

EXPERIMENT NO: - 05

Name:- Kunal Punjabi

Class:- D15A

Roll:No: - 43

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

Theory: -

In Flutter, navigation, routing, and gestures play a key role in making an app interactive and user-friendly. The Navigator class is used to manage app navigation by pushing and popping screens (routes). The CartPage in this application demonstrates how navigation and state management work together to create a smooth user experience.

Navigation and Routing in Flutter

Navigation allows switching between different screens in an app. Flutter provides a Navigator widget to handle this using a stack-based approach.

1. Using Navigator for Routing

- The CartPage is navigated to when the user taps the cart icon on the home screen.
- This is done using Navigator.push(), which adds a new screen to the navigation stack.
- To go back, Navigator.pop(context) can be used.

2. State Management in Navigation

The Provider package is used in this app for managing cart data. The CartProvider class:

- Stores cart items using ValueNotifier.
 - Updates UI automatically when items are added or removed.
 - Provides the total price calculation for the cart.
-

Handling Gestures in Flutter

Flutter provides the GestureDetector widget and built-in button widgets to handle gestures. In CartPage, gestures are used for user interactions:

1. Tap Gesture (Adding & Removing Items)

- The IconButton with a delete icon allows users to remove an item from the cart when tapped.

2. ListView Scrolling Gesture

- `ListView.builder()` is used to display cart items in a scrollable list.
 - The user can scroll vertically to see all items.
-

Conclusion:

The `CartPage` effectively demonstrates navigation, routing, and gesture handling in Flutter.

- Navigation: Uses `Navigator.push()` to open the cart screen.
- State Management: Uses `Provider` to dynamically update the cart.
- Gestures: Uses tap gestures for adding/removing items and scrolling for viewing the cart list.

Code: -

Cart_page.dart:

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

class CartPage extends StatelessWidget {
  const CartPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final cartProvider =
    Provider.of<CartProvider>(context);

    return Scaffold(
      appBar: AppBar(title: const Text('Your
      Cart')),
      body:
      ValueListenableBuilder<List<Map<String,
      dynamic>>>>(
        valueListenable:
        cartProvider.cartItems,
        builder: (context, cartItems, child) {
          return cartItems.isEmpty
            ? const Center(child: Text("Your
            cart is empty!"))
            : ListView.builder(
              itemCount: cartItems.length,
              itemBuilder: (context, index) {
                var item = cartItems[index];
                return Card(
                  margin: const
```

```

EdgeInsets.symmetric(vertical: 10,
horizontal: 10),
      child: ListTile(
        leading:
Image.asset(item['imagePath'] ??
'assets/placeholder.jpg', width: 50),
        title: Text(item['productName']
?? 'No Name'),
        subtitle: Text("₹${item['price']}
x ${item['quantity'] ?? 1}"),
        trailing: IconButton(
          icon: const Icon(Icons.delete,
color: Colors.red),
          onPressed: () {

cartProvider.removeFromCart(index);
          },
        ),
      ),
    );
  },
);
},
),
);
}
}
}

```

cart_provider.dart

import

'package:flutter/material.dart';

class CartProvider with

ChangeNotifier {

final

ValueNotifier<List<Map<String

, dynamic>>> _cartItems =

ValueNotifier([]);

ValueNotifier<List<Map<String

, dynamic>>> get cartItems =>

_cartItems;

```

void addToCart(Map<String,
dynamic> item) {
    _cartItems.value =
[... _cartItems.value, item];
    _cartItems.notifyListeners();
}

```

```

void removeFromCart(int
index) {
    _cartItems.value =
List.from(_cartItems.value)..re
moveAt(index);
    _cartItems.notifyListeners();
}

```

```

void updateQuantity(int index,
int quantity) {
    if (quantity > 0) {

        _cartItems.value[index]['quantit
y'] = quantity;
    } else {
        removeFromCart(index);
    }
    _cartItems.notifyListeners();
}

```

```

double getTotalPrice() {
    return
    _cartItems.value.fold(0, (total,
item) => total + (item['price'] *
(item['quantity'] ?? 1)));
}
}

```

8:41



Your Cart



Classic Blue Jeans

₹1599.0 x 1



Ripped Denim

₹1799.0 x 1

