

## **Experiment No. 3**

**AIM:** To include icons, images, fonts in Flutter app

### **THEORY:**

Flutter provides built-in support for adding icons, images, and custom fonts to enhance the visual appeal of applications. These assets help in creating a more engaging and personalized user interface.

### **1. Adding Icons in Flutter**

Icons are commonly used to represent actions, categories, or UI elements. Flutter offers both built-in icons and custom icon support.

#### **1.1. Built-in Material Icons**

Flutter provides a rich set of **Material Icons**, which can be used directly in the UI. These icons follow Google's Material Design guidelines and do not require additional setup.

#### **1.2. Custom Icons**

For unique icons, custom icon packs can be added using third-party libraries like **FontAwesome** or by importing **SVG** or **PNG** icons as images.

#### **1.3. Vector Icons (SVG)**

Vector icons are scalable and provide better quality across different screen sizes. The **flutter\_svg** package is commonly used for adding SVG icons.

---

## 2. Adding Images in Flutter

Images enhance the visual representation of an app. Flutter supports various types of images:

### 2.1. Asset Images (Local Images)

Local images are stored within the app's **assets** folder. These images must be declared in the **pubspec.yaml** file before use.

### 2.2. Network Images

Flutter allows loading images from the internet using their URLs. This is useful for displaying dynamic content fetched from APIs.

### 2.3. Memory and File Images

Flutter can also load images directly from device storage or memory, making it useful for user-uploaded images.

---

## 3. Adding Custom Fonts in Flutter

Custom fonts improve typography and branding in the app.

### 3.1. Using System Fonts

Flutter supports default system fonts available on the device, such as Roboto and San Francisco.

### 3.2. Adding Custom Fonts

To include a custom font, the font file (e.g., **.ttf**, **.otf**) must be added to the **assets/fonts** directory and declared in **pubspec.yaml**.

### 3.3. Applying Fonts

Once declared, the custom font can be applied globally or to specific text widgets for a unique look.

#### **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(

```



```

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

```

## Wishlist:

```

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

class WishlistProvider with ChangeNotifier {
  Map<String, WishlistModel> _wishlistItems = {};

  Map<String, WishlistModel> get getWishlistItems {
    return _wishlistItems;
  }

  void addRemoveProductToWishlist({required String productId}) {
    if (_wishlistItems.containsKey(productId)) {
      removeOneItem(productId);
    } else {
      _wishlistItems.putIfAbsent(
        productId,
        () => WishlistModel(
          id: DateTime.now().toString(), productId: productId));
    }
    notifyListeners();
  }

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

```

```

}

void clearWishlist() {
  _wishlistItems.clear();
  notifyListeners();
}
}

```

## **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],
          );
        }),
      ),
    ));
}

```

## Profile:

Hi, MyName

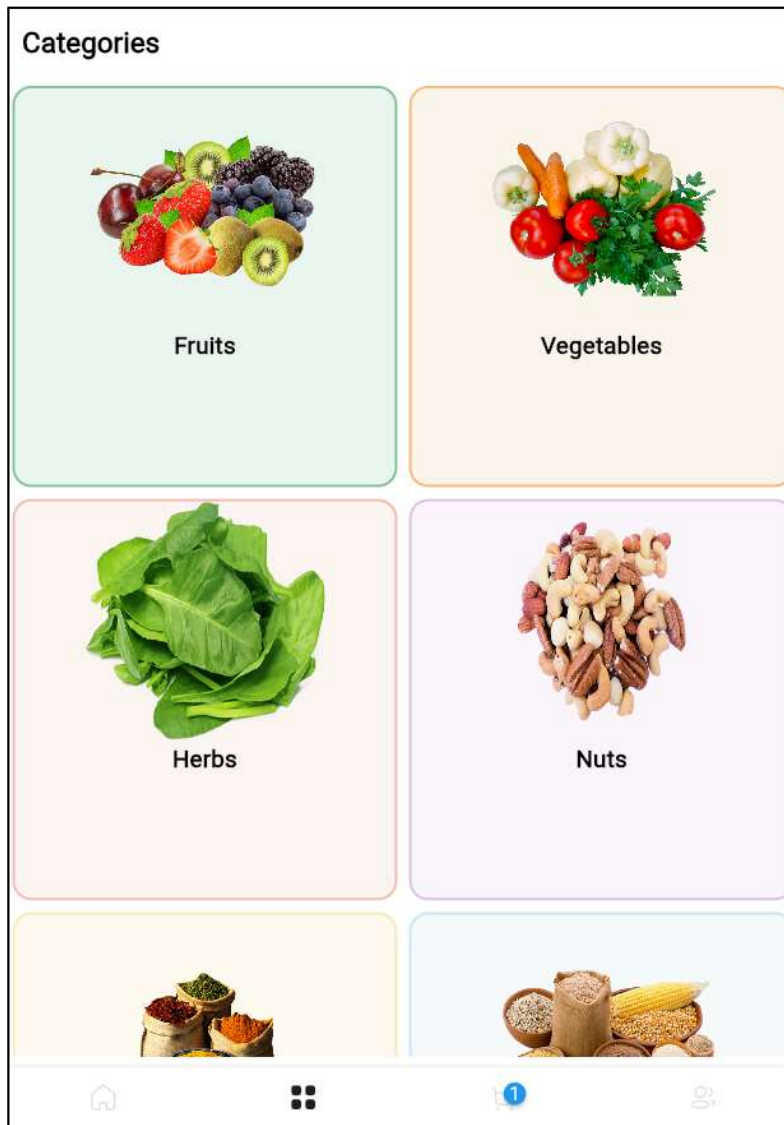
Email@email.com

-  Address 2  
My subtitle >
-  Orders >
-  Wishlist >
-  Viewed >
-  Forget password >
-  Light mode ☒
-  Logout >

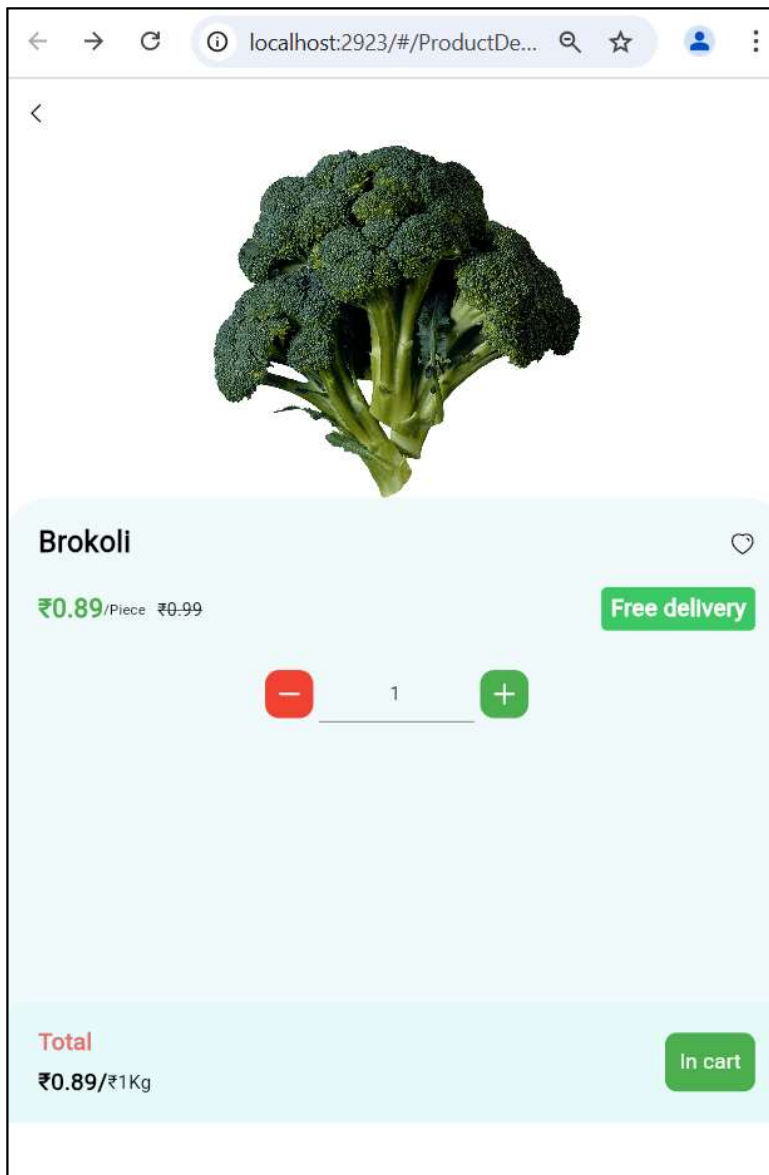


## Images:

### 1. Categories screen:



## 2. Product detail image:



### 3. Wishlist screen:

