

Experiment 3

Name: Ameya Angne

D15B/01

Aim: To include icons, images, fonts in Flutter app

Introduction:

In a Flutter app, incorporating icons, images, and fonts is essential for enhancing visual appeal and user engagement. Icons serve as intuitive visual cues, aiding in navigation and providing a polished interface. Flutter's extensive library of customizable icons allows developers to seamlessly integrate them into the app's design. Additionally, the inclusion of images not only adds vibrancy but also conveys information effectively. Flutter's Image widget facilitates the display of various image formats, enabling developers to showcase graphics, logos, or dynamic content effortlessly. Furthermore, the choice of fonts contributes significantly to the app's aesthetic. Flutter supports diverse font styles and allows developers to import custom fonts, empowering them to align the app's typography with the brand identity. By leveraging these Flutter features, developers can craft visually appealing and cohesive user interfaces, elevating the overall user experience.

Code:

```
import 'package:flutter/material.dart';
import 'package:zoom_clone_tutorial/resources/auth_methods.dart';
import 'package:zoom_clone_tutorial/screens/history_meeting_screen.dart';
import 'package:zoom_clone_tutorial/screens/meeting_screen.dart';
import 'package:zoom_clone_tutorial/utils/colors.dart';
import 'package:zoom_clone_tutorial/widgets/custom_button.dart';

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

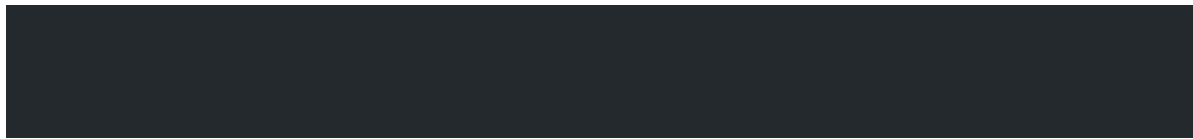
  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  int _page = 0;
  onPageChanged(int page) {
    setState(() {
      _page = page;
    });
  }
}
```

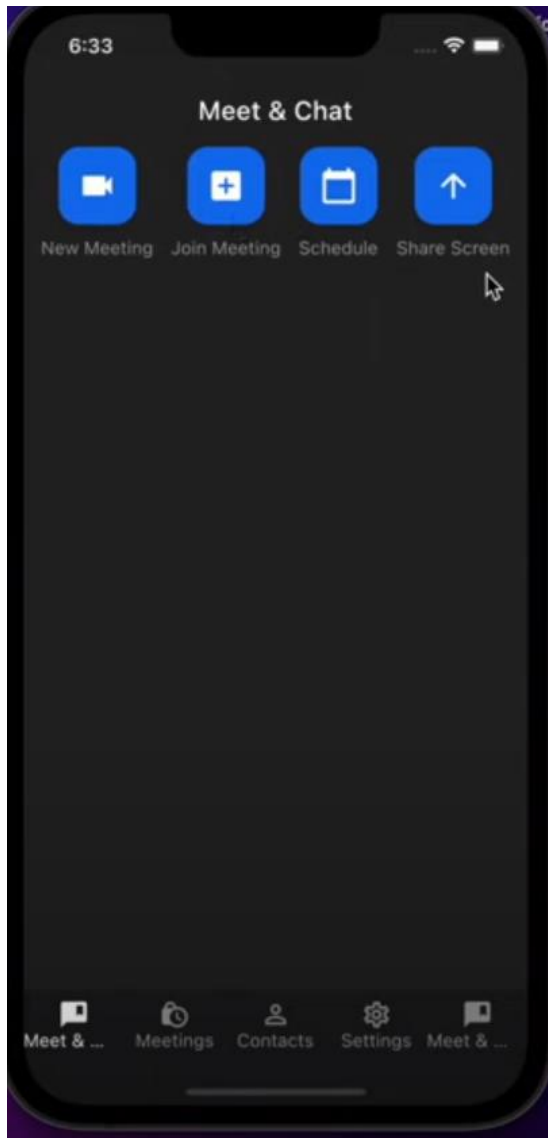
```
List<Widget> pages = [
  MeetingScreen(),
  const HistoryMeetingScreen(),
  const Text('Contacts'),
  CustomButton(text: 'Log Out', onPressed: () => AuthMethods().signOut()),
];
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: backgroundColor,
      elevation: 0,
      title: const Text('Meet & Chat'),
      centerTitle: true,
    ),
    body: pages[_page],
    bottomNavigationBar: BottomNavigationBar(
      backgroundColor: footerColor,
      selectedItemColor: Colors.white,
      unselectedItemColor: Colors.grey,
      onTap: onPageChanged,
      currentIndex: _page,
      type: BottomNavigationBarType.fixed,
      unselectedFontSize: 14,
      items: const [
        BottomNavigationBarItem(
          icon: Icon(
            Icons.comment_bank,
          ),
          label: 'Meet & Char',
        ),
        BottomNavigationBarItem(
          icon: Icon(
            Icons.lock_clock,
          ),
          label: 'Meetings',
        ),
        BottomNavigationBarItem(
          icon: Icon(
            Icons.person_outline,
          ),
          label: 'Contacts',
        ),
      ],
    ),
  );
```

```
        BottomNavigationBarItem(  
          icon: Icon(  
            Icons.settings_outlined,  
          ),  
          label: 'Settings',  
        ),  
      ],  
    ),  
  );  
}
```



Output:



Explanation:

1.Imports:

`import 'package:flutter/material.dart';` Imports the Flutter material library, which provides a set of widgets implementing Material Design.

`import 'package:zoom_clone_tutorial/resources/auth_methods.dart';` Imports functionality related to authentication methods from an external source, likely another file or package.

`import 'package:zoom_clone_tutorial/screens/history_meeting_screen.dart';` Imports a screen widget for displaying the history of meetings.

`import 'package:zoom_clone_tutorial/screens/meeting_screen.dart';` Imports a screen widget for hosting or joining meetings.

`import 'package:zoom_clone_tutorial/utlis/colors.dart';` Imports color constants from an external source.

`import 'package:zoom_clone_tutorial/widgets/custom_button.dart';` Imports a custom button widget from an external source, likely another file.

2.HomeScreen Class:

HomeScreen is a `StatefulWidget`, meaning it's a widget that maintains state.

It doesn't have any parameters in its constructor.

It overrides the createState() method to return an instance of _HomeScreenState.

3. _HomeScreenState Class:

_HomeScreenState is the state class associated with HomeScreen.

It has an integer _page to keep track of the current page being displayed.

onPageChanged(int page) is a method to handle changes in the displayed page. It updates the _page state variable accordingly.

pages is a list of Widget objects representing the different pages of the HomeScreen. It includes MeetingScreen, HistoryMeetingScreen, a Text widget for displaying "Contacts", and a CustomButton widget for logging out. The CustomButton's onPressed callback invokes the signOut() method from the AuthMethods class.

4.build() Method:

This method builds the UI of the HomeScreen.

It returns a Scaffold widget as the root widget, providing structure for the screen.

The Scaffold includes an AppBar with a title "Meet & Chat".

The body of the Scaffold is set to the current page determined by `_page`.

The `bottomNavigationBar` provides navigation between different pages. It includes icons and labels for "Meet & Chat", "Meetings", "Contacts", and "Settings". Tapping on an item invokes the `onPageChanged` method to update the `_page` variable.

Conclusion:

Overall, this code represents the home screen of an application, featuring navigation between different sections such as meetings, meeting history, contacts, and settings. Additionally, it allows users to log out.