Aim:

To learn how to implement navigation, routing, and gestures in a Flutter app for better user interaction and app flow.

Theory:

In Flutter, navigation allows users to move between different screens (pages), and routing defines how the app handles these transitions. Flutter uses `Navigator` for navigation, while routing is managed through `MaterialPageRoute` or named routes. Gestures enable interactive actions like tapping, swiping, or dragging. Flutter provides gesture detection through widgets like `GestureDetector` to handle user inputs.

Code :

navbarwidget.dart

```
import 'package:flutter/material.dart';
import 'package:nutrilab/cartpage.dart';
import 'package:nutrilab/menupage.dart';
import 'package:nutrilab/profile.dart';
import 'package:nutrilab/savedpage.dart';

class BottomNav extends StatefulWidget {
final Map<String, int> cart;

const BottomNav({Key? key, required this.cart}) : super(key: key);

@override
State<BottomNav> createState() => _BottomNavState();
}

class _BottomNavState extends State<BottomNav> {
int myIndex = 0;

// Local cart state
Map<String, int> cart = {};

@override
void initState() {
super.initState();
// Initialize the local cart state with the passed cart
cart = Map.from(widget.cart);
```

```dart
}

// Callback to handle cart updates
void _handleCartUpdated(Map<String, int> updatedCart) {
setState(() {
cart = updatedCart; // Update the local cart state
});
print('Parent Cart Updated: $cart');
}

@override
Widget build(BuildContext context) {
List<Widget> pages = [
GoToMenuPage(
cart: cart, // Pass the cart map
onCartUpdated: _handleCartUpdated, // Pass the callback
),
GoToSavedPage(cart: cart), // Pass the cart map
GoToCartPage(cart: cart), // Pass the cart map
GoToProfile(),
];

return SafeArea(
child: Scaffold(
backgroundColor: Color.fromARGB(255, 225, 226, 209),
body: pages[myIndex],
bottomNavigationBar: BottomNavigationBar(
type: BottomNavigationBarType.fixed,
showUnselectedLabels: false,
selectedItemColor: Color.fromARGB(255, 253, 253, 253),
unselectedItemColor: Colors.white,
iconSize: 35,
backgroundColor: Color.fromARGB(255, 24, 79, 87),
onTap: (index) {
setState(() {
myIndex = index;
});
},
currentIndex: myIndex,
items: [
BottomNavigationBarItem(
icon: Icon(Icons.local_dining),
```

```dart
          label: 'Menu',
        ),
        BottomNavigationBarItem(
        icon: Icon(Icons.favorite_border_outlined),
        label: 'Saved',
        ),
        BottomNavigationBarItem(
        icon: Icon(Icons.shopping_cart_outlined),
        label: 'Cart',
        ),
        BottomNavigationBarItem(
        icon: Icon(Icons.person_sharp),
        label: 'Profile',
            ),

          ],

          ),

        ),

      );

    }

  }
```

Cartpage.dart
```dart
import 'package:flutter/material.dart';
import 'package:nutrilab/buildcart.dart';
class GoToCartPage extends StatefulWidget {
final Map<String, int> cart;
const GoToCartPage({Key? key, required this.cart}) : super(key: key);

@override
State<GoToCartPage> createState() => _GoToCartPageState();
}

class _GoToCartPageState extends State<GoToCartPage> {
@override
Widget build(BuildContext context) {
return Padding(padding: const EdgeInsets.all(25),
child: Column(
children: [
Align(
```
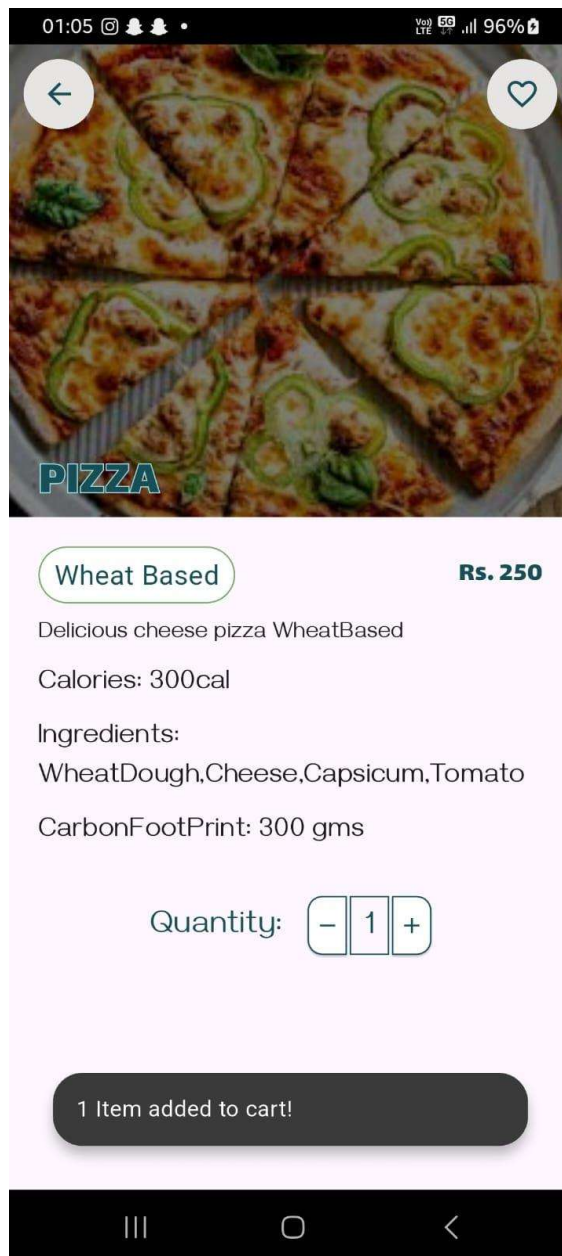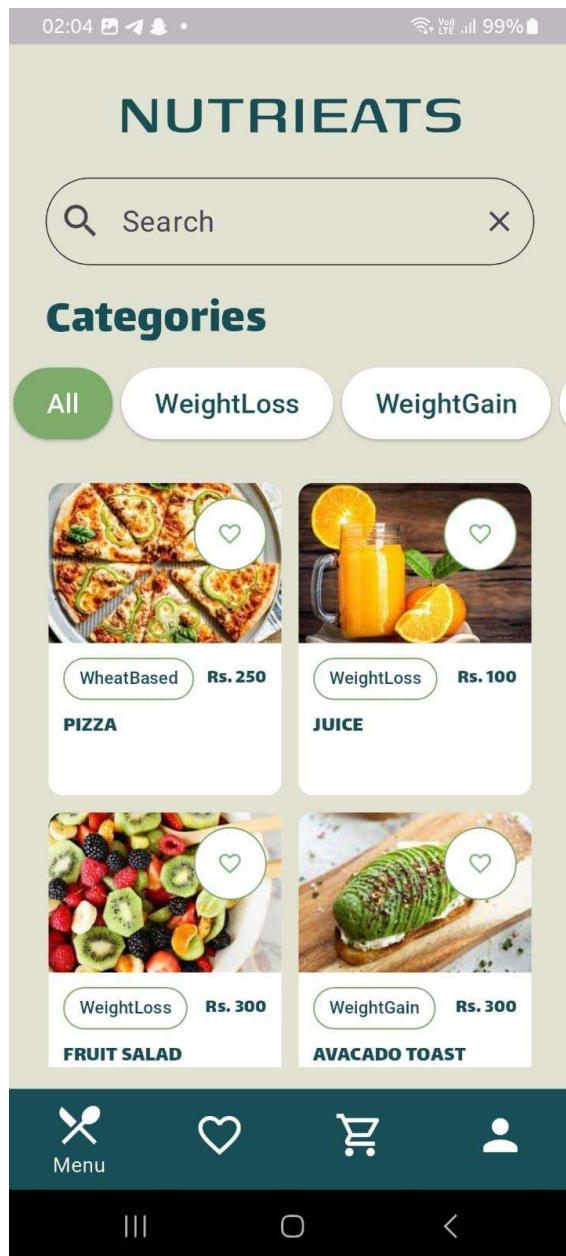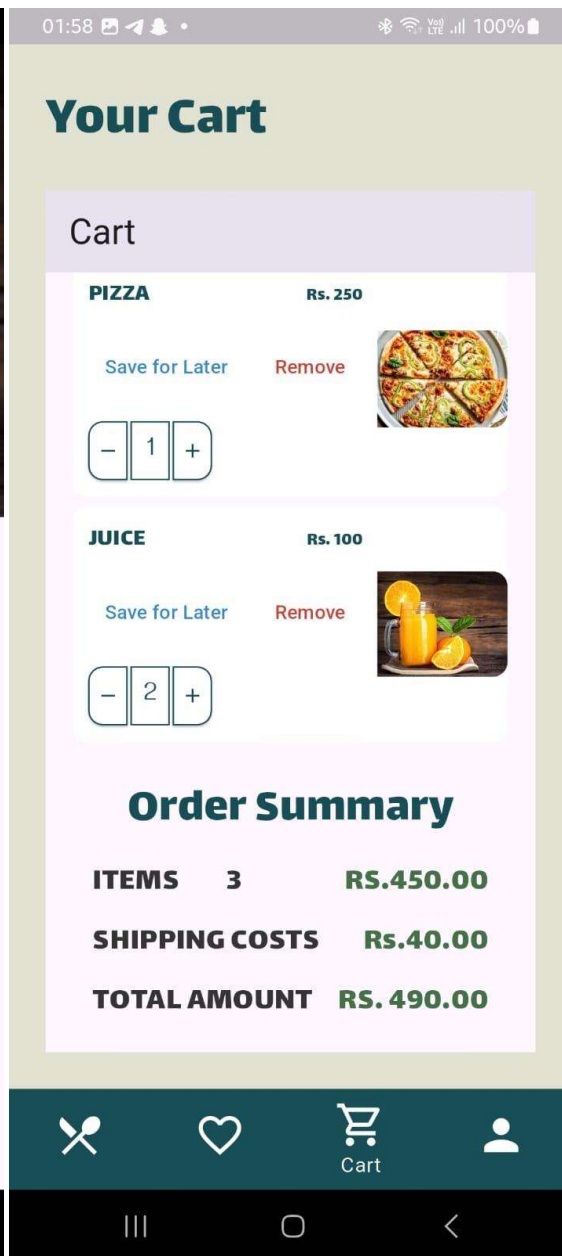
```
alignment: Alignment.centerLeft,
child: Padding(
padding: const EdgeInsets.only(bottom: 20.0),
child: Text(
'Your Cart',
style: TextStyle(
color: Color.fromARGB(255, 24, 79, 87),
fontSize: 35,
fontWeight: FontWeight.w500,
fontFamily: 'Lalezar'),
),
),
),
Expanded(
child: BuildCart(cart:widget.cart),
),



],
)
);
}
}
```

Output:

NUTRIEATS

🔍 Search ✕

Categories

All WeightLoss WeightGain

WheatBased **Rs. 250**
PIZZA

WeightLoss **Rs. 100**
JUICE

WeightLoss **Rs. 300**
FRUIT SALAD

WeightGain **Rs. 300**
AVACADO TOAST

🍴 Menu ♡ 🛒 👤

PIZZA

Wheat Based **Rs. 250**

Delicious cheese pizza WheatBased

Calories: 300cal

Ingredients:
WheatDough,Cheese,Capsicum,Tomato

CarbonFootPrint: 300 gms

Quantity: − 1 +

1 Item added to cart!

Conclusion:

This experiment showed how to use navigation, routing, and gestures in a Flutter app to create a smooth user experience. By applying these concepts, we can easily manage app flow and add interactivity, improving overall usability and app responsiveness.