PRACTICAL 5

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

1. Navigation in Flutter

Navigation in Flutter is the process of **switching between screens (routes)** within an app. There are two main ways to implement navigation:

1. **Using `Navigator.push()` and `Navigator.pop()`** (for simple navigation)
2. **Using Named Routes (`Navigator.pushNamed()`)** (for better route management)

Eg:
Navigator.pushNamed(context, '/add_medications');

2. Routing in Flutter

Routing defines how screens are structured and navigated. In our code, we used **named routes** to manage screen transitions.

Defining Routes in `main.dart`:

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Senior Care Assistant',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      initialRoute: '/login',
      routes: {
        '/': (context) => const HomeScreen(),
        '/login': (context) => const LoginScreen(),
        '/signup': (context) => const SignupScreen(),
        '/reset_password': (context) => const ResetPassword(),
        '/settings': (context) => const SettingsScreen(),
        '/search_users': (context) => const UserSearchScreen(),
        '/add_medications': (context) => const AddMedicationReminderScreen() ,
      },
    ); }}
```

3. Gesture Detection in Flutter

Flutter provides the `GestureDetector` widget to capture user gestures like **tap, swipe, and long press**.

Gestures Implemented in the Code:

```
GestureDetector(
  onTap: () {
    Navigator.pushNamed(context, '/add_medications');
  },
  onLongPress: () {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text("Long Press Detected!")),
    );
  },
  onHorizontalDragEnd: (details) {
    if (details.primaryVelocity! < 0) {
      Navigator.pushNamed(context, '/add_medications');
    }
  },
  child: Icon(Icons.medication, size: 80, color: Colors.blue.shade700),
);
```

CODE

Folder Structure

```
lib/
├── reuseable_widgets/
│   └── reuseable_widgets.dart
├── screens/
│   ├── features/
│   │   ├── add_medication_reminder_screen.dart
│   │   ├── caregiver_connection_screen.dart
│   │   ├── emergency_assistance_screen.dart
│   │   ├── health_records_screen.dart
│   │   └── medication_reminders_screen.dart
│   ├── settings/
│   │   ├── font_size_settings.dart
│   │   ├── language_settings.dart
│   │   └── settings_screen.dart
│   ├── home_screen.dart
│   ├── login_screen.dart
│   ├── reset_password.dart
│   └── signup_screen.dart
├── util/
│   └── color_util.dart
├── widgets/
│   └── custom_text_field.dart
├── firebase_options.dart
└── main.dart
```

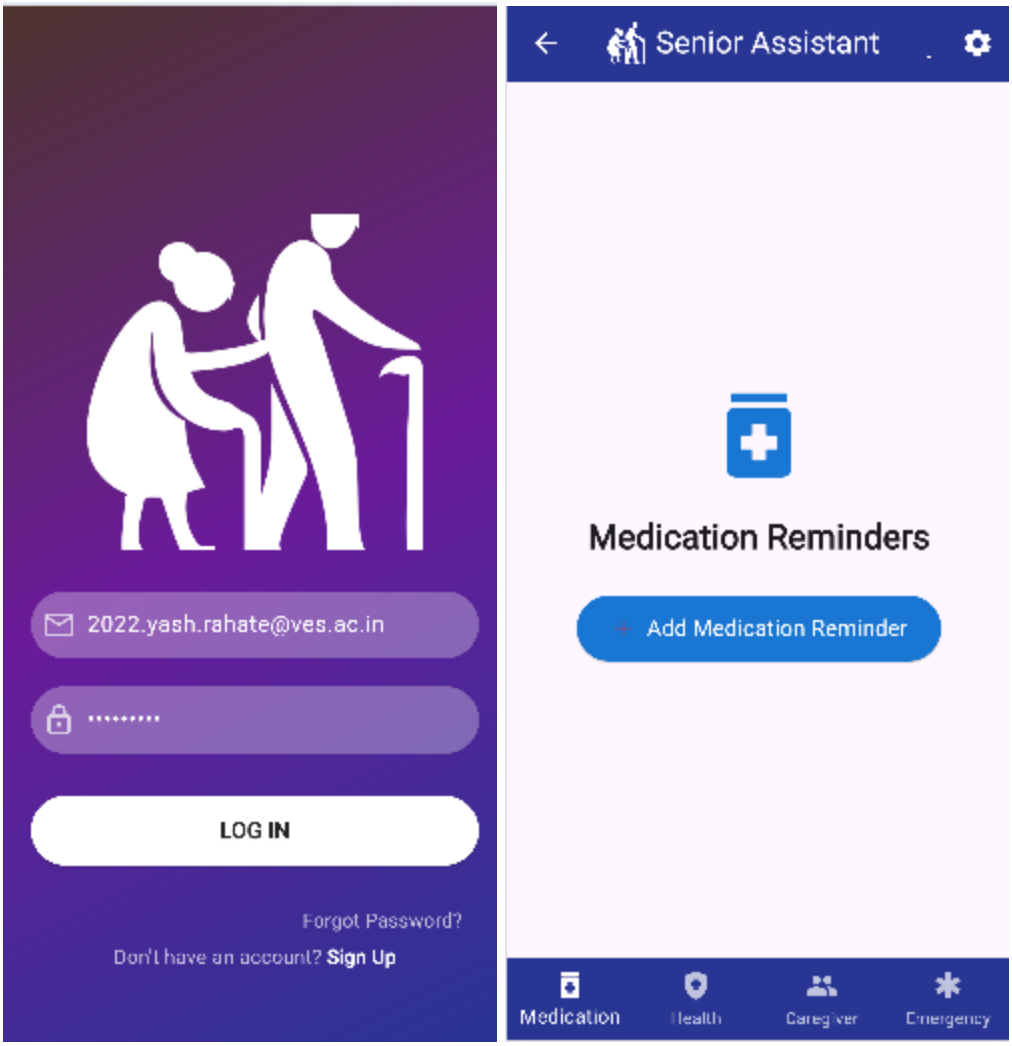
Navigation:

lib/screens/login_screen.dart:

```
signInSignUpButton(context, true, () {
                  if (_formKey.currentState!.validate()) {
                    _loginUser(context);
                  }
                }),

signUpOption(),
```

```
Future<void> _loginUser(BuildContext context) async {
   try {
     // Show a loading indicator
     showDialog(
       context: context,
       barrierDismissible: false,
       builder: (context) => Center(child: CircularProgressIndicator()),
     );

     // Attempt to log in
     UserCredential userCredential = await FirebaseAuth.instance
         .signInWithEmailAndPassword(
       email: _emailTextController.text.trim(),
       password: _passwordTextController.text.trim(),
     );

     print("Logged in successfully: ${userCredential.user!.email}");

     // Navigate to HomeScreen after successful login
     Navigator.pop(context); // Close the loading indicator
     Navigator.pushReplacement(
       context,
       MaterialPageRoute(builder: (context) => HomeScreen()),
     );
   } catch (error) {
     Navigator.pop(context); // Close the loading indicator
     _showErrorDialog(context, error.toString());
   }
 }
```

```
Row signUpOption() {
   return Row(
     mainAxisAlignment: MainAxisAlignment.center,
     children: [
       const Text("Don't have an account?",
           style: TextStyle(color: Colors.white70)),
       GestureDetector(
         onTap: () {
           Navigator.push(context,
               MaterialPageRoute(builder: (context) => SignupScreen()));
         },
         child: const Text(
           " Sign Up",
           style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold),
         ),
       )
     ],
   );
 }
```



Routes

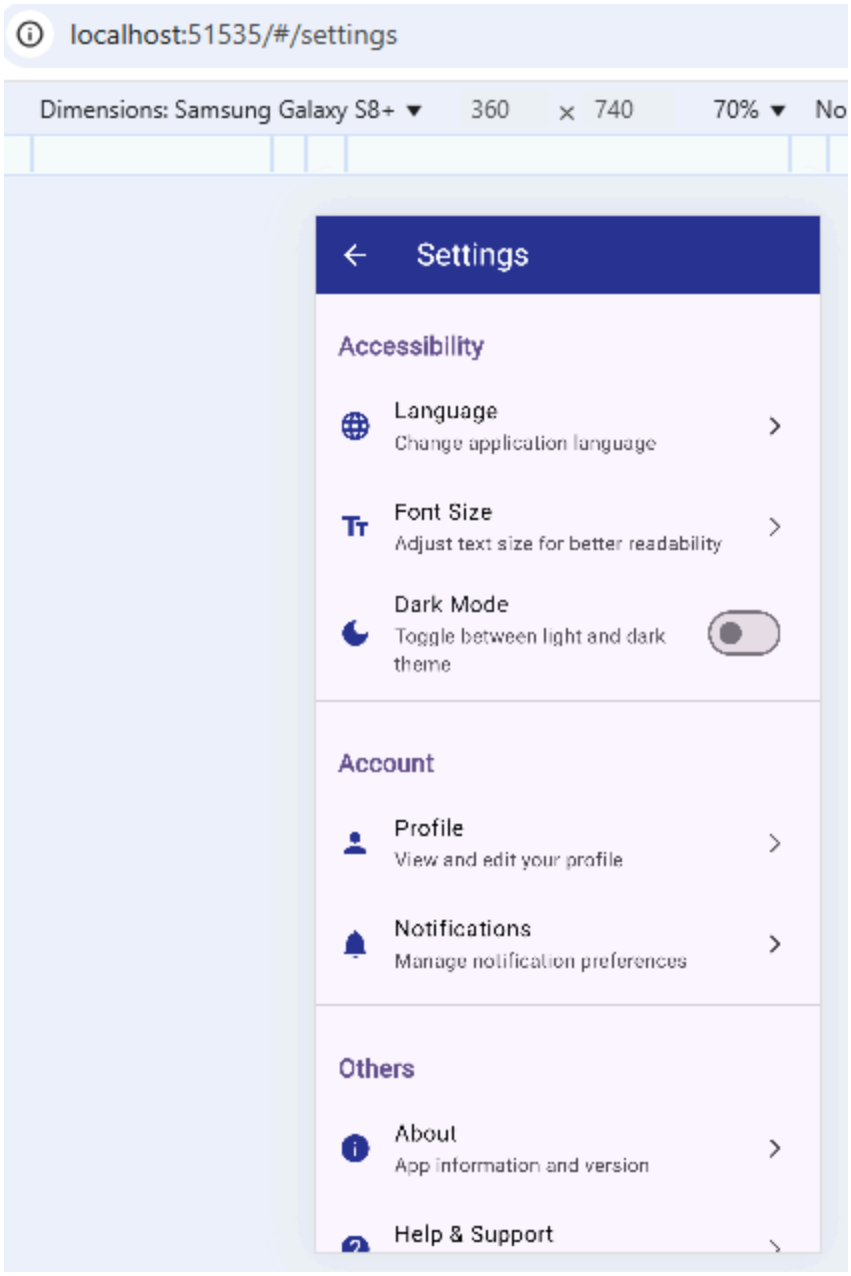
lib/main.dart:

```
import 'package:sca/screens/features/add_medication_reminder_screen.dart';
import 'package:sca/screens/login_screen.dart';
import 'package:sca/screens/home_screen.dart';
import 'package:sca/screens/signup_screen.dart';
import 'package:sca/screens/reset_password.dart';
import 'package:sca/screens/features/user_search_screen.dart';
import 'package:sca/screens/settings/settings_screen.dart';
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
```

```
      title: 'Senior Care Assistant',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      initialRoute: '/login',
      routes: {
        '/': (context) => const HomeScreen(),
        '/login': (context) => const LoginScreen(),
        '/signup': (context) => const SignupScreen(),
        '/reset_password': (context) => const ResetPassword(),
        '/settings': (context) => const SettingsScreen(),
        '/search_users': (context) => const UserSearchScreen(),
        '/add_medications': (context) => const AddMedicationReminderScreen() ,
      },
    );
  }
}
```

lib/screens/features/medication_reminders_screen.dart

```
class _MedicationRemindersScreenState extends State<MedicationRemindersScreen> {
  void _navigateToAddMedications() {
    Navigator.pushNamed(context, '/add_medications');
  }
```



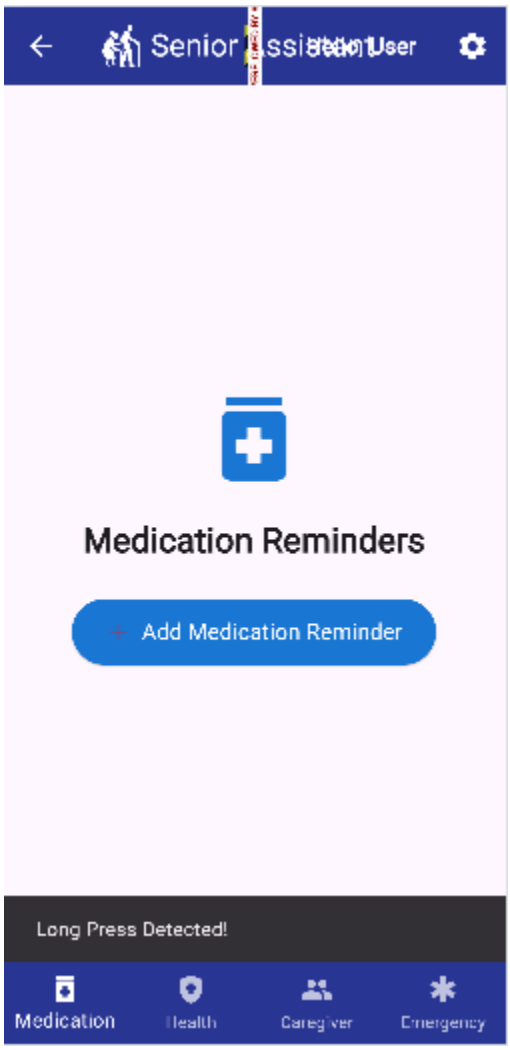
Gesture

lib/screens/features/medication_reminders_screen.dart:

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

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

  @override
  State<MedicationRemindersScreen> createState() => _MedicationRemindersScreenState();
}
```

```
class _MedicationRemindersScreenState extends State<MedicationRemindersScreen> {
  void _navigateToAddMedications() {
    Navigator.pushNamed(context, '/add_medications');
  }

  @override
  Widget build(BuildContext context) {
    return Center(
      child: GestureDetector(
        onTap: _navigateToAddMedications, // Tap to navigate
        onLongPress: () {
          ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text("Long Press Detected!")),
          );
        },
        onHorizontalDragEnd: (details) {
          if (details.primaryVelocity! < 0) {
            _navigateToAddMedications(); // Swipe Left to Navigate
          }
        },
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Icon(
              Icons.medication,
              size: 80,
              color: Colors.blue.shade700,
            ),
            const SizedBox(height: 16),
            const Text(
              'Medication Reminders',
              style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 24),
            ElevatedButton.icon(
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.blue.shade700,
                foregroundColor: Colors.white,
                padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 12),
                shape: RoundedRectangleBorder(
```

```
            borderRadius: BorderRadius.circular(30),
          ),
        ),
        onPressed: _navigateToAddMedications,
        icon: const Icon(Icons.add),
        label: const Text('Add Medication Reminder', style: TextStyle(fontSize: 16)),
      ),
    ],
  ),
 ),
); }}
```



Conclusion

In this experiment, we successfully applied **navigation, routing, and gestures** in a Flutter app. The following objectives were achieved:

Implemented Named Routing for better screen management.
Added Gesture-Based Navigation (tap, swipe, and long press).
Ensured a Smooth User Experience by making icons and buttons interactive.