EXPERIMENT NO: 3

Aim: To include icons, images, fonts in Flutter app

Theory:

1. Including Icons in Flutter

Icons are essential UI elements used for navigation, buttons, and branding.

Built-in Icons (Material Icons)

Flutter provides a set of predefined Material icons that can be used directly.

Example:

```
Icon(Icons.home, size: 30, color: Colors.blue);
```

- Icons is a predefined class containing various Material icons.
- Can be customized using size and color properties.

Custom Icons using Packages (Font Awesome, Feather Icons, etc.)

You can use external icon libraries like Font Awesome by adding dependencies in

pubspec.yaml:

```
dependencies:
  flutter:
    sdk: flutter
  font_awesome_flutter: ^10.4.0
```

Usage in Dart:

```
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
Icon(FontAwesomeIcons.twitter, size: 30, color: Colors.blue);
```

2. Including Images in Flutter

Flutter allows adding images from assets or network sources.

Adding Local Images (Assets)

1. Place images in the assets/images/ directory (create one if not present).
2. Add assets in pubspec.yaml:

```
flutter:

  assets:
    - assets/images/logo.png
```

3. Use in Dart Code:

```
Image.asset("assets/images/logo.png", width: 100, height: 100);
```

Using Network Images

You can directly fetch images from the internet using Image.network():

```
Image.network("https://example.com/image.png", width: 100, height: 100);
```

3. Including Custom Fonts in Flutter

Flutter supports custom fonts for a unique app design.

Steps to Add a Custom Font

1. Place font files (.ttf) inside assets/fonts/ directory.
2. Declare fonts in pubspec.yaml:

Example:

```
flutter:
  fonts:
    - family: CustomFont
      fonts:
        - asset: assets/fonts/CustomFont-Regular.ttf
        - asset: assets/fonts/CustomFont-Bold.ttf
          weight: 700
```

3. Use the font in Dart Code:

Example:

```
Text(
  "Hello Flutter!",
  style: TextStyle(fontFamily: "CustomFont", fontSize: 20),
);
```

Code:

Network_image_with_loader.dart:

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';

import '../constants.dart';
import 'skleton/skelton.dart';

class NetworkImageWithLoader extends StatelessWidget {
  final BoxFit fit;
  const NetworkImageWithLoader(
    this.src, {
    super.key,
    this.fit = BoxFit.cover,
    this.radius = defaultPadding,
  });

  final String src;
  final double radius;

  @override
  Widget build(BuildContext context) {
    return ClipRRect(
      borderRadius: BorderRadius.all(Radius.circular(radius)),
      child: CachedNetworkImage(
        fit: fit,
        imageUrl: src,
        imageBuilder: (context, imageProvider) => Container(
          decoration: BoxDecoration(
            image: DecorationImage(
              image: imageProvider,
              fit: fit,
            ),
          ),
        ),
        placeholder: (context, url) => const Skeleton(),
        errorWidget: (context, url, error) => const Icon(Icons.error),
      ),
    );
  }
}
```

Order_status_card.dart:

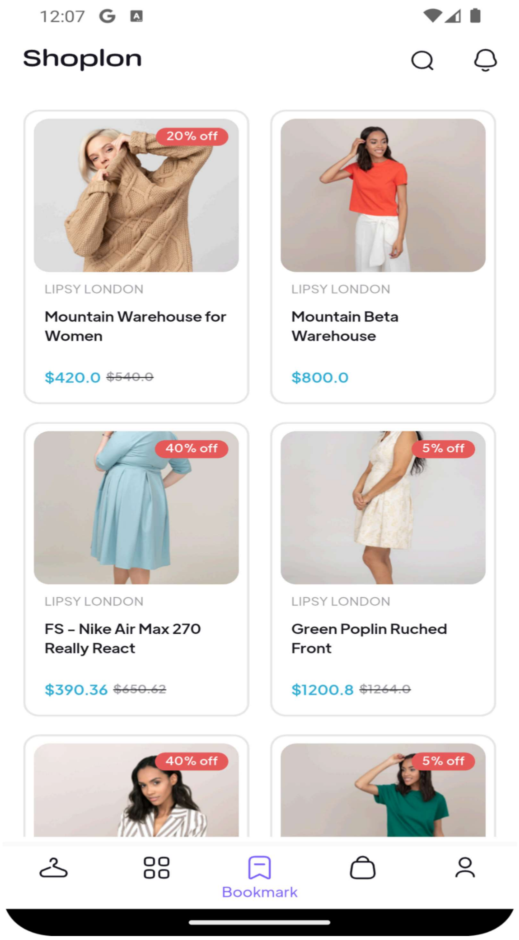
```
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';

import '../constants.dart';
import 'order_process.dart';

class OrderStatusCard extends StatelessWidget {
  const OrderStatusCard({
    super.key,
    required this.orderId,
    required this.placedOn,
    this.products,
    required this.orderStatus,
    required this.processingStatus,
    required this.packedStatus,
    required this.shippedStatus,
    required this.deliveredStatus,
    this.press,
    this.isCancled = false,
  });

  final String orderId, placedOn;
  final List<Widget>? products;
  final OrderProcessStatus orderStatus,
      processingStatus,
      packedStatus,
      shippedStatus,
      deliveredStatus;
  final VoidCallback? press;
  final bool isCancled;

  @override
  Widget build(BuildContext context) {
    return InkWell(
      borderRadius:
          const BorderRadius.all(Radius.circular(defaultBorderRadious)),
      onTap: press,
      child: Column(
        mainAxisSize: MainAxisSize.min,
        children: [
          Container(
            decoration: BoxDecoration(
              borderRadius:
                  const BorderRadius.all(Radius.circular(defaultBorderRadious)),
              border: Border.all(color: Theme.of(context).dividerColor),
            ),
            child: Column(
              children: [
                Padding(
                  padding: const EdgeInsets.all(defaultPadding),
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                      Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          DefaultTextStyle(
                            style: TextStyle(
                              fontWeight: FontWeight.w500,
              fontSize: 12,
              color:
                  Theme.of(context).textTheme.bodyMedium!.color,
            ),
            child: Row(
              children: [
                const Text("Order"),
                const SizedBox(width: defaultPadding / 2),
                Text("#$orderId"),
              ],
            ),
          ),
          const SizedBox(height: defaultPadding / 2),
          Text(
            "Placed on $placedOn",
            style: Theme.of(context).textTheme.titleSmall,
          ),
        ],
      ),
      SvgPicture.asset(
        "assets/icons/miniRight.svg",
        height: 24,
        width: 24,
        colorFilter: ColorFilter.mode(
            Theme.of(context).dividerColor, BlendMode.srcIn),
      )
    ],
  ),
),
const Divider(height: 1),
Padding(
  padding: const EdgeInsets.symmetric(vertical: defaultPadding),
  child: OrderProgress(
    orderStatus: orderStatus,
    processingStatus: processingStatus,
    packedStatus: packedStatus,
    shippedStatus: shippedStatus,
    deliveredStatus: deliveredStatus,
    isCanceled: isCancled,
  ),
),
Padding(
  padding:
      const EdgeInsets.symmetric(horizontal: defaultPadding),
  child: Column(
    children: products ?? [],
  ),
)
```

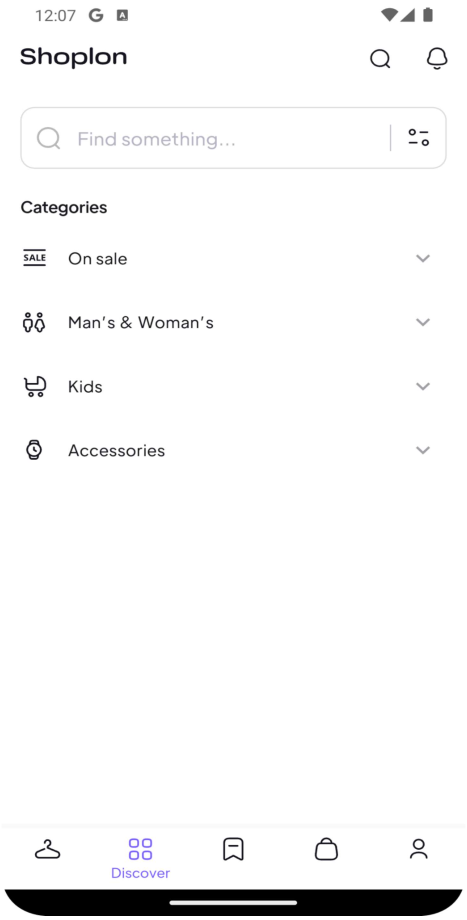
```
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

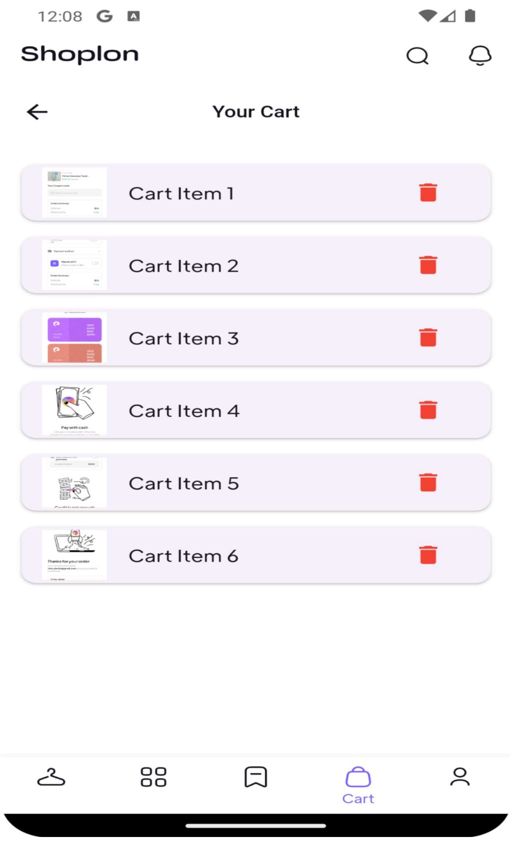
Shopping_bag.dart:

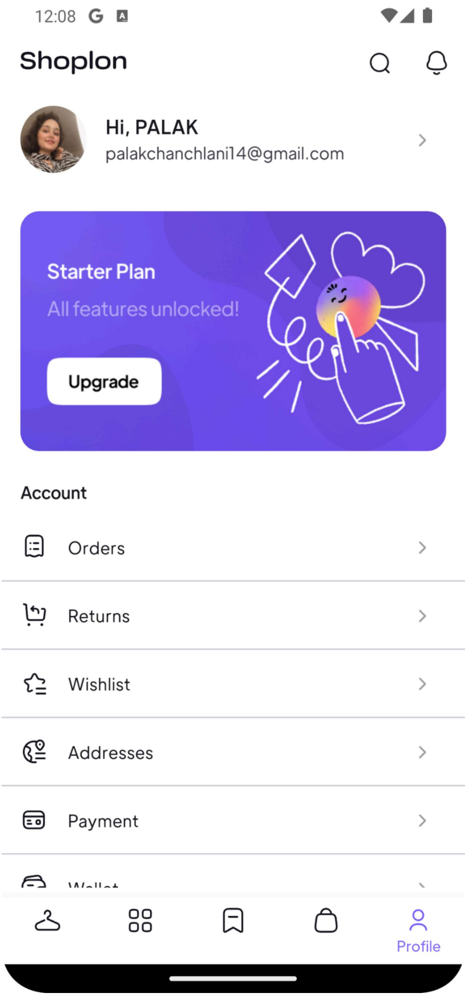
```
import 'package:flutter/material.dart';
import 'package:flutter_svg/flutter_svg.dart';

import '../constants.dart';

class ShoppingBag extends StatelessWidget {
  const ShoppingBag({
    super.key,
    this.color,
    this.numOfItem,
  });

  final Color? color;
  final int? numOfItem;

  @override
  Widget build(BuildContext context) {
    return IconButton(
      onPressed: () {},
      icon: Stack(
        alignment: Alignment.center,
        clipBehavior: Clip.none,
        children: [
          SvgPicture.asset(
            numOfItem == null
                ? "assets/icons/Bag.svg"
                : "assets/icons/bag_full.svg",
            height: 24,
            width: 24,
            colorFilter: ColorFilter.mode(
                color ?? Theme.of(context).iconTheme.color!, BlendMode.srcIn),
          ),
          if (numOfItem != null)
            Padding(
              padding: const EdgeInsets.only(top: 4),
              child: Text(
                numOfItem!.toString(),
                style: TextStyle(
                  fontFamily: grandisExtendedFont,
                  fontSize: 10,
                  fontWeight: FontWeight.bold,
                  color: Theme.of(context).brightness == Brightness.dark
                      ? blackColor
                      : Colors.white,
                ),
              ),
            )
        ],
      ),
    );
  }
}
```

Output:









Conclusion:

Flutter provides flexible ways to include **icons, images, and custom fonts** to enhance the app's visual appeal. **Material and custom icons** help in better navigation, **asset and network images** improve the UI, and **custom fonts** give a unique branding to the app. Implementing these elements correctly results in a professional and aesthetically pleasing application.