

Aim: To apply navigation , routing and gestures in Flutter app

In a Flutter app, navigation, routing, and gestures play crucial roles in providing users with a seamless and interactive experience. Here's a brief overview of these concepts:

Navigation:

Navigation refers to the process of moving between different screens or "routes" within the app. Flutter provides the Navigator class, which manages a stack of routes and facilitates navigation between them.

You can push new routes onto the stack using `Navigator.push()` and remove routes using `Navigator.pop()`.

Named routes can be pre-defined in the app's route table, making it easier to navigate to specific screens by providing their names.

Nested navigation allows for hierarchical navigation structures, such as tab-based navigation or modal dialogs.

Routing:

Routing involves defining and managing the routes or paths that users can take through the app.

Routes are logical representations of screens or pages within the app and are associated with unique identifiers (route names or route keys).

Route management includes defining routes, specifying transitions between routes, passing data between routes, and handling route navigation events.

Gestures:

Gestures enable users to interact with the app's UI elements through touch-based interactions.

Flutter provides a wide range of gesture recognizers, such as `GestureDetector`, `InkWell`, `Draggable`, `LongPressGestureDetector`, etc., to detect and respond to user gestures.

Gesture recognizers can detect taps, swipes, drags, pinches, and other touch-based actions, allowing for rich and intuitive user interactions.

You can customize gesture behaviors, such as sensitivity, velocity, and directionality, to meet specific app requirements.

By implementing navigation, routing, and gestures effectively in your Flutter app, you can create a smooth and engaging user experience, enabling users to navigate between screens, interact with UI elements, and perform actions with ease.

CODE:**Navigation :**

```
import 'package:todark/app/modules/tasks/view/all_tasks.dart';
import 'package:todark/app/modules/settings/view/settings.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:iconsax/iconsax.dart';
import 'package:todark/app/modules/tasks/widgets/tasks_action.dart';
import 'package:todark/app/modules/todos/view/calendar_todos.dart';
import 'package:todark/app/modules/todos/view/all_todos.dart';
import 'package:todark/app/modules/todos/widgets/todos_action.dart';
import 'package:todark/theme/theme_controller.dart';
```

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

  @override
  State<HomePage> createState() => _HomePageState();
}
```

```
class _HomePageState extends State<HomePage> {
  final themeController = Get.put(ThemeController());
  int tabIndex = 0;
```

```
  final pages = const [
    AllTasks(),
    AllTodos(),
    CalendarTodos(),
    SettingsPage(),
  ];
```

```
  void changeTabIndex(int index) {
    setState(() {
      tabIndex = index;
    });
  }
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: IndexedStack(
      index: tabIndex,
```

```
        children: pages,
    ),
    bottomNavigationBar: NavigationBar(
      onDestinationSelected: (int index) => changeTabIndex(index),
      selectedIndex: tabIndex,
      destinations: [
        NavigationDestination(
          icon: const Icon(Icons.folder_2),
          selectedIcon: const Icon(Icons.folder_25),
          label: 'categories'.tr,
        ),
        NavigationDestination(
          icon: const Icon(Icons.task_square),
          selectedIcon: const Icon(Icons.task_square5),
          label: 'allTodos'.tr,
        ),
        NavigationDestination(
          icon: const Icon(Icons.calendar_1),
          selectedIcon: const Icon(Icons.calendar5),
          label: 'calendar'.tr,
        ),
        NavigationDestination(
          icon: const Icon(Icons.category),
          selectedIcon: const Icon(Icons.category5),
          label: 'settings'.tr,
        ),
      ],
    ),
    floatingActionButton: tabIndex == 3
      ? null
      : FloatingActionButton(
          onPressed: () {
            showModalBottomSheet(
              enableDrag: false,
              context: context,
              isScrollControlled: true,
              builder: (BuildContext context) {
                return tabIndex == 0
                  ? TasksAction(
                      text: 'create'.tr,
                      edit: false,
                    )
                  : TodosAction(
                      text: 'create'.tr,
```

```
                edit: false,
                category: true,
            );
        },
    );
},
child: const Icon(Icons.add),
),
);
}
}
```

Routing:

```
// Routing for "Appearance" setting card
SettingCard(
  icon: const Icon(Icons.brush_1),
  text: 'appearance'.tr,
  onPressed: () {
    showModalBottomSheet(
      context: context,
      builder: (BuildContext context) {
        // Bottom sheet modal dialog content
      },
    );
  },
),

// Routing for "Functions" setting card
SettingCard(
  icon: const Icon(Icons.code),
  text: 'functions'.tr,
  onPressed: () {
    showModalBottomSheet(
      context: context,
      builder: (BuildContext context) {
        // Bottom sheet modal dialog content
      },
    );
  },
),

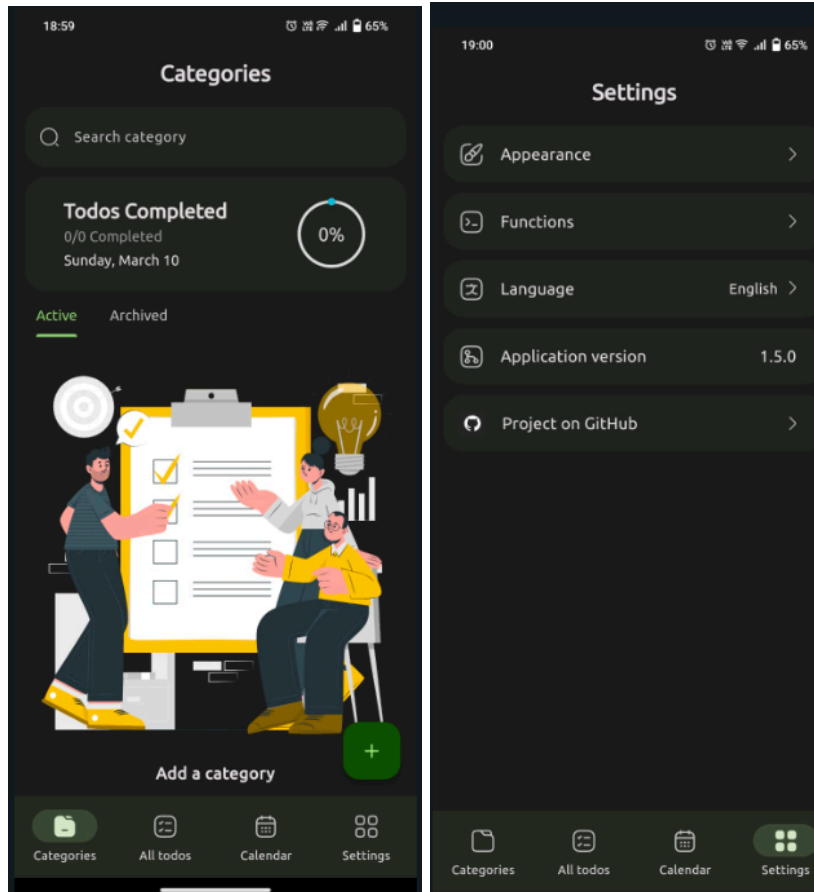
// Routing for "Language" setting card
SettingCard(
```

```
icon: const Icon(Iconsax.language_square),
text: 'language'.tr,
onPressed: () {
  showModalBottomSheet(
    context: context,
    builder: (BuildContext context) {
      // Bottom sheet modal dialog content
    },
  );
},
),
```

Gestures:

// Gesture for opening GitHub page

```
SettingCard(
  icon: Image.asset(
    'assets/images/github.png',
    scale: 20,
  ),
  text: '${'project'.tr} GitHub',
  onPressed: () async {
    final Uri url = Uri.parse('https://github.com/sumeet05xcaliber');
    if (!await launchUrl(url, mode: LaunchMode.externalApplication)) {
      throw Exception('Could not launch $url');
    }
  },
),
```



Conclusion: From the above code we have successfully implemented gesture , routing and navigation for our flutter project.