EXP5 : To apply navigation, routing and gestures in Flutter App

Navigation and Routing in Flutter enable moving between screens while maintaining state and history. Gestures add interactivity and enhance the user experience.

Navigation

- Uses MaterialApp with defined routes.
- Navigator.pushNamed() for named routes.
- Navigator.pop() to go back to the previous screen.

Routing Types

1. **Named Routes** (e.g., '/login', '/home').
2. **Dynamic Routes** (Passing arguments to a route).
3. **Nested Navigation** (Handling multiple navigators within the app).

Common Gestures

- **Tap / Double Tap** – Used for button clicks, quick interactions.
- **Long Press** – Triggers additional options.
- **Swipe / Drag** – Enables navigation, dismissal, or interactions.
- **Pan / Scale** – Used for zooming and panning in images or maps.

1. Setting Up Routes in MaterialApp

Define routes inside the MaterialApp widget:

```
MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => WelcomeScreen(),
    '/support': (context) => EmergencySupport(),
  },
);
```

2. Adding Gesture Detection

Use GestureDetector to handle gestures:

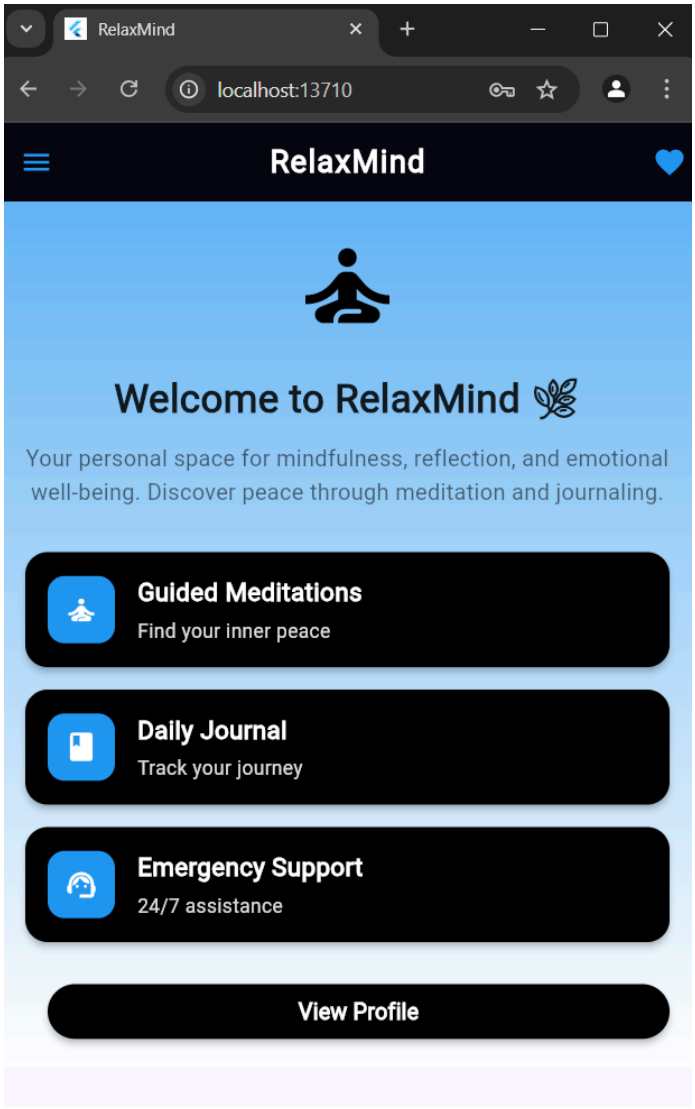
```
GestureDetector(
  onDoubleTap: () => handleDoubleTap(),
  onLongPress: () => handleLongPress(),
  child: theWidget(),
);
```

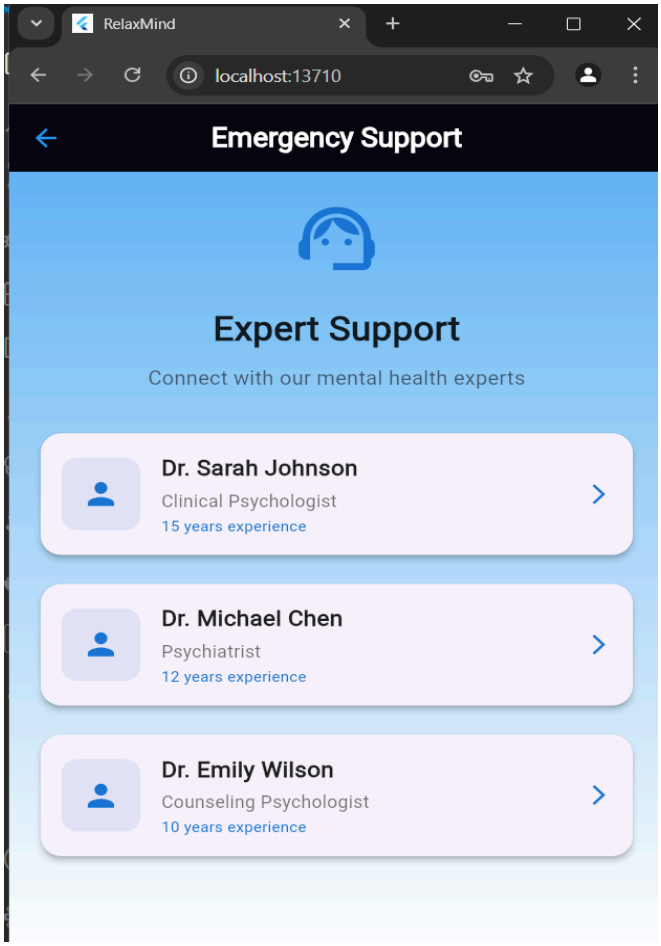
3. Navigating Between Screens

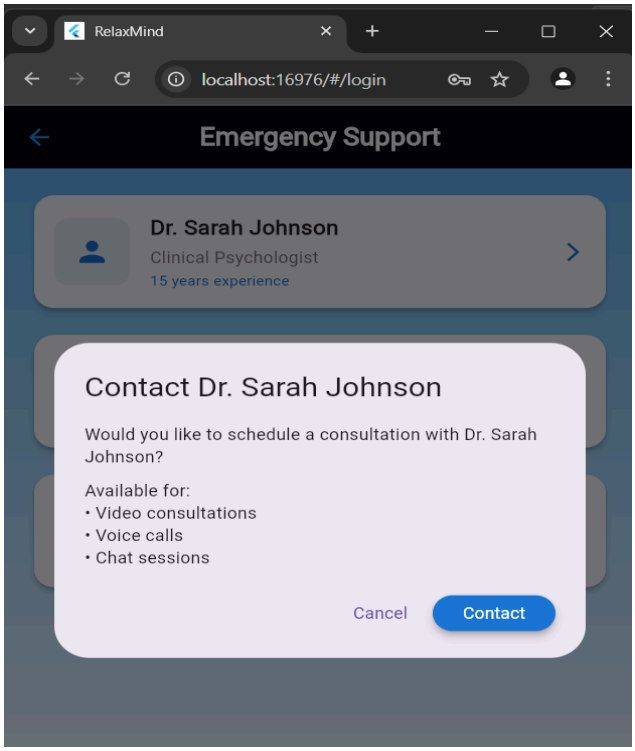
- Navigate to another screen using pushNamed:

  Navigator.pushNamed(context, '/support');

- Go back to the previous screen using pop: Navigator.pop(context);




RelaxMind
localhost:13710
Emergency Support
Expert Support
Connect with our mental health experts
Dr. Sarah Johnson
Clinical Psychologist
15 years experience
Dr. Michael Chen
Psychiatrist
12 years experience
Dr. Emily Wilson
Counseling Psychologist
10 years experience


RelaxMind
localhost:16976/#/login
Emergency Support
Dr. Sarah Johnson
Clinical Psychologist
15 years experience
Contact Dr. Sarah Johnson
Would you like to schedule a consultation with Dr. Sarah Johnson?
Available for:
• Video consultations
• Voice calls
• Chat sessions
Cancel
Contact

Conclusion: Implementing navigation, routing, and gestures in Flutter enhances user experience by enabling smooth transitions and interactive elements. However, challenges like gesture conflicts, incorrect route names, and state management issues can arise. Proper handling of named routes, dynamic navigation, and gesture detection ensures a seamless and user-friendly app.