PRANAV RAIKAR
D15A 47

AIM: To design Flutter UI by including common widgets.

Theory:

In summary, Flutter widgets are fundamental components in constructing the user interface of a Flutter application. They can be broadly categorized into two types: `StatelessWidget` representing immutable parts of the UI and `StatefulWidget` representing mutable components that can change over time.

Some key Flutter widgets include:

1. Scaffold: The basic structure for a Flutter app, providing layout elements such as AppBar, BottomNavigationBar, and a body for main content.

2. Container: A versatile box model used for layout, padding, margin, decoration, and constraints, capable of containing other widgets.

3. Row & Column: Widgets for arranging child widgets horizontally (Row) or vertically (Column), essential for creating flexible and responsive layouts.

4. Text: Used for displaying text on the screen with support for various styling options like font size, color, and alignment.

5. TextField: Captures user input, such as text, numbers, or passwords, with the `onChanged` property for dynamic updates based on user input.

6. Buttons: Various button widgets like `ElevatedButton` or `TextButton` trigger actions when pressed, providing a means for user interaction.

7. Forms: The `Form` widget manages a group of `TextFormField` widgets, facilitating input validation and submission.

8. Icons: The `Icon` widget displays icons from libraries, enhancing visual elements and conveying meaning through symbols.

Key Design Principles highlighted include:

- Consistency: Common widget usage fosters a consistent design language throughout the app.

- Responsive Layouts: Widgets like `Row` and `Column` aid in creating responsive and flexible layouts, adapting to different screen sizes.

- User Input Handling: `TextField` and `Form` widgets facilitate proper handling, ensuring data integrity and validation.

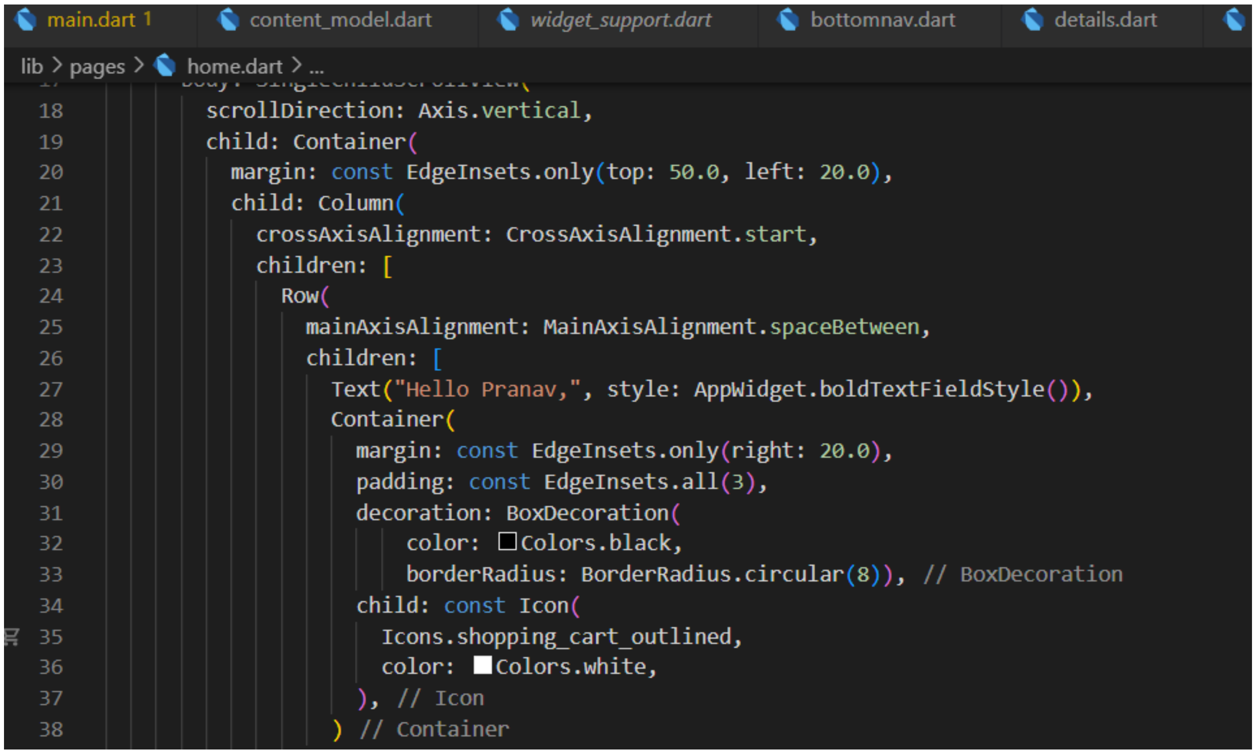
- Interactive Elements: Buttons and icons contribute to interactivity and user engagement within the app.

- Visual Styling: The `Container` widget and styling properties of other widgets allow for visual customization and theming.

Common widgets is used for different type of fonts:

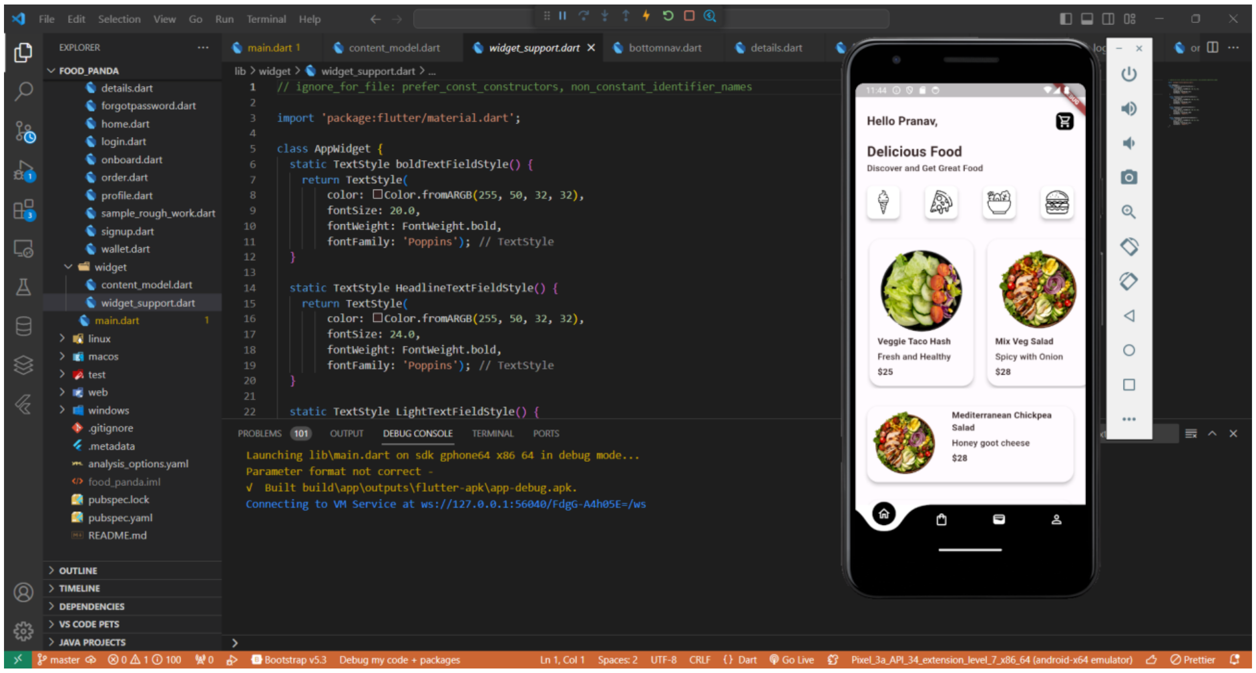
```
// ignore_for_file: prefer_const_constructors, non_constant_identifier_names

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

```
class AppWidget {
  static TextStyle boldTextFieldStyle() {
    return TextStyle(
        color: Color.fromARGB(255, 50, 32, 32),
        fontSize: 20.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'Poppins');
  }

  static TextStyle HeadlineTextFieldStyle() {
    return TextStyle(
        color: Color.fromARGB(255, 50, 32, 32),
        fontSize: 24.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'Poppins');
  }

  static TextStyle LightTextFieldStyle() {
    return TextStyle(
        color: Color.fromARGB(255, 50, 32, 32),
        fontSize: 15.0,
        fontWeight: FontWeight.w500,
        fontFamily: 'Poppins');
  }

  static TextStyle semiBoldTextFieldStyle() {
    return TextStyle(
        color: Color.fromARGB(255, 50, 32, 32),
        fontSize: 15.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'Poppins');
  }
}
```



```dart
        scrollDirection: Axis.vertical,
        child: Container(
          margin: const EdgeInsets.only(top: 50.0, left: 20.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Text("Hello Pranav,", style: AppWidget.boldTextFieldStyle()),
                  Container(
                    margin: const EdgeInsets.only(right: 20.0),
                    padding: const EdgeInsets.all(3),
                    decoration: BoxDecoration(
                        color: Colors.black,
                        borderRadius: BorderRadius.circular(8)), // BoxDecoration
                    child: const Icon(
                      Icons.shopping_cart_outlined,
                      color: Colors.white,
                    ), // Icon
                  ) // Container
```

In above picture AppWidget is the widget which is common at many places in code



```dart
// ignore_for_file: prefer_const_constructors, non_constant_identifier_names

import 'package:flutter/material.dart';

class AppWidget {
  static TextStyle boldTextFieldStyle() {
    return TextStyle(
        color: Color.fromARGB(255, 50, 32, 32),
        fontSize: 20.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'Poppins'); // TextStyle
  }

  static TextStyle HeadlineTextFieldStyle() {
    return TextStyle(
        color: Color.fromARGB(255, 50, 32, 32),
        fontSize: 24.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'Poppins'); // TextStyle
  }

  static TextStyle LightTextFieldStyle() {
```

```
Launching lib\main.dart on sdk gphone64 x86 64 in debug mode...
Parameter format not correct -
√  Built build\app\outputs\flutter-apk\app-debug.apk.
Connecting to VM Service at ws://127.0.0.1:56040/FdgG-A4h05E=/ws
```

Conclusion :Thus I learnt to create and use common widgets