

EXPERIMENT NO: - 02

Name:- Kunal Punjabi

Class:- D15A

Roll:No: - 43

AIM: - To design Flutter UI by including common widgets.

Theory: Flutter UI in Jeans Shopping App

Each element on the screen of a Flutter app is a widget. The UI structure is built using a tree of widgets, where each widget defines either a visual component, a layout structure, or user interactions.

The appearance and behavior of the app are determined by the choice, sequence, and nesting of these widgets.

Flutter follows a reactive UI approach, meaning when any code modification occurs, only the necessary widgets are rebuilt based on the difference between the previous and the current state. This ensures efficient UI rendering.

In this Jeans Shopping App, various single-child and multi-child widgets are used to construct the UI efficiently.

Types of Widgets in Jeans Shopping App

1. Stateful and Stateless Widgets

- **StatelessWidget:** These widgets do not change once built. The HomePage class extends StatelessWidget, displaying a list of jeans products.
- **StatefulWidget:** These widgets maintain state and update dynamically. The CartProvider class (used with Provider) helps manage the cart's state, ensuring live updates when products are added.

Commonly Used Widgets in Jeans Shopping App

1. Layout Widgets

- Scaffold: Provides the base structure, including the AppBar and Body.
- GridView.builder: Used to display the list of jeans in a structured grid format, making product browsing easier.
- Card: Each product is enclosed within a material Card widget for better visual appeal.
- Column & Row: Used to align text, buttons, and product details properly inside each product card.

2. Interactive Widgets

- GestureDetector: Detects taps on a product and navigates to ProductDetailsPage.
- IconButton: The shopping cart icon on the AppBar allows users to navigate to the cart page.
- Stack & Positioned: Used to display a dynamic cart count badge over the shopping cart icon.

3. Image and Text Widgets

- Image.asset: Displays product images stored in the app's assets.
- Text: Displays product names and prices with customized styles.

4. Navigation and State Management

- Navigator.push: Handles page transitions when the user selects a product.
- Provider & ValueListenableBuilder: Manage cart updates and dynamically display the number of items in the cart.

Code : homepage.dart

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

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

  final List<Map<String, dynamic>>> jeansProducts = const [
    {
      'imagePath': 'assets/jeans0.jpg',
      'productName': 'Classic Blue Jeans',
      'price': 1599.0,
      'colors': ['Blue', 'Dark Blue', 'Light Blue'],
    },
    {
      'imagePath': 'assets/jeans1.jpg',
      'productName': 'Ripped Denim',
      'price': 1799.0,
      'colors': ['Black', 'Gray', 'Navy Blue'],
    },
  ];
}

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

  return Scaffold(
    appBar: AppBar(
      title: const Text('Jeans Shop'),
      actions: [
        Stack(
          children: [
            IconButton(
              icon: const Icon(Icons.shopping_cart),
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => const CartPage(), // 
                );
              },
            ),
            ValueListenableBuilder<List<Map<String,
dynamic>>>(

```

```
valueListenable: cartProvider.cartItems,
builder: (context, cartItems, child) {
  return cartItems.isNotEmpty
    ? Positioned(
        right: 10,
        top: 5,
        child: Container(
          padding: const EdgeInsets.all(6),
          decoration: const BoxDecoration(
            color: Colors.red,
            shape: BoxShape.circle,
          ),
        child: Text(
          cartItems.length.toString(),
          style: const TextStyle(
            color: Colors.white,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
    ),
  ),
),
],
),
],
),
),
body: GridView.builder(
  padding: const EdgeInsets.all(10),
  gridDelegate: const
SliverGridDelegateWithFixedCrossAxisCount(
  crossAxisCount: 2,
  crossAxisSpacing: 10,
  mainAxisSpacing: 10,
  childAspectRatio: 0.8,
),
itemCount: jeansProducts.length,
itemBuilder: (context, index) {
  var product = jeansProducts[index];
  return GestureDetector(
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => ProductDetailsPage(
            imagePath: product['imagePath'],
            productName: product['productName'],
            price: product['price'],
            colors: product['colors'],
          ),
        ),
      );
    },
  );
},
```

```
        ),
    );
},
child: Card(
  elevation: 5,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(10),
  ),
),
child: Column(
  mainAxisAlignment: MainAxisAlignment.start,
  children: [
    Expanded(
      child: ClipRRect(
        borderRadius: const BorderRadius.vertical(
          top: Radius.circular(10),
        ),
        child: Image.asset(
          product['imagePath'],
          fit: BoxFit.cover,
          width: double.infinity,
        ),
      ),
    ),
  ],
),
Padding(
  padding: const EdgeInsets.all(8),
  child: Column(
    mainAxisAlignment:
CrossAxisAlignment.start,
    children: [
      Text(
        product['productName'],
        style: const TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 14,
        ),
      ),
      const SizedBox(height: 4),
      Text(
        "₹${product['price'].toStringAsFixed(2)}",
        style: const TextStyle(
          color: Colors.green,
          fontWeight: FontWeight.bold,
        ),
      ),
    ],
),
),
],
),
),
),
);
});
```

```
    },  
    ),  
    );  
}  
}
```

Screenshots:



