

## **Experiment No. 5**

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

### **THEORY:**

Flutter provides powerful tools for handling navigation between screens, managing routes, and detecting user gestures. These features help in building interactive and seamless user experiences.

### **1. Navigation in Flutter**

Navigation refers to moving between different screens or pages in a Flutter app. Flutter uses a stack-based navigation system where each new screen is pushed onto the navigation stack.

#### **1.1. Types of Navigation**

- Direct Navigation – Moving from one screen to another using the **Navigator** class.
- Named Routes – Predefined routes declared in **MaterialApp** to simplify navigation.
- Navigation with Arguments – Passing data between screens while navigating.

#### **1.2. Back Navigation**

Flutter automatically provides a back button for navigating to the previous screen. However, custom back navigation can be implemented using the **Navigator.pop()** method.

### **2. Routing in Flutter**

Routing is the process of defining and managing the different screens in an app.

#### **2.1. Types of Routing**

- **Static Routing** – Routes are defined in **MaterialApp** and accessed by their names.

- **Dynamic Routing** – Routes are generated dynamically based on user interactions or API responses.

## 2.2. Route Management Approaches

Flutter offers various state management solutions for handling complex navigation:

- **Navigator 1.0 (Imperative Navigation)** – Manages navigation using push and pop methods.
  - **Navigator 2.0 (Declarative Navigation)** – Allows managing the navigation stack declaratively, useful for web and deep linking.
- 

## 3. Gestures in Flutter

Gestures enable user interactions such as taps, swipes, and drags. Flutter provides a built-in `GestureDetector` widget to recognize different gestures.

### 3.1. Common Gestures in Flutter

- **Tap Gesture** – Used for detecting single or double taps on widgets.
- **Swipe Gesture** – Used for implementing swipe actions like dismissing notifications.
- **Long Press Gesture** – Detects prolonged presses, commonly used for showing tooltips or context menus.
- **Drag Gesture** – Enables drag-and-drop functionality.

### 3.2. GestureDetector Widget

The `GestureDetector` widget wraps UI elements and listens for touch interactions, triggering specific actions based on user input.

## CODE & OUTPUT:

## **Product detail:**

```
import 'package:fancy_shimmer_image/fancy_shimmer_image.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_icons/flutter_icons.dart';
import 'package:grocery_app/widgets/heart_btn.dart';
import 'package:provider/provider.dart';

import '../providers/cart_provider.dart';
import '../providers/products_provider.dart';
import '../providers/viewed_prod_provider.dart';
import '../providers/wishlist_provider.dart';
import '../services/utils.dart';
import '../widgets/text_widget.dart';

class ProductDetails extends StatefulWidget {
  static const routeName = '/ProductDetails';

  const ProductDetails({Key? key}) : super(key: key);

  @override
  _ProductDetailsState createState() => _ProductDetailsState();
}

class _ProductDetailsState extends State<ProductDetails> {
  final _quantityTextController = TextEditingController(text: '1');

  @override
  void dispose() {
    // Clean up the controller when the widget is disposed.
    _quantityTextController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    Size size = Utils(context).getScreenSize;
    final Color color = Utils(context).color;
    final productProvider = Provider.of<ProductsProvider>(context);
    final cartProvider = Provider.of<CartProvider>(context);
    final wishlistProvider = Provider.of<WishlistProvider>(context);
    final productId = ModalRoute.of(context)?.settings.arguments as String;
    final getCurrProduct = productProvider.findProdById(productId);
    double usedPrice = getCurrProduct.isOnSale
      ? getCurrProduct.salePrice
```

```

        : getCurrProduct.price;
double totalPrice = usedPrice * int.parse(_quantityTextController.text);
bool? _isInCart = cartProvider.getCartItems.containsKey(getCurrProduct.id);

bool? _isInWishlist =
    wishlistProvider.getWishlistItems.containsKey(getCurrProduct.id);

final viewedProdProvider = Provider.of<ViewedProdProvider>(context);
return WillPopScope(
  onWillPop: () async {
    viewedProdProvider.addProductToHistory(productId: productId);
    return true;
  },
  child: Scaffold(
    appBar: AppBar(
      leading: InkWell(
        borderRadius: BorderRadius.circular(12),
        onTap: () =>
          Navigator.canPop(context) ? Navigator.pop(context) : null,
        child: Icon(
          IcononlyLight.arrowLeft2,
          color: color,
          size: 24,
        ),
      ),
      elevation: 0,
      backgroundColor: Theme.of(context).scaffoldBackgroundColor),
    body: Column(children: [
      Flexible(
        flex: 2,
        child: FancyShimmerImage(
          imageUrl: getCurrProduct.imageUrl,
          boxFit: BoxFit.scaleDown,
          width: size.width,
          // height: screenHeight * .4,
        ),
      ),
      Flexible(
        flex: 3,
        child: Container(
          decoration: BoxDecoration(
            color: Theme.of(context).cardColor,
            borderRadius: const BorderRadius.only(
              topLeft: Radius.circular(40),
              topRight: Radius.circular(40),
            ),
          ),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,

```

```

children: [
  Padding(
    padding:
      const EdgeInsets.only(top: 20, left: 30, right: 30),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Flexible(
          child: TextWidget(
            text: getCurrProduct.title,
            color: color,
            textSize: 25,
            isTitle: true,
          ),
        ),
        HeartBTN(
          productId: getCurrProduct.id,
          isInWishlist: _isInWishlist,
        )
      ],
    ),
  ),
  Padding(
    padding:
      const EdgeInsets.only(top: 20, left: 30, right: 30),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.start,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        TextWidget(
          text: '\₹${usedPrice.toStringAsFixed(2)}',
          color: Colors.green,
          textSize: 22,
          isTitle: true,
        ),
        TextWidget(
          text: getCurrProduct.isPiece ? '/Piece' : '/Kg',
          color: color,
          textSize: 12,
          isTitle: false,
        ),
        const SizedBox(
          width: 10,
        ),
        Visibility(
          visible: getCurrProduct.isOnSale ? true : false,
          child: Text(
            '\₹${getCurrProduct.price.toStringAsFixed(2)}',
            style: TextStyle(

```

```

        fontSize: 15,
        color: color,
        decoration: TextDecoration.lineThrough),
    ),
  ),
  const Spacer(),
  Container(
    padding: const EdgeInsets.symmetric(
      vertical: 4, horizontal: 8),
    decoration: BoxDecoration(
      color: const Color.fromRGBO(63, 200, 101, 1),
      borderRadius: BorderRadius.circular(5)),
    child: TextWidget(
      text: 'Free delivery',
      color: Colors.white,
      textSize: 20,
      isTitle: true,
    ),
  ),
],
),
),
const SizedBox(
  height: 30,
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    quantityControl(
      fct: () {
        if (_quantityTextController.text == '1') {
          return;
        } else {
          setState(() {
            _quantityTextController.text =
              (int.parse(_quantityTextController.text) - 1)
                .toString();
          });
        }
      },
      icon: CupertinoIcons.minus,
      color: Colors.red,
    ),
    const SizedBox(
      width: 5,
    ),
    Flexible(
      flex: 1,
      child: TextField(

```

```

        controller: _quantityTextController,
        key: const ValueKey('quantity'),
        keyboardType: TextInputType.number,
        maxLines: 1,
        decoration: const InputDecoration(
          border: UnderlineInputBorder(),
        ),
        textAlign: TextAlign.center,
        cursorColor: Colors.green,
        enabled: true,
        inputFormatters: [
          FilteringTextInputFormatter.allow(RegExp('[0-9]')),
        ],
        onChanged: (value) {
          setState(() {
            if (value.isEmpty) {
              _quantityTextController.text = '1';
            } else {}
          });
        },
      ),
    ),
    const SizedBox(
      width: 5,
    ),
    quantityControl(
      fct: () {
        setState(() {
          _quantityTextController.text =
            (int.parse(_quantityTextController.text) + 1)
              .toString();
        });
      },
      icon: CupertinoIcons.plus,
      color: Colors.green,
    ),
  ],
),
const Spacer(),
Container(
  width: double.infinity,
  padding: const EdgeInsets.symmetric(
    vertical: 20, horizontal: 30),
  decoration: BoxDecoration(
    color: Theme.of(context).colorScheme.secondary,
    borderRadius: const BorderRadius.only(
      topLeft: Radius.circular(20),
      topRight: Radius.circular(20),
    ),
  ),
),

```

```

    ),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Flexible(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              TextWidget(
                text: 'Total',
                color: Colors.red.shade300,
                textSize: 20,
                isTitle: true,
              ),
              const SizedBox(
                height: 5,
              ),
              FittedBox(
                child: Row(
                  children: [
                    TextWidget(
                      text:
'\₹${totalPrice.toStringAsFixed(2)}/',
                      color: color,
                      textSize: 20,
                      isTitle: true,
                    ),
                    TextWidget(
                      text:
'\₹${_quantityTextController.text}Kg',
                      color: color,
                      textSize: 16,
                      isTitle: false,
                    ),
                  ],
                ),
              ],
            ),
          ),
          const SizedBox(
            width: 8,
          ),
          Flexible(
            child: Material(
              color: Colors.green,
              borderRadius: BorderRadius.circular(10),
              child: InkWell(

```



```

        onTap: _isInCart
          ? null
          : () {
              // if (_isInCart) {
              //   return;
              // }
              cartProvider.addProductsToCart(
                productId: getCurrProduct.id,
                quantity: int.parse(
                  _quantityTextController.text));
            },
        borderRadius: BorderRadius.circular(10),
        child: Padding(
          padding: const EdgeInsets.all(12.0),
          child: TextWidget(
            text:
              _isInCart ? 'In cart' : 'Add to
cart',

            color: Colors.white,
            textSize: 18)),
      ),
    ),
  ],
),
],
),
),
),
),
),
),
),
),
),
);
}

Widget quantityControl(
  {required Function fct, required IconData icon, required Color color}) {
  return Flexible(
    flex: 2,
    child: Material(
      borderRadius: BorderRadius.circular(12),
      color: color,
      child: InkWell(
        borderRadius: BorderRadius.circular(12),
        onTap: () {
          fct();
        },
        child: Padding(
          padding: const EdgeInsets.all(8.0),

```

```
        child: Icon(  
          icon,  
          color: Colors.white,  
          size: 25,  
        ),  
      )),  
    ),  
  );  
}  
}
```

### **Categories Screen:**

```
import 'package:flutter/material.dart';  
import 'package:grocery_app/services/utils.dart';  
import 'package:grocery_app/widgets/categories_widget.dart';  
import 'package:grocery_app/widgets/text_widget.dart';  
  
class CategoriesScreen extends StatelessWidget {  
  CategoriesScreen({Key? key}) : super(key: key);  
  
  List<Color> gridColors = [  
    const Color(0xff53B175),  
    const Color(0xffF8A44C),  
    const Color(0xffF7A593),  
    const Color(0xffD3B0E0),  
    const Color(0xffFDE598),  
    const Color(0xffB7DFF5),  
  ];  
  
  List<Map<String, dynamic>> catInfo = [  
    {  
      'imgPath': 'assets/images/cat/fruits.png',  
      'catText': 'Fruits',  
    },  
    {  
      'imgPath': 'assets/images/cat/veg.png',  
      'catText': 'Vegetables',  
    },  
    {  
      'imgPath': 'assets/images/cat/Spinach.png',  
      'catText': 'Herbs',  
    },  
    {  
      'imgPath': 'assets/images/cat/nuts.png',  
      'catText': 'Nuts',  
    },  
  ],  
}
```

```

    },
    {
      'imgPath': 'assets/images/cat/spices.png',
      'catText': 'Spices',
    },
    {
      'imgPath': 'assets/images/cat/grains.png',
      'catText': 'Grains',
    },
  ],
];
@override
Widget build(BuildContext context) {

  final utils = Utils(context);
  Color color = utils.color;
  return Scaffold(
    appBar: AppBar(
      elevation: 0,
      backgroundColor: Theme.of(context).scaffoldBackgroundColor,
      title: TextWidget(
        text: 'Categories',
        color: color,
        textSize: 24,
        isTitle: true,
      ),
    ),
    body: Padding(
      padding: const EdgeInsets.all(8.0),
      child: GridView.count(
        crossAxisCount: 2,
        childAspectRatio: 240 / 250,
        crossAxisSpacing: 10, // Vertical spacing
        mainAxisSpacing: 10, // Horizontal spacing
        children: List.generate(6, (index) {
          return CategoriesWidget(
            catText: catInfo[index]['catText'],
            imgPath: catInfo[index]['imgPath'],
            passedColor: gridColors[index],
          );
        })),
    ),
  );
}

```

## **Cart screen:**

```
import 'package:flutter/cupertino.dart';
import 'package:grocery_app/models/cart_model.dart';

class CartProvider with ChangeNotifier {
  Map<String, CartModel> _cartItems = {};

  Map<String, CartModel> get getCartItems {
    return _cartItems;
  }

  void addProductsToCart({
    required String productId,
    required int quantity,
  }) {
    _cartItems.putIfAbsent(
      productId,
      () => CartModel(
        id: DateTime.now().toString(),
        productId: productId,
        quantity: quantity,
      ),
    );
    notifyListeners();
  }

  void reduceQuantityByOne(String productId) {
    _cartItems.update(
      productId,
      (value) => CartModel(
        id: value.id,
        productId: productId,
        quantity: value.quantity - 1,
      ),
    );
    notifyListeners();
  }

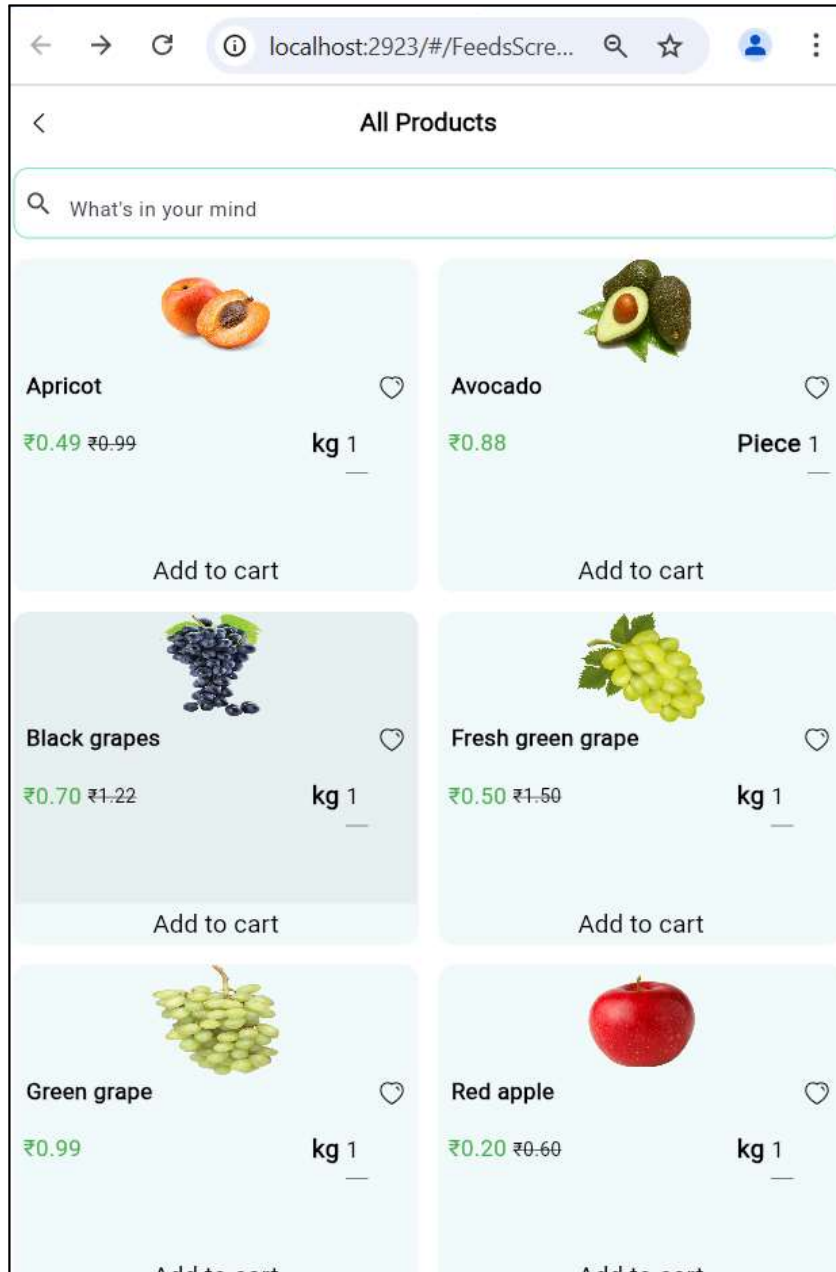
  void increaseQuantityByOne(String productId) {
    _cartItems.update(
      productId,
      (value) => CartModel(
        id: value.id,
        productId: productId,
        quantity: value.quantity + 1,
      ),
    );
  }
}
```

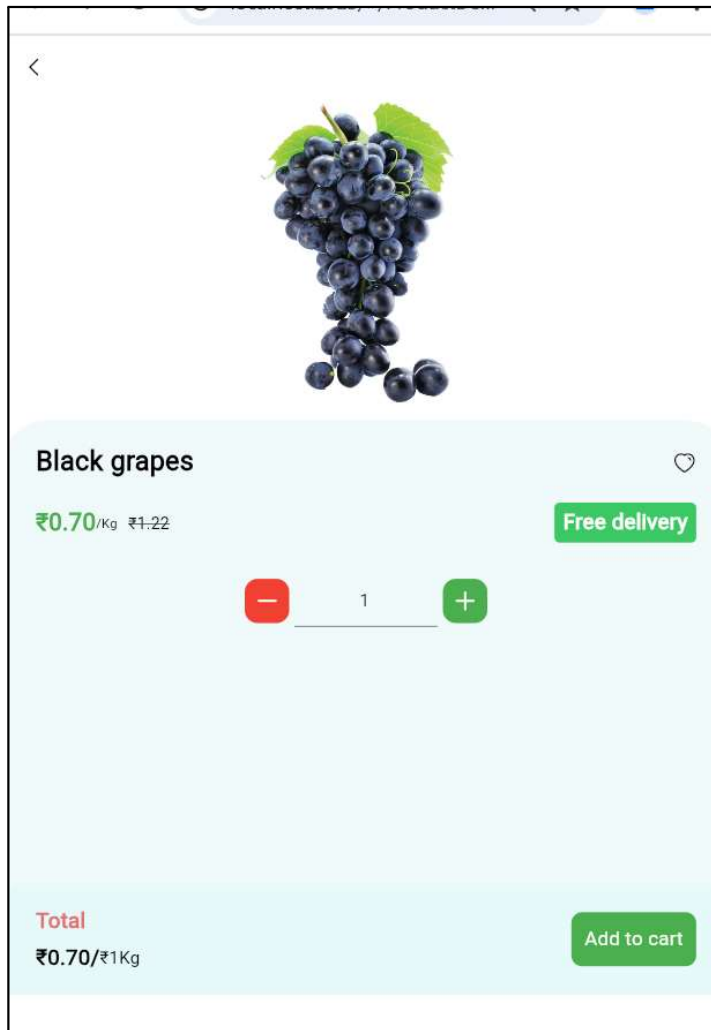
```
        notifyListeners();
    }

    void removeOneItem(String productId) {
        _cartItems.remove(productId);
        notifyListeners();
    }

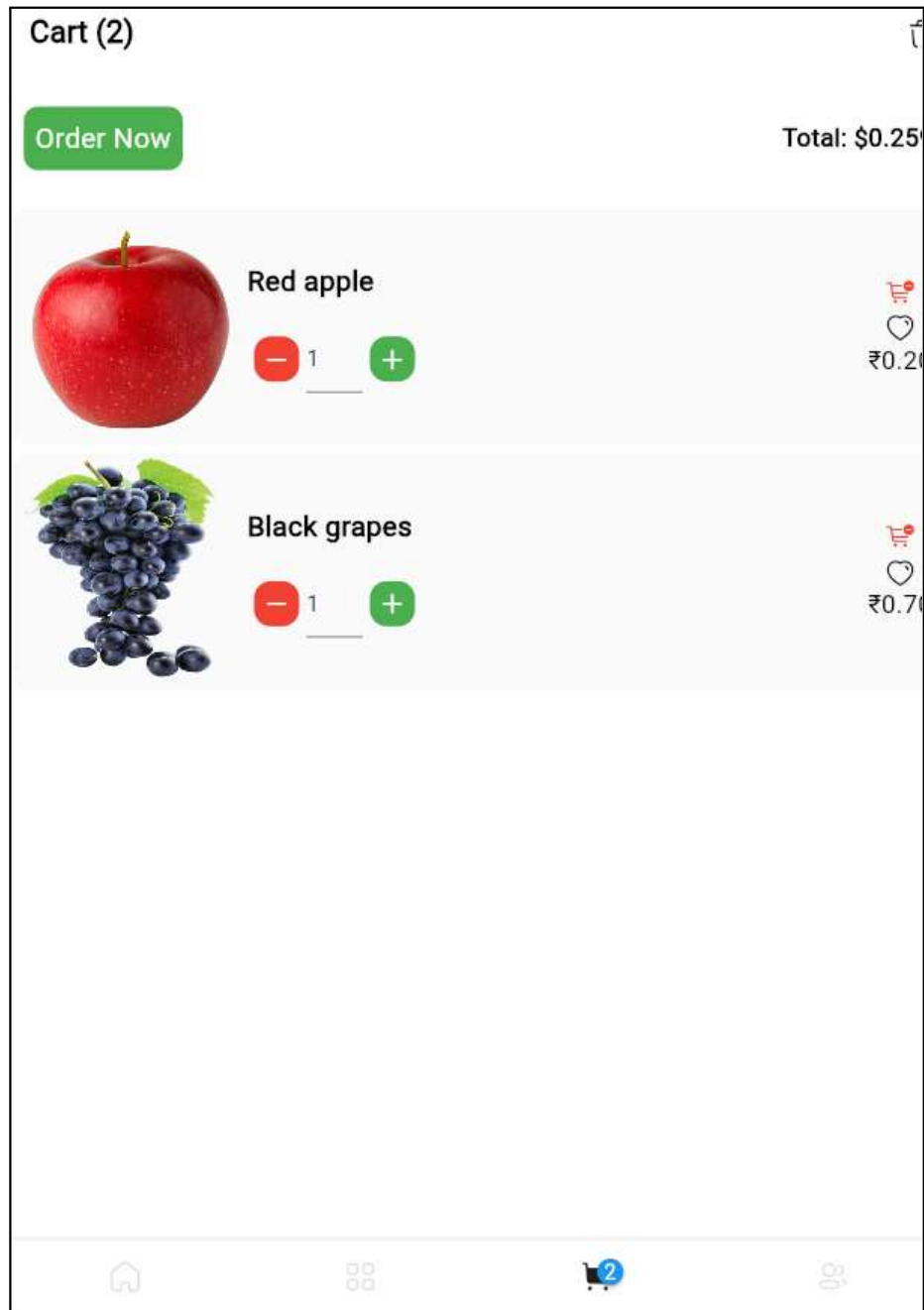
    void clearCart() {
        _cartItems.clear();
        notifyListeners();
    }
}
```

1. Clicked a product then it navigatesto product detail page .





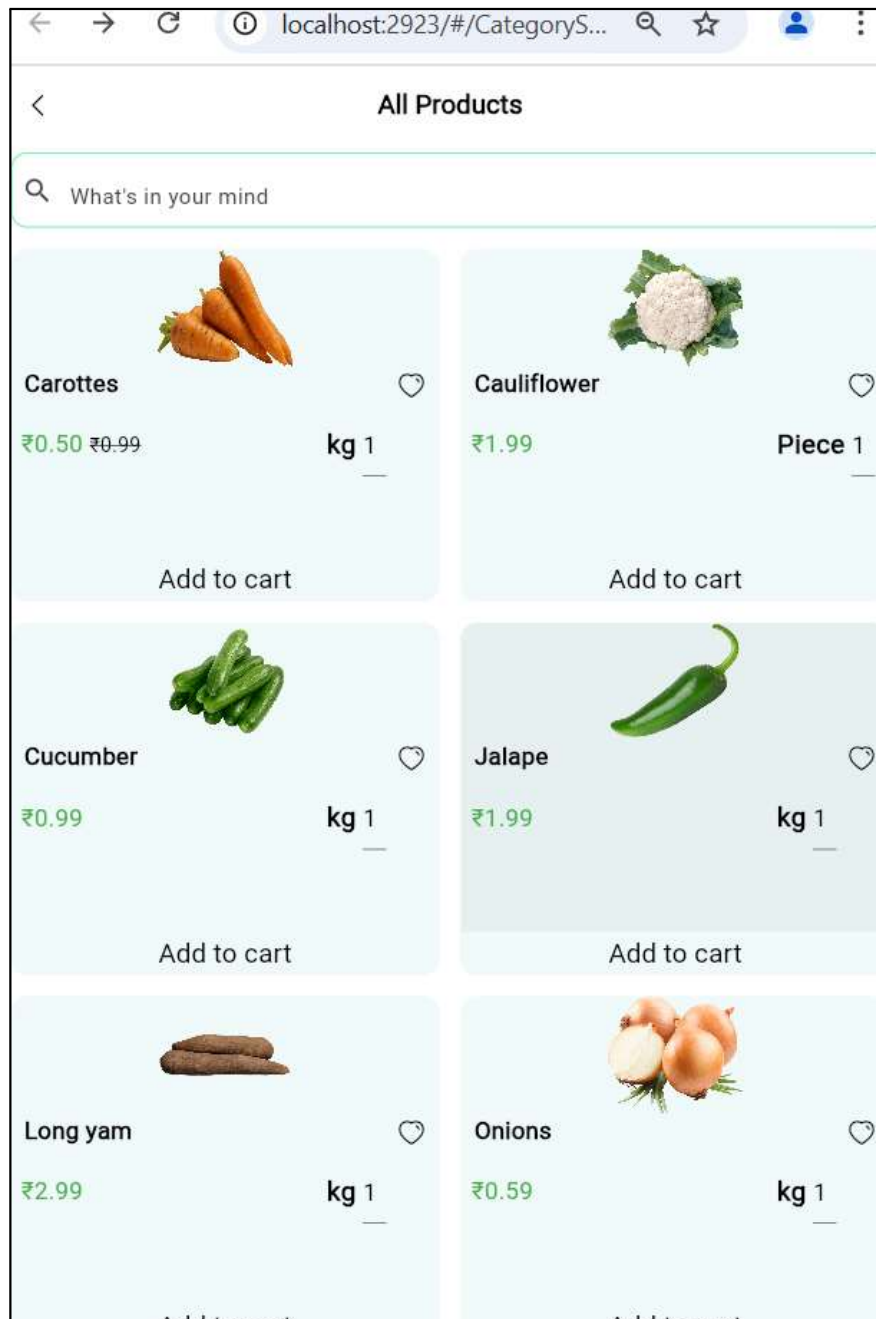
**2 You add the product in cart it routes and gets store in cart:**





**3. When you click any category it goes to that particular page**





Sonam chhabaidiya D15A 09

Sonam chhabaidiya D15A 09