MAD & PWA Lab Journal

Experiment No.	05
Experiment Title.	To apply navigation, routing and gestures in Flutter App
Roll No.	14
Name	Komal Milind Deolekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO2: Design and Develop interactive Flutter App by using widgets, layouts, gestures and animation
Grade:	

AIM : To apply navigation, routing and gestures in Futter App.

Theory :

Navigation, routing, and gestures are essential for creating a seamless and interactive user experience in Flutter applications. These features allow users to move between different screens, perform actions through touch interactions, and improve overall app usability.

1. Navigation and Routing in Flutter

What is Navigation?

Navigation refers to the ability of an app to transition between different screens (pages). In Flutter, navigation is managed using the Navigator class, which maintains a stack of routes. Each screen is considered a route, and users navigate between them using push and pop operations.

Types of Navigation in Flutter

1. **Imperative Navigation (Using Navigator Class)**
   - Uses functions like Navigator.push() and Navigator.pop() to move between screens.
   - Each new screen is added to the stack when pushed and removed when popped.
2. **Declarative Navigation (Using Named Routes)**
   - Screens (routes) are predefined and assigned names in the MaterialApp widget.
   - Users navigate using the assigned names instead of direct widget calls.

Working of Navigation in Flutter

1. **Pushing a New Screen:**
   - Navigator.push(context, MaterialPageRoute(builder: (context) => NewScreen()))
   - Adds a new screen on top of the navigation stack.
2. **Popping a Screen:**
   - Navigator.pop(context)
   - Removes the current screen and returns to the previous one.
3. **Using Named Routes:**

- Define routes in MaterialApp:

```
routes: {
  '/home': (context) => HomeScreen(),
  '/profile': (context) => ProfileScreen(),
}
```

- Navigate using:
-

```
Navigator.pushNamed(context, '/profile');
```

Importance of Navigation

- Allows users to transition between different screens.
- Provides a structured way to manage app flow.
- Enhances user experience by ensuring smooth screen transitions.

2. Gestures in Flutter

Gestures allow users to interact with an app through touch-based actions like tapping, swiping, pinching, and dragging. Flutter provides the GestureDetector widget to detect and respond to various gestures.

Common Gestures in Flutter

- **Tap Gesture:** Detects single or double taps on a widget.

```
GestureDetector(
  onTap: () {
    print("Tapped!");
  },
  child: Container(
    color: Colors.blue,
    height: 100,
    width: 100,
  ),
)
```

- **Long Press Gesture:** Detects when a user presses and holds a widget.

```
GestureDetector(
  onLongPress: () {
    print("Long Pressed!");
  },
)
```

- **Swipe (Drag) Gesture:** Detects horizontal or vertical swipes.
-

```
GestureDetector(
  onHorizontalDragUpdate: (details) {
    print("Swiped horizontally!");
  },
)
```

- **Double Tap Gesture:** Detects a quick double tap.

```
GestureDetector(
  onDoubleTap: () {
    print("Double Tapped!");
  },
)
```

Importance of Gestures in Flutter

- Enables user interaction with UI elements.
- Enhances accessibility and user experience.
- Makes apps more dynamic and responsive.

3. Combining Navigation, Routing, and Gestures

Navigation and gestures often work together to improve app usability. For example:

- A swipe gesture can navigate between screens.
- A long press gesture can open a context menu for navigation options.
- Tapping a button can trigger navigation using Navigator.push().

Example Use Case

1. The user taps a button → Navigates to a new screen.
2. The user swipes left → Moves to the previous screen.
3. The user long-presses an item → Opens a menu with navigation options.

Code :

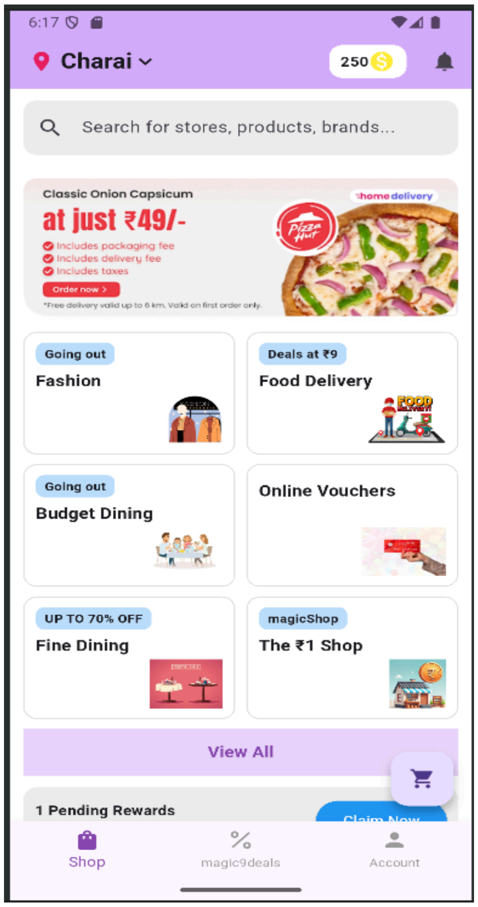
main_screen.dart

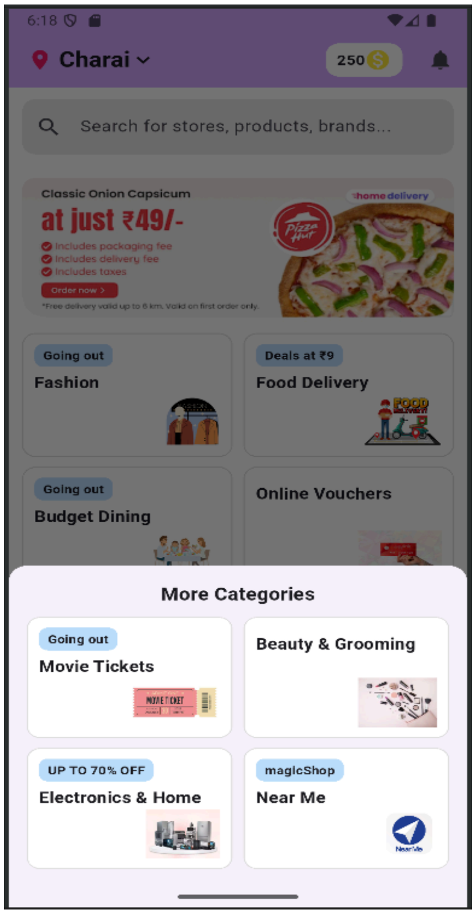
```
import 'package:flutter/material.dart';
import 'package:lucide_icons/lucide_icons.dart';
import 'home_page.dart';
import 'magic9_deals.dart';
import 'account_page.dart';

class MainScreen extends StatefulWidget {
  @override
  _MainScreenState createState() =>
_MainScreenState();
}

class _MainScreenState extends State<MainScreen>
{
  int _currentIndex = 0;

  // List of pages
  final List<Widget> _pages = [
    HomePage(),
    Magic9Deals(),
    AccountPage(),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: IndexedStack(
        index: _currentIndex,
        children: _pages,
      ),
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: _currentIndex,
        // selectedItemColor: Colors.blue,
        selectedItemColor: Color.fromRGBO(137, 74,
176,1),
        unselectedItemColor: Colors.grey,
        onTap: (index) {
          setState(() {
            _currentIndex = index;
          });
        },
        items: [
          BottomNavigationBarItem(icon:
Icon(Icons.shopping_bag), label: "Shop"),
          BottomNavigationBarItem(icon:
Icon(LucideIcons.percent), label: "magic9deals"),
          BottomNavigationBarItem(icon:
Icon(Icons.person), label: "Account"),
        ],
      ),
    );
  }
}
```

main.dart

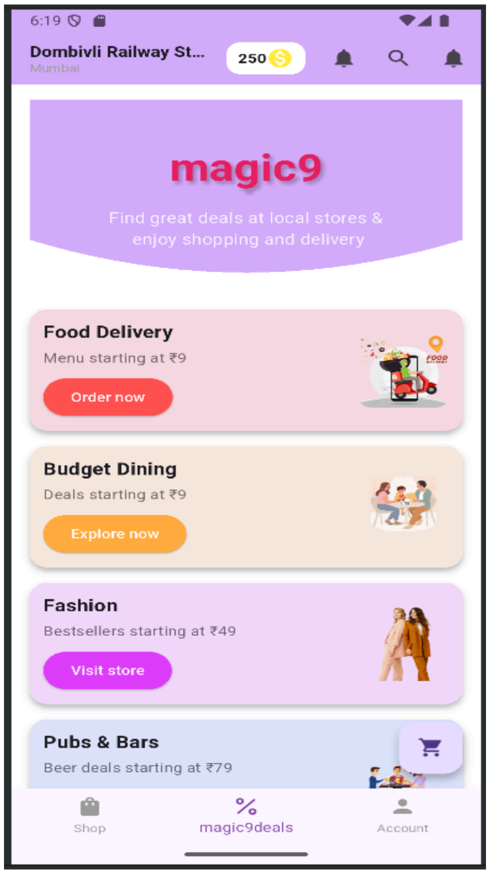
```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:provider/provider.dart';
import 'screens/phone_number_page.dart';
import 'screens/otp_verification_page.dart';
import 'screens/main_screen.dart';
import 'theme/theme.dart';
import 'screens/cart_provider.dart';
import 'screens/cart_page.dart';
import 'screens/sign_in_page.dart';
import 'screens/sign_up_page.dart';
import 'screens/user_provider.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (context) =>
CartProvider()),
        ChangeNotifierProvider(create: (context) =>
UserProvider()),
      ],
      child: MyApp(),
    ),
  );
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Magicpin Clone',
      theme: AppTheme.lightTheme,
      home: AuthWrapper(),  // Determines which page
to show
      routes: {
        '/signup' : (context) => SignUpPage(),
         '/login' : (context) => SignInPage(),
        '/phone' : (context) => PhoneNumberPage(),
        '/otp': (context) => OTPVerificationPage(),
        '/home': (context) => MainScreen(),
        '/cart': (context) => CartPage(),
        // '/login': (context) => LoginPage(),
      },
    );
  }
}

// ✅ AuthWrapper: Redirects user based on
authentication state
class AuthWrapper extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return StreamBuilder<User?>(
      stream:
FirebaseAuth.instance.authStateChanges(),
      builder: (context, snapshot) {
        if (snapshot.connectionState ==
ConnectionState.waiting) {
          return Scaffold(body: Center(child:
CircularProgressIndicator())); // Loading state
        } else if (snapshot.hasData) {
          return MainScreen(); // User is signed in
        } else {
          // return LoginPage(); // User is not signed in
          return SignInPage();
        }
      },
    );
  }
}
```

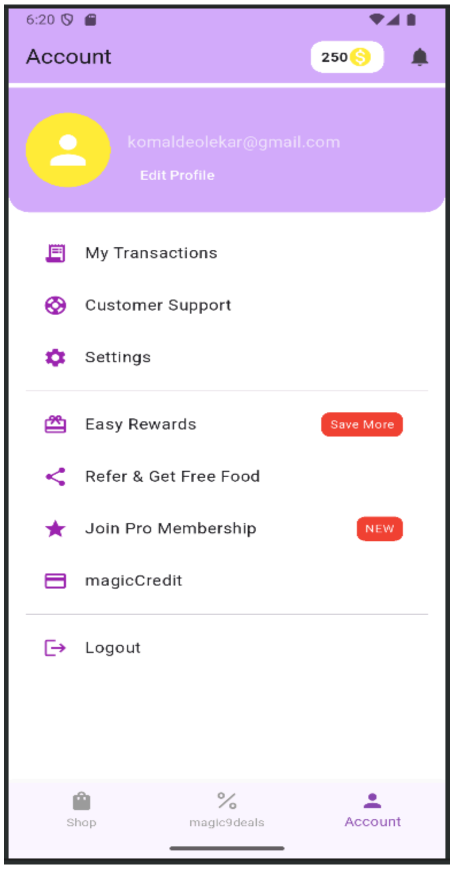
Project Title: MagicPin **Roll No.: 14**





Project Title: MagicPin **Roll No.: 14**


6:19
Dombivli Railway St...
Mumbai
250
magic9
Find great deals at local stores & enjoy shopping and delivery
Food Delivery
Menu starting at ₹9
Order now
Budget Dining
Deals starting at ₹9
Explore now
Fashion
Bestsellers starting at ₹49
Visit store
Pubs & Bars
Beer deals starting at ₹79
Shop
magic9deals
Account


6:20
Account
250
komaldeolekar@gmail.com
Edit Profile
My Transactions
Customer Support
Settings
Easy Rewards
Save More
Refer & Get Free Food
Join Pro Membership
NEW
magicCredit
Logout
Shop
magic9deals
Account