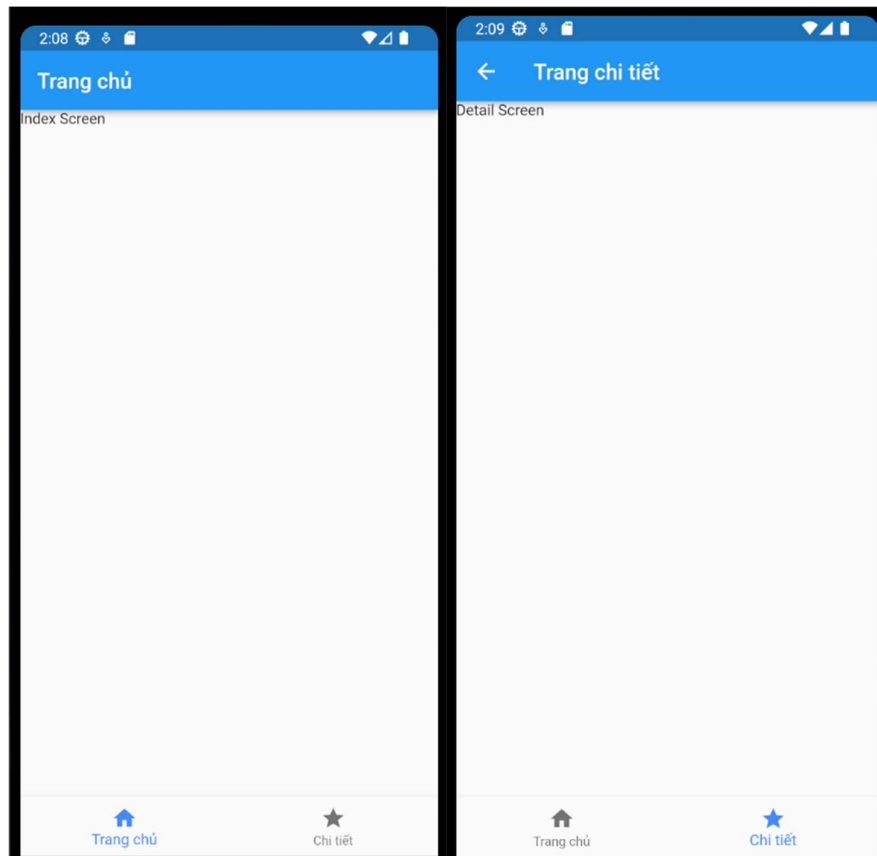
8	highContrastDarkTheme	Giao diện khi ứng dụng ở chế độ tương phản cao và tối
9	darkTheme	Giao diện khi ứng dụng ở chế độ tối
10	themeMode	Chế độ hiển thị ứng dụng (system, light, dark)
11	themeAnimationCurve	Hiệu ứng khi chuyển đổi theme
12	themeAnimationDuration	Thời gian chuyển đổi hiệu ứng
13	debugShowCheckedModeBanner	Bật/tắt hiển thị lỗi trên banner
14	debugShowMaterialGrid	Bật/tắt hiển thị lỗi trên giao diện lưới
15	showSemanticsDebugger	Hiển thị lỗi ngữ nghĩa
16	home	Màn hình ứng dụng mặc định hiển thị (route mặc định)
17	initialRoute	Route mặc định
18	locale	Ngôn ngữ
19	localizationsDelegate	Cơ chế ủy quyền (delegate) cho các ngôn ngữ (trong trường hợp App có nhiều ngôn ngữ)
20	navigatorObservers	Danh sách NavigatorState để quản lý điều hướng các màn hình
21	scrollBehavior	Xử lý cuộn
22	onGenerateInitialRoutes	Xử lý điều hướng route mặc định
23	onGenerateRoute	Xử lý điều hướng route
24	onGenerateTitle	Xử lý tạo ra title ứng dụng
25	onUnknownRoute	Xử lý điều hướng các route chưa khai báo

2.6.1. Ví dụ cơ bản

```
class MyApp extends StatelessWidget {
```

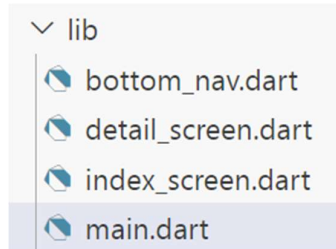
```
const MyApp({super.key});  
  
@override  
Widget build(BuildContext context) {  
  return MaterialApp(  
    title: 'Flutter Demo',  
    theme: ThemeData(  
      primarySwatch: Colors.blue,  
    ),  
    darkTheme: ThemeData(primarySwatch: Colors.grey),  
    color: Colors.amberAccent,  
    supportedLocales: {const Locale('en', ' ')},  
    debugShowCheckedModeBanner: false,  
    home: Scaffold(  
      appBar: AppBar(  
        title: const Text('Demo Material App'),  
      ),  
      body: const Center(child: Text('MaterialApp Demo')),  
    ));  
}
```

2.6.2. Ví dụ 2 – Điều hướng các màn hình



Xây dựng ứng dụng có 2 màn hình trang chính (index) và trang chi tiết (detail), chuyển đổi qua lại thông qua việc nhấn vào thanh menu dưới

Bước 1. Xây dựng cấu trúc tập tin như sau



Bước 2. Tạo màn hình index (index_screen.dart) (Phần menu dưới bottom_nav.dart sẽ xây dựng sau)

```
import 'package:flutter/material.dart';
import 'bottom_nav.dart';
class IndexScreen extends StatelessWidget {
  const IndexScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(appBar:AppBar(title:const Text('Trang chủ')),
      body:const Text('Index Screen'),
      bottomNavigationBar: const BottomNav(id:0),);
  }
}
```

Bước 3. Tạo màn hình detail (detail_screen.dart) (Phần menu dưới bottom_nav.dart sẽ xây dựng sau)

```
import 'package:flutter/material.dart';
import 'bottom_nav.dart';
class DetailScreen extends StatelessWidget {
  const DetailScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(appBar:AppBar(title:const Text('Trang chi tiết')),
      body:const Text('Detail Screen'),
      bottomNavigationBar: const BottomNav(id:1),);
  }
}
```

Bước 4. Tạo menu dưới (bottom_nav.dart)

```

import 'package:flutter/material.dart';
import 'detail_screen.dart';
import 'index_screen.dart';
class BottomNav extends StatelessWidget {
  const BottomNav({super.key, required this.idx});
  final idx;
  @override
  Widget build(BuildContext context) {
    return BottomNavigationBar(
      fixedColor: Colors.blueAccent,
      items: const [
        BottomNavigationBarItem(
          label: "Trang chủ",
          icon: Icon(Icons.home),
        ),
        BottomNavigationBarItem(
          label: "Chi tiết",
          icon: Icon(Icons.star,),
        ),
      ],
      currentIndex: idx,
      onTap: (int indexOfItem) {
        if(indexOfItem==0){
          Navigator.popUntil(context, (route) => route.isFirst);
          Navigator.push(context, MaterialPageRoute(builder: (context)
=>const IndexScreen()));
        }
        else
        {
          Navigator.popUntil(context, (route) => route.isFirst);
          Navigator.push(context, MaterialPageRoute(builder: (context)
=>const DetailScreen()));
        }
      });
  }
}

```

Bước 5. Xây dựng màn hình chính

```

import 'package:flutter/material.dart';
import 'index_screen.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,),
    darkTheme: ThemeData(primarySwatch: Colors.grey),
    color: Colors.amberAccent,
    supportedLocales: {const Locale('en', ' ')},
    debugShowCheckedModeBanner: false,
    home: const IndexScreen();
  }
}

```

2.6.3. Ví dụ 3 – Sử dụng thuộc tính routes

Xây dựng ứng dụng tương tự như ví dụ 2, sử dụng thuộc tính routes để khai báo các màn hình

Bước 1. Sao chép và chỉnh sửa main.dart

```

Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Demo',
    theme: ThemeData(
      primarySwatch: Colors.blue,
    ),
    darkTheme: ThemeData(primarySwatch: Colors.grey),
    color: Colors.amberAccent,
    supportedLocales: {const Locale('en', ' ')},
    debugShowCheckedModeBanner: false,
    initialRoute: '/',
    routes: {
      '/': (context) => const IndexScreen(),
      '/detail': (context) => const DetailScreen(),
    });
}

```

Bước 2. Sao chép và chỉnh sửa bottom_nav.dart

```

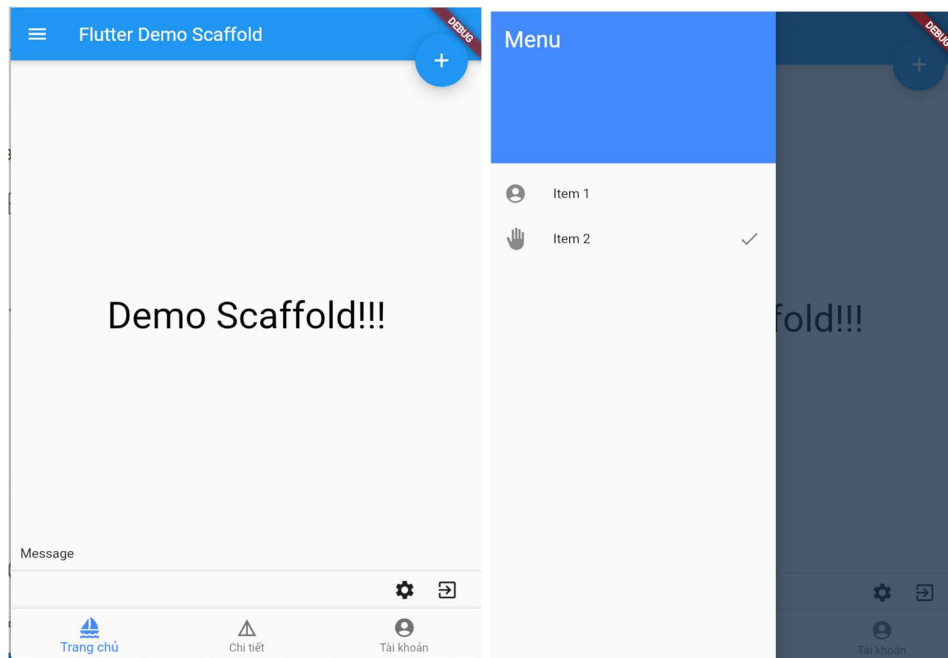
onTap: (int indexOfItem) {
  if(indexOfItem==0){
    Navigator.popUntil(context, (route) => route.isFirst);
    Navigator.pushNamed(context, '/');}
}

```

```
else
{
  Navigator.popUntil(context, (route) => route.isFirst);
  Navigator.pushNamed(context, '/detail');
}
});
```

2.7. Bài tập chương 2

Bài tập 1. Thiết kế giao diện Scaffold như sau



Bài tập 2. Xây dựng ứng dụng gồm 2 màn hình trang chủ (index), trang cá nhân (profile), chuyển đổi các trang sử dụng menu (drawer). Lưu ý các thành phần được tách thành các tập tin dart và các widget riêng biệt.

