Name: Atharva Chavan Div: D15A Roll no: 10 Batch 'A'

MAD PWA Lab Exp 4

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

Theory:

In the realm of Flutter app development, the effective application of navigation, routing, and gestures is pivotal for creating an intuitive and engaging user experience. This theoretical exploration delves into the foundational concepts and principles guiding the implementation of navigation, routing, and gestures in Flutter applications.

Navigation and Routing

Navigation is the process of moving between different screens or sections within an app. Flutter employs a structured routing system to facilitate seamless transitions between these screens. This system is driven by the following principles:

Navigator.push()

The Navigator.push() method is employed to navigate from one screen to another by adding a new page to the navigation stack. This method is integral to the dynamic flow of a Flutter application.

Principles:

1. Screen Transition:

Invoking Navigator.push() triggers a transition to a new screen, pushing it onto the navigation stack.

This method is typically used when moving from one screen to another, such as transitioning from a home screen to a details screen.

2. Named Routes:

Navigator.pushNamed() is a variation of this method used when navigating to a screen defined by a named route.

Named routes provide a clear and maintainable way to organize and navigate between different screens in the app.

3. Arguments:

The method allows the passing of arguments to the destination screen, enabling dynamic content rendering.

Arguments facilitate the customization of the new screen based on the context of the navigation.

Navigator.pop()

Conversely, the Navigator.pop() method is employed to remove the current screen from the navigation stack, returning to the previous screen. It plays a vital role in controlling the flow of the application.

Principles:

1. Screen Removal:

When Navigator.pop() is called, the current screen is popped off the navigation stack, reverting to the previous screen.

This method is typically used in scenarios where a user completes a task on one screen and returns to the previous screen.

2. Data Passing:

Information can be passed back to the previous screen using the Navigator.pop() method.

This allows for a seamless exchange of data between screens, enhancing the overall user experience.

3. Return Values:

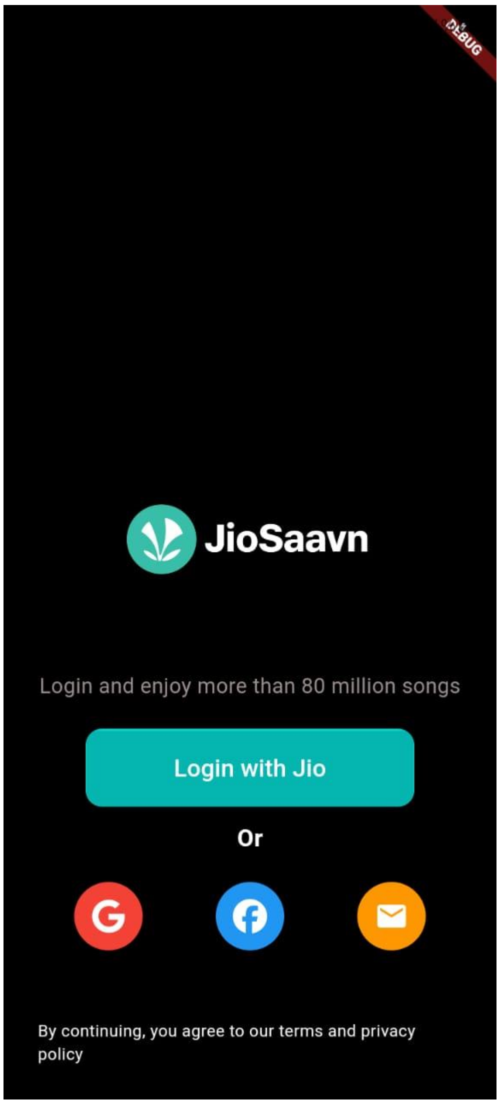
Developers can retrieve values returned from the popped screen, enabling dynamic updates or actions based on the user's interactions.

Code:

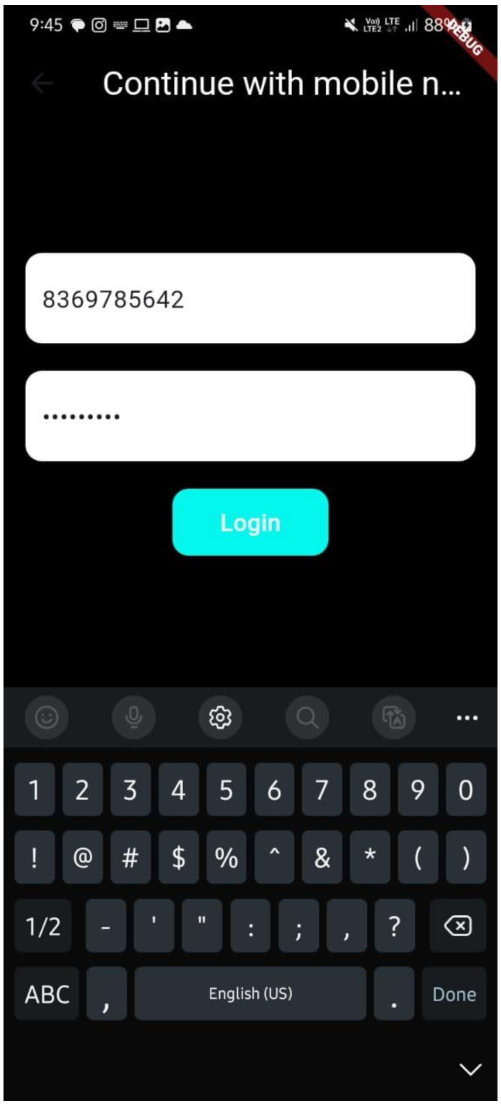
//mobilenumber.dart

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

class MobileNumberPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Continue with mobile number',
          style: TextStyle(color: Colors.white),
        ),
        backgroundColor: Colors.black,
      ),
      backgroundColor: Colors.black,
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              keyboardType: TextInputType.phone,
              decoration: InputDecoration(
                hintText: 'Enter mobile number',
                fillColor: Colors.white,
                filled: true,
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(12.0),
                  borderSide: BorderSide.none,
                ),
              ),
            ),
            SizedBox(height: 20),

            // Password TextField
            TextField(
              obscureText: true, // Hide the entered text for passwords
              decoration: InputDecoration(
                hintText: 'Enter password',
                fillColor: Colors.white,
                filled: true,
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(12.0),
                  borderSide: BorderSide.none,
                ),
              ),
            ),
            SizedBox(height: 20),

            ElevatedButton(
              onPressed: () {

              },
              style: ElevatedButton.styleFrom(
                primary: Color.fromRGBO(0, 255, 240, 1),
                shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(12.0),
                ),
              ),
              child: Padding(
                padding: const EdgeInsets.all(12.0),
                child: Text(
                  'Login',
                  style: TextStyle(fontSize: 16, color: Colors.white),
                ),
              ),
            ),
          ],
        ),
      ),
    );
```

Output:


DEBUG
JioSaavn
Login and enjoy more than 80 million songs
Login with Jio
Or
By continuing, you agree to our terms and privacy policy

Name: Atharva Chavan　　Div: D15A　　Roll no: 10　　Batch 'A'



Conclusion:

In conclusion, Navigator.push() and Navigator.pop() are foundational methods in Flutter navigation, facilitating the seamless movement between screens. Navigator.push() initiates the addition of a new screen to the stack, allowing for dynamic navigation, while Navigator.pop() removes the current screen, providing a mechanism for controlled navigation and data exchange between screens.